

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERRI de Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

PROJET DE FIN D'ETUDES

En vue de l'obtention du diplôme

D'INGENIEUR D'ETAT EN AUTOMATIQUE

Thème

Classification non supervisée par la carte auto-organisatrice de Kohonen : application à la segmentation d'images couleur

Proposé et dirigé par : Mme **Sadia ALKAMA**

Présenté par : M^{elle} **Sonia BELOUNIS**

Soutenu le : 08 / 07 /2009 Devant le jury d'examen composé de : A.DIRAMI
K.HAMMOUCHE
F.BOUDJEMEEA

Promotion juin2009

Remerciements

Avant tout développement de ce travail, il apparaît opportun de commencer par des remerciements. A ceux qui m'ont beaucoup appris au cours de mon cursus, et même à ceux qui ont juste eu la gentillesse d'en faire un moment profitable et très agréable.

*Je tiens à exprimer mon immense gratitude à **Mme Alkama**, ma promotrice qui m'a formé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie.*

Je remercie également tous les membres de ma famille pour leurs aides précieuses, leurs soutient et encouragements.

Tous mes remerciements vont aussi aux membres du jury ayant accepté d'analyser et de corriger mon travail.

Je tiens aussi, à remercier vivement tout les enseignants que j'ai eu durant ma formation en particulier ceux du département d'automatique, notamment

MR DIRAMI.

Ainsi que tous les ami(e)s qui ont colorés mes journées et soutenus tout au long des périodes difficiles

Enfin, je remercie l'ensemble des personnes que j'ai été amené à côtoyer durant ma formation, pour les conseils qu'ils ont pu me prodiguer.

Sonia BELOUNIS

Avant propos

Le traitement d'images est une des grandes orientations du traitement de l'information acquise à partir des capteurs, qui sont de plus en plus performants. De nombreuses opérations relevant de l'analyse d'image qui auparavant s'effectuées manuellement peuvent aujourd'hui être résolues automatiquement. De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, afin d'améliorer leur aspect visuel et d'en extraire des informations jugées pertinentes.

Pour reconnaître des objets afin d'interpréter les images, il faut souvent au préalable les segmenter. Cette étape est donc très importante dans l'analyse des images. C'est une opération qui consiste à partitionner une image en régions homogènes. L'emploi de la couleur en segmentation d'images est relativement récent, et constitue un sujet de recherche encore en développement. La plus part de ses méthodes exploitent les approches de segmentation des images en niveau de gris en les appliquant à des différents espaces couleurs, les résultats de ces méthodes peuvent varier selon l'espace colorimétrique utilisée.

Quelles que soit les méthodes utilisées, lorsqu'on travaille sur la notion de couleur, il se pose le problème de la dimension des données. On trouve par conséquent trois types d'approches de la segmentation d'images couleur : l'approche marginale, l'approche scalaire, et l'approche vectorielle. Le principal désavantage des approches marginales et scalaires est la non prise en compte des interactions colorimétriques inter composantes. Dans notre travail on usera de l'approche vectorielle.

Nous allons introduire cette notion de couleur dans la segmentation en utilisant l'approche par classification des pixels. Comme les réseaux de neurones apportent un gain en efficacité, précision et rapidité dans le traitement des données, nous avons opté pour leur utilisation.

Nous allons segmenter des images réelles dont l'appartenance des pixels aux différentes classes est inconnue. De ce fait nous utiliserons les cartes topologiques

auto-organisatrices, qui sont une famille des réseaux de neurones à apprentissage non supervisé, après organisation de la carte, on appliquera une classification hiérarchique ascendante afin de regrouper les neurones en nombres de classes souhaitées. L'affectation aux différentes classes se fera en suite par rapport aux valeurs de ces neurones

Pour mener a bien notre travail, nous l'avons organisé en quatre chapitres :

- Le premier chapitre, traitera des notions générales des images couleur.
- Le second chapitre, sera consacré à l'étude des réseaux de neurones.
- Le troisième chapitre portera sur l'étude de la carte auto-organisatrice de Kohonen
- Les tests effectués, les résultats obtenus ainsi que les différents commentaires seront présentés dans le quatrième chapitre.

Et enfin, nous terminerons notre travail par une conclusion générale.

I.1. Introduction

Dans ce chapitre nous définirons l'image et donnerons quelques notions sommaires de celles-ci. Nous développerons ensuite les notions relatives à l'image couleur et aux différents systèmes de sa représentation. Nous expliquerons aussi les méthodes utilisées pour traiter ces images couleur et aussi les diverses méthodes de segmentation qui lui sont appliquées.

I.2. Définition de l'image [1]

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film etc...

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

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés pixels, ayant chacun comme caractéristique un niveau de gris ou de couleur prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter. C'est donc une matrice bidimensionnelle de valeurs numériques $I(x,y)$, où x et y , représentent les coordonnées cartésiennes d'un point de l'image et $I(x,y)$ correspond au niveau de gris ou de couleur de ce point dont la dimension est de $M*N$ éléments, tel que, M est le nombre de lignes et N le nombre de colonnes dans l'image.

La quantité d'informations que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, généralement, chaque pixel est codé sur un octet, dans ce cas, on parle d'image à niveau de gris. Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Un exemple d'images en niveau de gris est donné par la figure I.1.



Figure I.1 Exemple d'images a niveaux de gris

Même s'il est parfois utile de pouvoir représenter des images en niveaux de gris, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.), et en anglais RGB (pour Red Green Blue). Ainsi dans le cas de ces images, chaque pixel est généralement codé sur trois octets (24 bits), un octet pour chacune de ces couleurs primaires. Un mélange quelconque de ces couleurs pures engendre une autre couleur, qu'on peut qualifier en termes de nuances selon différents critères plus ou moins intuitifs. La figure I.2 illustre un exemple d'images couleur.



Figure I.2 Exemple d'images couleur

I.3. La couleur et la colorimétrie [2]

Le système visuel humain ne peut détecter, dans le spectre de la lumière, que des longueurs d'ondes comprises entre environ 400 et 700 nanomètres. En dessous de ces limites on parle de l'ultraviolet, au dessus, de l'infrarouge. Notre système visuel perçoit cet intervalle de fréquences d'ondes lumineuses comme un arc-en-ciel de couleurs variant progressivement. On appelle cet intervalle de fréquences d'ondes lumineuses le spectre visible. L'illustration qui suit montre approximativement le spectre visible.

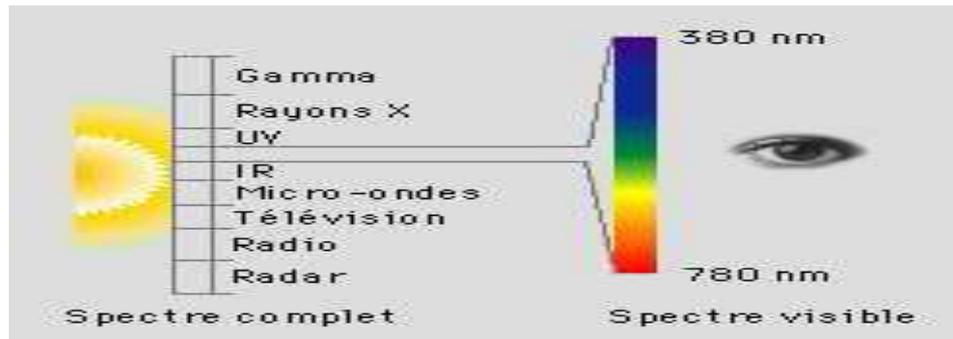


Figure I.3 : Le spectre visible dans le spectre complet

Une couleur est donc définie par sa longueur d'onde, ou par un mélange de longueurs d'onde. Par exemple, un vert "pur" est une radiation monochromatique de longueur d'onde 530 nm, tandis que la lumière blanche est un spectre continu contenant toutes les longueurs d'ondes du domaine du visible.

Les couleurs trouvent donc leurs origines dans la séparation de la lumière blanche naturelle en composantes absorbées et composantes réfléchies. La couleur d'un objet 'uni' est alors définie et mesurée par le spectre de réflectance ou de transmittance. Un même objet peut avoir plusieurs rendus couleurs selon que l'on modifie la source d'éclairage, la géométrie du dispositif d'éclairage, de mesure, ou selon que l'on change telle ou telle caractéristique physique de cet objet. Suivant l'état de surface, la structure chimique, la nature physique d'un objet, ses caractéristiques optiques peuvent changer du tout au tout.

On appelle stimulus le signal résultant du produit du spectre d'une source avec le spectre de réflectance d'un objet. Autrement dit c'est le signal spectral qui atteint l'œil ou tout autre capteur.

La colorimétrie a pour but d'associer à chaque stimulus visuel élémentaire (correspondant à une zone définie du champ visuel, placée sous un éclairage uniforme sur un fond de couleur neutre n'ayant pas de texture apparente), une mesure de stimulus couleur. Elle se base sur la recherche de différence entre stimuli. Elle est fondée sur la théorie

trichromatique qui statue que la plus part des stimuli peuvent être égalisés par la synthèse additive de trois primaires fixes et que si un stimulus ne peut être apparié avec ces trois primaires, il peut l'être en le combinant avec l'une d'entre elles.

Il existe plusieurs systèmes de représentation numérique de l'image couleur. En référence à la théorie trichromatique découlant du système visuel humain, on considère, de manière générale, que la couleur se définit selon trois composantes qui suivant les systèmes de couleur considérés conduisent à quatre familles de systèmes de représentations à savoir :

- Les systèmes de primaires,
- Les systèmes luminances-chrominance
- Les systèmes perceptuels
- Les systèmes d'axes indépendants

I.4. Les systèmes primaires [3]

Le principe de la trivariance visuelle repose sur une théorie selon la quelle toute couleur peut être reproduite visuellement à l'identique, dans des conditions d'observation déterminées, et ce par le mélange algébrique, en proportions définies de manière unique, de trois couleurs appelées primaires. Ces trois couleurs sont arbitrairement choisies sous la seule réserve qu'aucune d'entre elle ne puisse être reproduite par un mélange approprié des deux autres. Cette théorie constitue l'élément de base de la théorie trichromatique, qui à été validée grâce à des expériences d'égalisation de couleur.

L'ensemble des couleurs reproductibles par ce principe constitue l'espace des couleurs, lequel est un espace tridimensionnel pour lequel les trois primaires P_1, P_2 et P_3 représentent trois vecteurs non coplanaires qui constituent les vecteurs directeurs de base. Ainsi toute couleur C peut être exprimée, abstraction faite de l'aspect lumineux, dans cette base selon l'équation suivante :

$$C = p_1 P_1 + p_2 P_2 + p_3 P_3 \quad (\text{I.1})$$

Ce principe est à l'origine de tous les systèmes de représentation de la couleur, dont les espaces de représentation RVB.

I.4.1. Le système RVB (Rouge Vert Bleu) (RGB : Red Green Blue)

En traitement d'images, l'espace RGB reste de loin le plus utilisé et le plus simple à manipuler. Cependant, la définition de l'espace de représentation de la couleur qui dépend des primaires et du blanc de référence n'est pas unique. Comme le souligne Vandembroucke, de

nombreux systèmes RGB ont été définis par différents organismes et sont utilisés dans d'autres domaines d'application que la colorimétrie [3].

Parmi les systèmes RGB définis, nous pouvons citer :

- La norme NTSC (National Television Standards Committee), dédiée à la télévision américaine,
- La norme PAL (Phase Alternation by Line) élaborée par la FCC (Federal Communications Commission) pour la télévision européenne
- La norme SECAM (SEquentiel Couleur A Mémoire) utilisée par la télévision française.

Le plus utilisé des espaces RGB est celui introduit par la CIE (Commission internationale de l'Eclairage) en 1931. Celui-ci est défini à partir de trois primaires monochromatiques de couleurs rouge, verte et bleue. Les longueurs d'ondes associées à chacune de ces primaires sont 700.0 nm pour le rouge, 546.1 nm pour le vert et 435.8 nm pour le bleu. Il traduit le comportement d'un observateur de référence qui est vérifié par 90% des observateurs. Les fonctions colorimétriques de cet observateur dit standard sont données par la figure I.4.

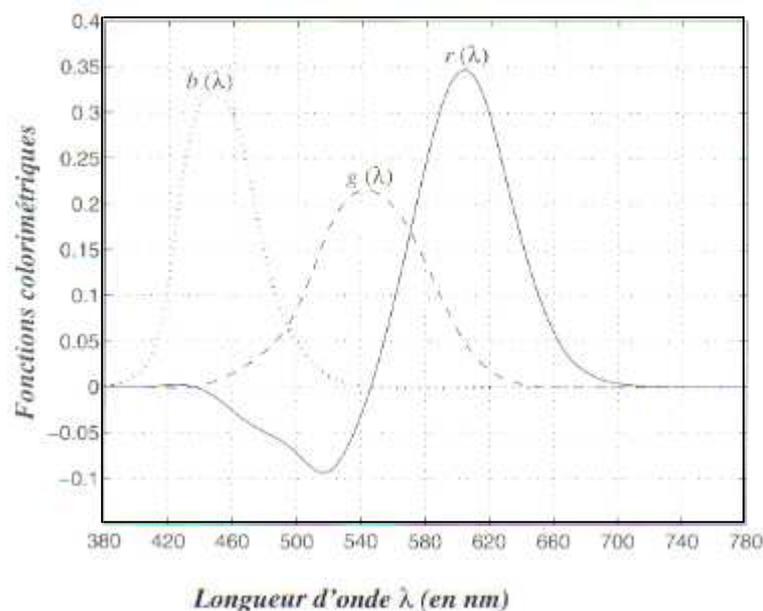


Figure I.4 : Fonctions colorimétriques de l'observateur du système RGB CIE1931

La figure I.4 définit le domaine des couleurs appariables par synthèse additive des primaires rouge, verte et bleue. Toutes les couleurs situées dans le triangle (appelé triangle de Maxwell) reliant les trois primaires ont toutes la même valeur de luminance. Toutes les couleurs situées dans un plan parallèle à ce triangle ont également toutes la même luminance.

Perpendiculairement à ce triangle, la diagonale qui relie le noir au blanc définit l'axe achromatique.

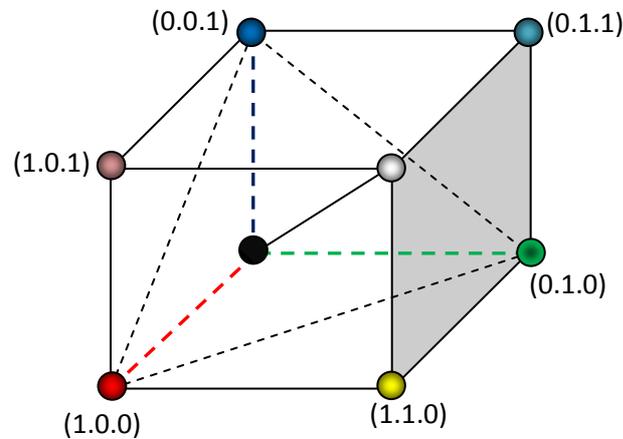


Figure I. 5 : Cube des couleurs RGB

L'espace RGB défini par la CIE 1931 présente quelques inconvénients comme l'existence d'une partie négative dans les spectres et par conséquent, l'impossibilité de reproduire un certain nombre de couleurs par superposition des trois spectres (ce qui contredit le principe de la synthèse additive). Afin de pallier cet inconvénient, la CIE a défini un espace de représentation de la couleur basée sur trois primaires non visibles X, Y et Z.

I.4.2. L'espace XYZ [3]

L'espace couleurs XYZ a été défini afin de corriger certains défauts de l'espace RGB. Cet espace résulte des travaux de Judd et est constitué de trois primaires X, Y et Z, dites virtuelles. Il présente les propriétés suivantes :

- Les triplets décrivent chaque couleur en fonction de ses primaires qui ont tous des valeurs positives pour les spectres visibles (voir figure I.6),
- la fonction $Y(\lambda)$ représente approximativement la sensibilité de l'œil humain à la luminosité. Par conséquent, la composante Y est usuellement considérée comme la composante luminance du spectre incident,
- tout spectre d'égale énergie est associé à un triplet dont toutes les composantes sont égales.

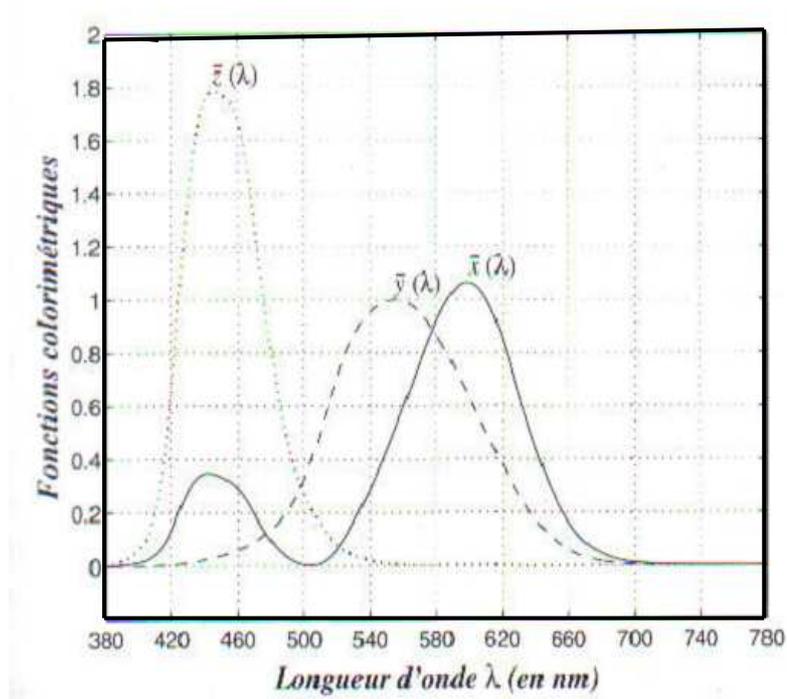


Figure I.6 : Fonction colorimétriques du système XYZ de l'observateur de référence CIE1931

Le passage de l'espace RGB à l'espace XYZ s'effectue simplement grâce à une transformation linéaire pouvant être interprétée comme un changement de base comme le donne l'équation I.2 :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = A * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (\text{I.2})$$

où A est une matrice 3*3, dite de passage, ayant la forme suivante :

$$A = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \quad (\text{I.3})$$

Sachant que les coefficients de la matrice A sont directement liés au blanc de référence choisi. Ainsi, les valeurs des primaires X, Y et Z sont conditionnées par ce blanc de référence en plus des primaires R, G et B. Par conséquent, plusieurs matrices de passage ont vu le jour suite aux travaux de différents auteurs.

Le passage du système de primaires RGB CIE 1931 au système de primaires XYZ CIE 1931 est donné par l'équation :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2.7690 & 1.7518 & 1.1300 \\ 1.0000 & 4.5907 & 0.0601 \\ 0.0000 & 0.0565 & 5.5943 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (\text{I.4})$$

Afin de pouvoir représenter et analyser assez simplement, dans un repère géométrique à 2 dimensions les coordonnées X, Y et Z, on utilise en général les coordonnées normalisées x, y et z suivantes :

$$\begin{cases} x = \frac{X}{X+Y+Z} \\ y = \frac{Y}{X+Y+Z} \\ z = \frac{Z}{X+Y+Z} \end{cases} \quad (\text{I.5})$$

Etant donné que la somme des trois coordonnées vaut 1, z se déduit directement des deux autres, ceci justifie pourquoi cette coordonnée n'est généralement pas utilisée.

L'espace xyY est communément utilisé pour représenter l'espace trichromatique. Il est le plus généralement décrit par une représentation 2D à Y constant dans un diagramme appelé «Spectrum locus». Une représentation à 3D est toujours possible. (Voir figure I.7)

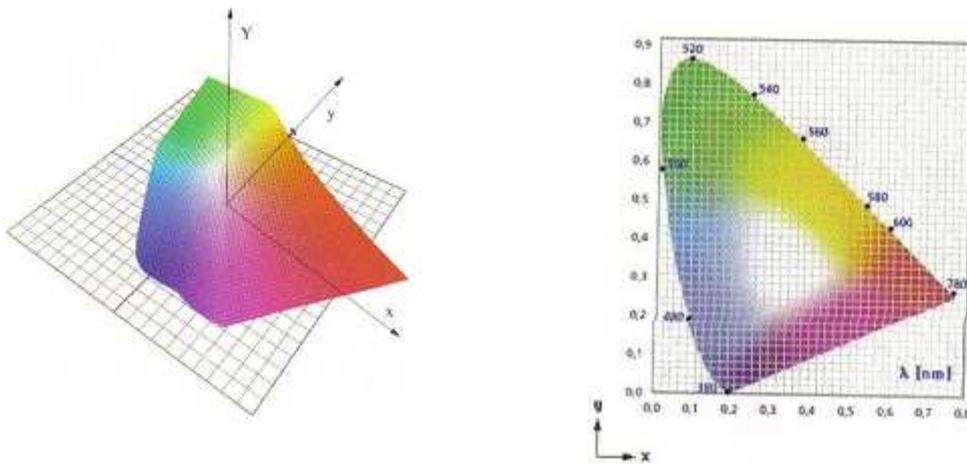


Figure I.7 : représentation du diagramme de chromaticité xyY

I.4.3. L'espace CMJ

L'espace CMJ est l'espace dédié à l'impression des couleurs. Il résulte d'une synthèse soustractive et est représenté par un cube comme l'espace RGB. Mais à l'inverse de ce dernier, le cube CMJ a le blanc pour origine et le cyan, le magenta et le jaune comme axes. Le passage du système de primaires RGB CIE 1931 au système CMJ est donné par l'équation :

$$\begin{pmatrix} C \\ M \\ J \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (\text{I.6})$$

I.5. Les systèmes luminance-chrominance

L'intérêt des espaces de type luminance-chrominance est qu'ils dissocient la composante de luminance des composantes de chrominance. De nombreux espaces de représentation couleur se rattachent à cette famille de système, dont notamment les espaces de type Luminance-Teinte-Saturation (LTS). Ces espaces se différencient essentiellement par la façon dont sont calculées les coordonnées de luminance et de chrominance.

I.5.1. Le système YC_bC_r

Ce système a été à l'origine développé afin d'assurer une compatibilité entre les téléviseurs couleurs et les téléviseurs noir et blanc, d'où la séparation des composantes de luminances et de chrominances. Une simple transformation linéaire permet de passer d'un système RVB au système YC_bC_r , notons toutefois que suivant les standards de télévisions (NTSC, PAL ou SECAM) considérés, cette transformation n'utilise pas les mêmes coefficients.

La première opération permet de calculer les valeurs de chrominances C_b et C_r d'une couleur exprimée en coordonnées RVB, en fonction des coefficients a_1, a_2, b_1, b_2 , spécifique au standard considérée de la luminance Y , selon le principe suivant :

$$\begin{cases} C_b = a_1(R - Y) + b_1(B - Y) \\ C_r = a_2(R - Y) + b_2(B - Y) \end{cases} \quad (I.7)$$

A cette opération s'ajoute une correction gamma (γ) qui s'applique non seulement aux composantes de chrominances, mais également à la composante de luminance. En effet, étant donné que pour chacune des primaires d'un téléviseur, le signal émis suit une loi x^γ , où x représente le signal transmis, il faut corriger le signal selon la loi inverse $x^{\frac{1}{\gamma}}$ pour pouvoir obtenir un signal émis proportionnel au signal transmis. Suivant le standard considéré la correction gamma, est plus ou moins forte.

Il existe plusieurs systèmes de type YC_bC_r . Ainsi le système YIQ correspond à la de la norme NTSC, le système YUV correspond à la de la norme PAL et le système YD_bD_r correspond à la de la norme SECAM.

La transformation qui permet de passer du système NTSC RVB au système YIQ est une transformation qui se décline en deux opérations données par les équations I.8 et I.9.

$$\begin{bmatrix} R' \\ V' \\ B' \end{bmatrix} = \begin{bmatrix} R_{2,2}^{\frac{1}{2}} \\ V_{2,2}^{\frac{1}{2}} \\ B_{2,2}^{\frac{1}{2}} \end{bmatrix} \quad (\text{I.8})$$

avec $R'V'B'$ qui représentent les coordonnées gamma corrigés des coordonnées NTSC RVB.

$$\begin{aligned} Y &= 0,299R' + 0,587V' + 0,114B' \\ I &= 0,74(R' - Y) - 0,27(B' - Y) \\ Q &= 0,48(R' - Y) + 0,41(B' - Y) \end{aligned} \quad (\text{I.9})$$

Cette seconde opération peut également se décrire à partir du système d'équation suivant :

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,273 & -0,322 \\ 0,212 & -0,522 & 0,315 \end{bmatrix} \begin{bmatrix} R' \\ V' \\ B' \end{bmatrix} \quad (\text{I.10})$$

Cet espace peut être décrit en coordonnées cylindriques de la façon suivante :

$$\begin{aligned} H &= \arctan\left(\frac{(B'-Y)}{(R'-Y)}\right) \\ C &= \left((R' - Y)^2 + (B' - Y)^2\right)^{\frac{1}{2}} \end{aligned} \quad (\text{I.11})$$

avec H qui est la teinte et C la saturation. On obtient alors un système de type Luminance Teinte Saturation (LTS).

I.4.2. Les systèmes antagonistes

Selon la théorie des couleurs opposées de Young et de Hering l'information couleur captée par le système visuel est transmise au cerveau sous la forme de trois signaux, un achromatique qui correspond à l'opposition de couleurs noir-blanc, et deux chromatiques l'un correspondant à l'opposition de couleurs vert-rouge et l'autre correspondant à l'opposition des couleurs bleu-jaune [4].

Plusieurs systèmes appartiennent à cette famille, mais le plus communément utilisé est le système AC_1C_2 . Il est basé sur le système d'équations suivant :

$$\begin{aligned} A &= a(\alpha \log(L) + \beta \log(M) + \gamma \log(S)) \\ C_{1=} &= \mu_1(\log(L) - \log(M)) \\ C_2 &= (\mu_2(\log(L) - \log(S))) \end{aligned} \quad (\text{I.12})$$

Ce système repose sur un système de primaire de type RVB pour lequel la primaire L est d'une longueur d'onde dominante monochromatique à 570nm, la primaire M d'une longueur

d'onde dominante monochromatique à 540nm et la primaire S d'une longueur d'onde dominante monochromatique à 445nm.

Les coefficients de ce système d'équations varient suivant les modèles perceptuels proposés. Par exemple dans le modèle de Faugeras ces coefficients ont pour valeurs : $a = 22,6$; $\alpha = 0,612$; $\beta = 0,369$; $\gamma = 0,019$; $\mu_1 = 64$; $\mu_2 = 10$. [5]

L'espace AC_1C_2 peut tout aussi être décrit selon un système de coordonnées cylindriques. On obtient alors un système de type LTS.

La plupart des autres systèmes appartenant à la famille des systèmes antagonistes ne sont que peu utilisées en imagerie couleur car ce sont soit des systèmes expérimentaux, soit des systèmes très simplifiés

I.6 Les systèmes perceptuels [3]

Sont regroupé dans cette famille de systèmes, les systèmes uniformes au sens de la perception visuelle. Plus précisément, la spécificité de ces systèmes est de pouvoir décrire fidèlement, au sens de la perception visuelle, les écarts couleur entre couleurs proches.

I.6.1. L'espace $L^*u^*v^*$

Cet espace est devenu un standard de la CIE en 1976. Le blanc de référence utilisée est caractérisé par les trois composantes trichromatiques ($X_0Y_0Z_0$) prises dans l'espace XYZ. Le passage des coordonnées XYZ vers les coordonnées $L^*u^*v^*$ est effectué grâce aux formules suivantes :

$$L^* = \begin{cases} 116 * \left(\frac{Y}{Y_0}\right)^{\frac{1}{3}} - 16 & \text{si } \frac{Y}{Y_0} > 0.008856 \\ 903.3 * \frac{Y}{Y_0} & \text{si } \frac{Y}{Y_0} \leq 0.008856 \end{cases} \quad (\text{I.13})$$

$$u^* = 13L^* (u' - u'_0)$$

$$v^* = 13L^* (v' - v'_0)$$

où les quantités u' , v' , u'_0 , v'_0 sont calculées comme suit :

$$\begin{aligned} u' &= \frac{4X}{X+15Y+3Z} \\ v' &= \frac{9Y}{X+15Y+3Z} \\ u'_0 &= \frac{4X_0}{X_0+15Y_0+3Z_0} \\ v'_0 &= \frac{9Y_0}{X_0+15Y_0+3Z_0} \end{aligned} \quad (\text{I.14})$$

Le fait de représenter la luminosité par une racine cubique a été démontré par les expériences psycho-visuelles menées sur l'appréciation des distances entre des stimuli monochromatiques. De plus, le seuil de 0.008856 est choisi afin de supprimer le caractère trop abrupt des variations de $(Y/Y_0)^{1/3}$ autour de 0. Cette racine cubique est néanmoins très intéressante pour le rapprochement du fonctionnement de l'œil humain. En effet, l'introduction du rapport Y/Y_0 permet de simuler grossièrement l'adaptation de l'œil humain à une luminosité donnée. Dans le cas de faibles luminances (vision de nuit par exemple), la pente de la courbe est assez importante marquant l'intervention des bâtonnets. Au contraire, un effet de saturation peut être observé pour les fortes luminances du fait de la saturation des récepteurs de l'œil.

Notons en dernier que la luminance, dans les termes de la CIE, représente la clarté et les composantes u^* et v^* représentent respectivement l'opposition de couleurs vert-rouge et l'opposition de couleur bleu-jaune.

I.6.2. L'espace CIE $L^*a^*b^*$

Dans la même année, 1976 en l'occurrence, la CIE a introduit un autre système perpétuellement uniforme, le système $L^*a^*b^*$. Celui-ci s'obtient grâce à des relations non linéaires à partir du système XYZ. La composante L^* est calculée de la même façon que dans le modèle de $L^*u^*v^*$. Les composantes chromatiques de cet espace sont données par les équations suivantes :

$$\begin{aligned} a^* &= 500 \left[f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \right] \\ b^* &= 300 \left[f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right) \right] \end{aligned} \quad (\text{I.15})$$

avec :

$$f(x) = \begin{cases} x^{\frac{1}{3}} & \text{si } x > 0.008856 \\ 7.787x + \frac{16}{116} & \text{si } x \leq 0.008856 \end{cases} \quad (\text{I.16})$$

L'espace $L^*a^*b^*$ appartient également à la famille des systèmes antagonistes puisque L^* met en évidence l'opposition des couleurs noir-blanc, a^* met en évidence l'opposition des couleurs vert-rouge, et b^* l'opposition des couleurs bleu-jaune. De même cet espace appartient également à la famille des systèmes Luminance-Chrominance, puisque L^* décrit la luminance et a^* et b^* décrivent la chrominance.

1.6.3. Les représentations géométriques

Plusieurs systèmes de représentation de la couleur, sont basés sur un partitionnement uniforme d'un espace couleur de type LTS (Luminance, Teinte, Saturation). La principale différence, entre un système et un autre, provient des unités et de la dynamique de chacun des trois axes couleur. Ces représentations ont été conçus afin de repérer les couleurs à partir des coordonnées géométriques, l'objectif étant de classer les couleurs selon un principe basé sur la perception psycho-visuelle. Ces représentations doivent être considérées avant tout comme des représentations descriptives.

Il existe deux familles de systèmes :

- Les systèmes géométriques (HSL, HSV, TLS, LCH,...) qui partitionnent l'espace couleur selon un espace de discrétisation arbitrairement fixé. A titre d'exemple, quelques systèmes de ce type sont donnés ci dessous. Tous sont basés sur des grandeurs de type Luminances, Teinte et Saturation, mais les modes de calculs sont différents d'un espace à un autre.
- Les systèmes uniformes (Munsell, NCS (Natural Color System), OSA (Optical Society of America),...), qui partitionnent l'espace couleur selon un pas de discrétisation qui s'apparente au seuil différentiel du système visuel humain.

Le système de coordonnées triangulaires **HSI** est un modèle de représentation de la couleur communément utilisé en traitement d'images couleurs. Sa modélisation est issue de la déformation du cube des couleurs RGB. En effet, il suffit d'imaginer le renversement du cube sur le coin représentant le noir. Ainsi, l'axe achromatique constituera l'axe des intensités I et la couleur sera définie par une position sur un palier circulaire où la saturation S représente le rayon et la teinte H représente l'angle.

Les formules exprimant la transformation de l'espace RGB à l'espace HSI sont données par :

$$\left\{ \begin{array}{l} I = \frac{R+G+B}{3} \\ S = 1 - \frac{3 \cdot \min(R,G,B)}{R+G+B} \\ H = \arccos \left(\frac{0.5 \cdot (R-G) + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right) \end{array} \right. \quad (I.17)$$

Le système **HSV**, connu aussi sous le nom du système de cône hexagonal, a été défini par Travis []. Deux modèles peuvent être distingués : le modèle de cône hexagonal simple et le modèle hexagonal double (figure I.8). Ils représentent la couleur sous forme d'un triplet : Teinte H (Hue), Saturation S et luminosité V (Value).

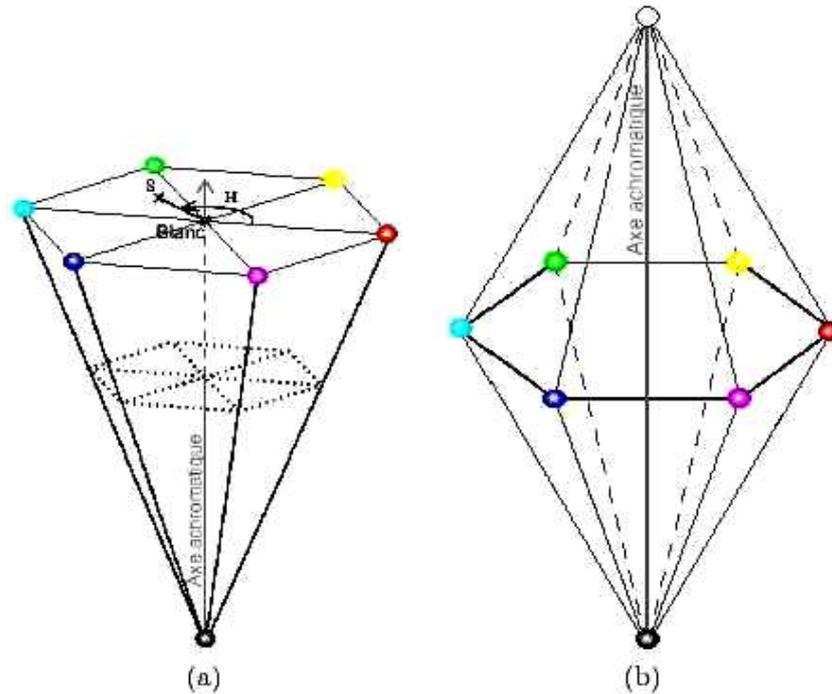


Figure I.8 : Modèle de cône hexagonal de Travis
 a - modèle de cône hexagonal simple
 b - modèle de cône hexagonal double dit aussi HSL

Dans le cas du modèle simple les transformations sont effectuées comme suit :

$$\begin{array}{l}
 \text{-----} \\
 \text{-----} \\
 \text{-----} \\
 \text{-----}
 \end{array}
 \tag{I.18}$$

Le modèle de cône hexagonal double est construit sur le même principe que le modèle précédent. Sachant que la formule pour calculer la teinte est la même que l'équation de H, la valeur et la saturation sont calculées comme suit :

$$\begin{array}{l}
 \text{-----} \quad \text{---} \\
 \text{-----} \quad \text{---} \\
 \text{-----}
 \end{array}
 \tag{I.19}$$

I.7. Les systèmes d'axes indépendants

Suivant la distribution des couleurs considérées, autrement dit suivant l'image traité et du système couleur utilisé, on peut observer une corrélation plus au moins forte entre les différentes composantes couleurs. Pour peu que cette corrélation soit relativement forte, il devient alors impossible de traiter indépendamment chacune des composantes couleurs du système considérés, sauf à perdre l'information commune à plusieurs composantes. Pour pallier ce problème il convient alors de passer dans un autre système couleur décorrélé. On note en générale $X_1X_2X_3$ cet espace couleur. Pour trouver ce système décorrélé, on utilise par exemple la transformée de Karhunen-Loeve. Le problème de cette transformée est qu'elle est couteuse en temps de traitement. Selon Ohta et al [6] elle converge pour la plupart des images couleur vers le système d'axes noté $I_1I_2I_3$. Ainsi, au lieu de calculer la transformée de Karhunen-Loeve, on utilise une transformation linéaire à partir de l'espace RGB définie par les formules suivantes :

$$\begin{aligned} I_1 &= \frac{R+G+B}{3} \\ I_2 &= \frac{R-B}{2} \\ I_3 &= \frac{2G-B-R}{4} \end{aligned} \quad (I.20)$$

Où I_1 correspond à la composante de luminance. I_2 et I_3 représentent respectivement les oppositions bleu-rouge et magenta-vert.

I.8. Les systèmes hybrides

Plutôt que de représenter les points de couleurs dans l'un des espaces couleur précédemment présentés, une autre approche consiste à retenir parmi K composantes couleurs présélectionnées, par exemple, les composantes : $R, V, B, X, Y, Z, L, H, S, V, L^*, a^*, b^*, I_1, I_2, I_3, C_1, C_2$, les trois composantes couleur les plus pertinentes par rapport à un cadre d'étude donné. Ces trois composantes couleur forment alors un espace de couleur hybride.

Afin de déterminer les composantes couleurs les plus discriminantes, différentes techniques peuvent être utilisées. A titre d'exemple Vandembrouke [7] utilise une technique d'apprentissage supervisé et Lezoray [8] utilise une technique basée sur un algorithme génétique associé à un réseau de neurones.

Dans l'absolue il est d'autant plus facile de réduire, de K à trois, le nombre de dimension d'études pertinente que :

- d'une part tous les espaces couleur sont liés les un aux autres, soit par des transformations linéaires, soit par des transformations non linéaires,
- d'autre part que tous ces espaces décrivent l'information couleur dans un repère à trois dimensions

Cela étant, étant donné qu'il ya une forte redondance entre les composantes couleur présélectionnées, et une forte corrélation entre les dimensions couleurs, il est assez difficile de définir des critères de sélection qui prennent en compte tous ces aspects.

I.9. Traitement d'image couleur [1]

Pour manipuler une image, on travaille sur le tableau d'entiers qui contient les composantes de chaque pixel. Un système de traitement d'image peut être représenté comme suit :

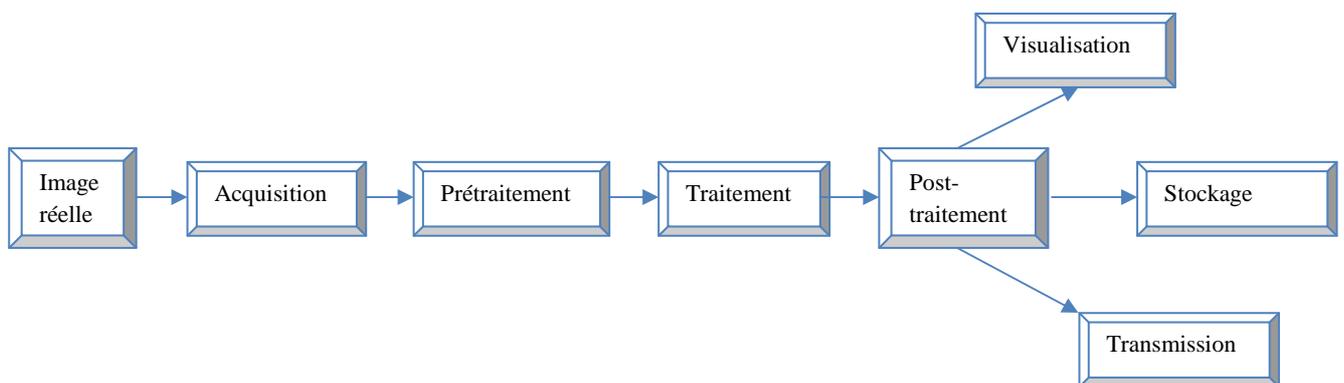


Figure I.9 : Système de traitement d'image

L'acquisition d'une image est l'opération qui permet le passage de l'information réelle à une représentation numérique.

L'opération de prétraitement a pour but principal la correction des imperfections du système d'acquisition. Les méthodes développées pour le prétraitement des images diffèrent selon la technique utilisée, les hypothèses sur le modèle du bruit, la nature de l'image à traiter et surtout le but recherché par le traitement.

Le post-traitement concerne les images traitées. Toutes les opérations utilisées en prétraitement, le sont aussi en post-traitement.

Le traitement d'image est l'ensemble, des méthodes qui permettent de décrire quantitativement le contenu d'une image. Les traitements applicables aux images sont nombreux et souvent sont en fonction du domaine d'application (télécommunication, la

médecine, l'imprimerie, etc.). Les traitements les plus utilisés sont la convolution et la segmentation. La convolution est le remplacement de la valeur d'un pixel par une combinaison linéaire de ses voisins. Elle consiste à faire balayer une fenêtre (masque) sur l'ensemble des points de l'image. La segmentation consiste à partitionner l'image en régions homogènes connexes. Tout système d'image est doté d'un dispositif de visualisation qui permet l'affichage des images. L'image numérisée et traitée peut être archivée en la stockant sur des supports destinés à cet effet comme par exemple les bandes magnétiques, les disques durs...

I.10. Stratégies de traitement d'images couleur [3]

On peut distinguer quatre principaux types de stratégies dans la construction des traitements des images couleur :

- les stratégies scalaires
- les stratégies marginales
- les stratégies vectorielles
- les stratégies hybrides

I.10.1. Stratégie scalaire

L'approche scalaire de la segmentation fusionne tout d'abord les composantes couleurs en une seule et traite celle-ci. Le passage d'une image couleur en une image monochrome peut se faire par exemple par la transformée de Karhunen-Loeve ou par une quantification. On se ramène alors à un problème de traitement scalaire (en niveau de gris) ou tout l'arsenal des méthodes de traitement des images monochromes peut être utilisé. Cette stratégie est illustrée à la figure I.10 dans le cas d'une image à trois composantes R,V,B.

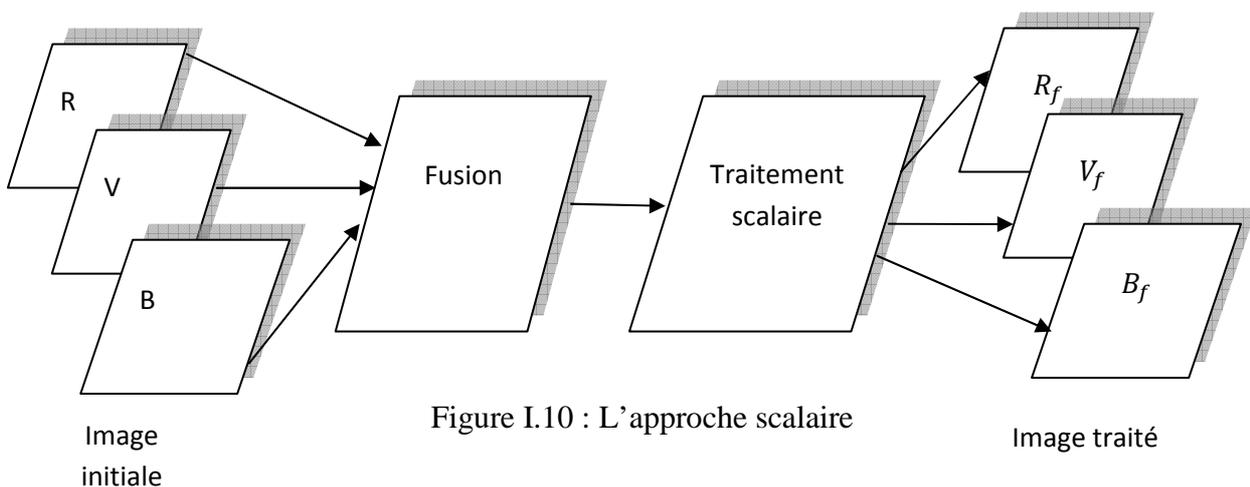


Figure I.10 : L'approche scalaire

I.10.2 Stratégie marginale

Cette stratégie consiste à traiter chaque composante séparément en utilisant des traitements mono-composants, empruntés à l'imagerie monochrome. La figure I.11 illustre cette stratégie dans le cas d'une image à trois composantes, Rouge, Vert, Bleu ou chaque composante est traité scalairement pour obtenir R_f , V_f , B_f .

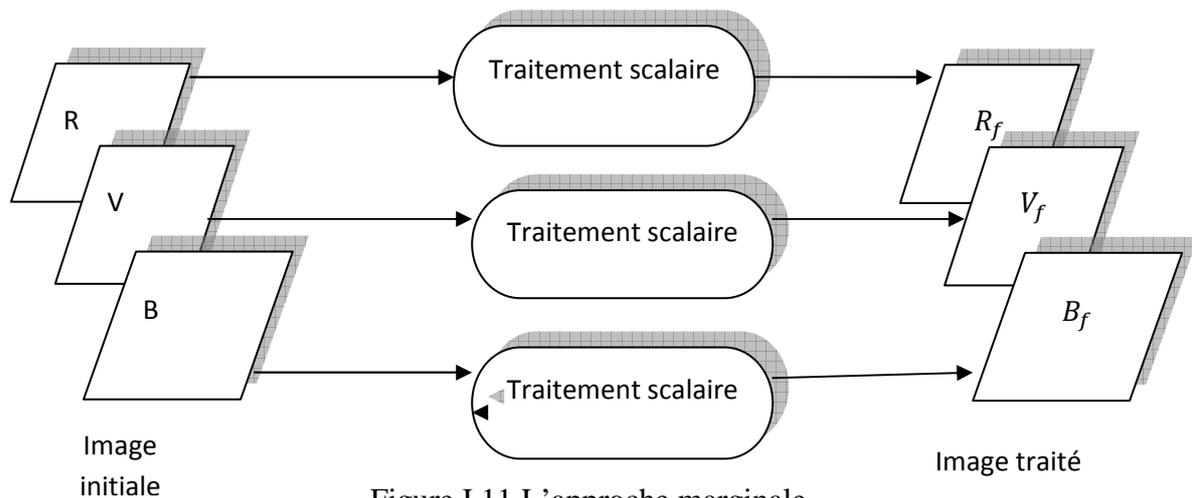


Figure I.11 L'approche marginale

I.10.3. Stratégie vectorielle

Cette stratégie traite de manière globale toute les composantes, elle est à priori plus satisfaisante du point de vue de la prise en compte du contexte multi-composantes. Cette approche présente aussi l'intérêt de ne nécessiter qu'un seul traitement, quel que soit le nombre de composantes. Cet avantage doit être modéré par une complexité accrue de ce seul traitement. La figure I.12 illustre cette stratégie dans le cas d'une image à trois composantes, R, V, B.

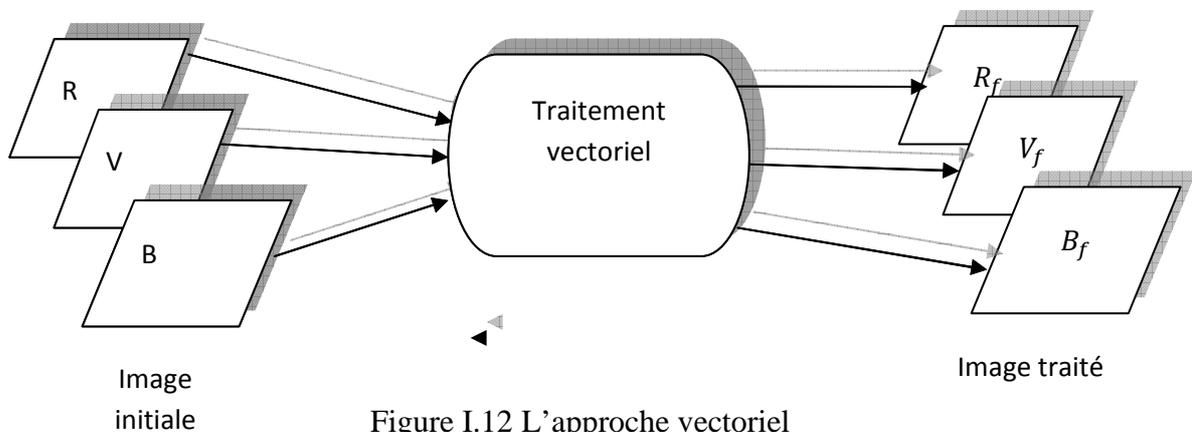


Figure I.12 L'approche vectoriel

I.10.4. Stratégie hybride

Une solution intermédiaire consiste, en gardant la structure marginale à définir les traitements en tenant compte de l'ensemble des composantes

Elle est typiquement utilisée pour les traitements adaptatifs. L'analyse locale, au voisinage du pixel courant, de l'ensemble des composantes permet de décider de la nature des traitements qui vont ensuite être appliqués à ce pixel composante par composante.

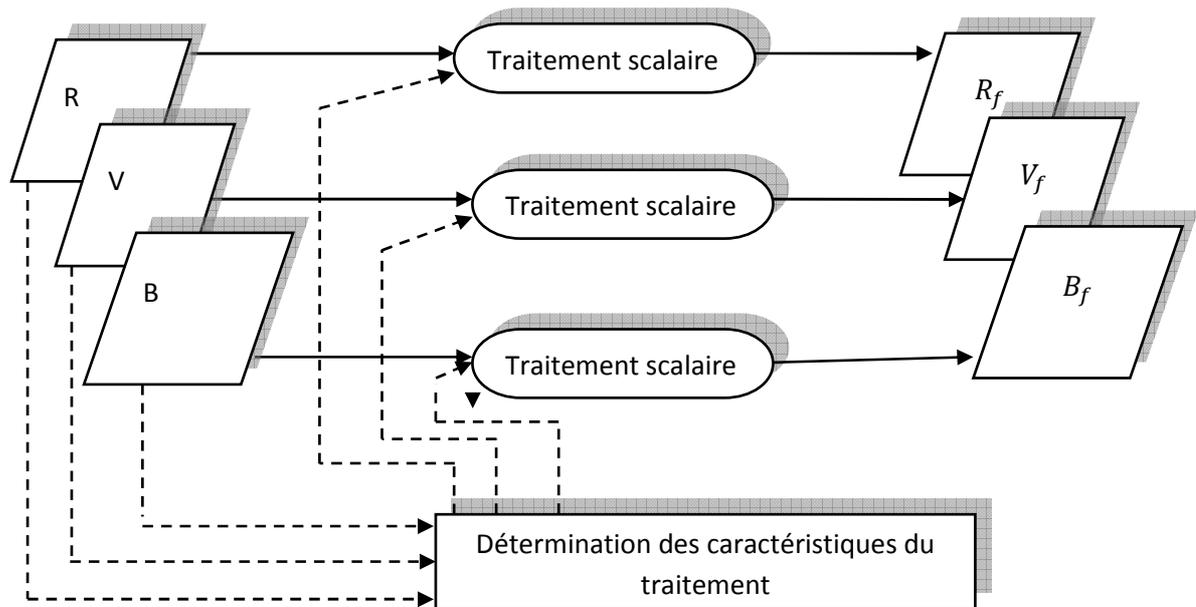


Image
initiale

Figure I.13 : L'approche hybride

Image traité

I.11. La segmentation d'images couleur [9]

La segmentation consiste à créer des partitions dans l'image en identifiant des pixels similaires. Le critère de similarité permet de contrôler l'aspect final de la segmentation (regroupement par couleur, par homogénéité, par taille) et elle a pour objectif de différencier des zones d'intérêt (par exemple objets / fond).

La segmentation consiste donc à classer les pixels de l'image de façon à ce que chaque classe soit homogène sous contrainte qu'il n'existe pas deux classes adjacentes dont l'union donne une région homogène. La définition formelle de la segmentation d'image en s'appuyant sur la partition d'un ensemble I en n sous ensembles R_1, R_2, \dots, R_n est tels que :

- $R_i \cup R_i = I$, pour $i = 1, \dots, n$.
- $R_i \cap R_j = \emptyset$ avec ($i \neq j$)
- Les sous ensembles R_i , $i = 1, \dots, n$ sont connexes.
- Il existe un prédicat P tel que :

- $P(R_i) = \text{vrai}, \forall i = 1, \dots, n.$
- $P(R_i \cup R_j) = \text{faux}, \forall (i, j), i \neq j$ et R_i, R_j étant adjacent dans $I.$

La segmentation d'image a pour but de regrouper des pixels connexes ayant des attributs similaires, et cela afin de constituer des régions homogènes et cohérentes, ces régions étant en générale, liées a un objet ou a une partie d'objet de la scène. Dans les images couleurs, les attributs les plus utilisés sont issus de l'information achromatique et l'information chromatique. La plus part des méthodes de segmentation d'images couleurs exploitent les approches de segmentation des images en niveau de gris en les appliquant à des différents espaces couleurs, les résultats de ces méthodes peuvent varier selon l'espace colorimétrique utilisée. En générale ces méthodes peuvent être classées en deux famille selon qu'elles utilisent ou non la répartition spacio colorimétrique.

Les méthodes de construction de régions par analyse spatiale parcourent l'image afin de regrouper des pixels connexes respectant le prédicat utilisé pour construire les régions. Les méthodes de segmentation par classification de pixels dans l'espace couleur associent à chaque pixel un point couleur. Ces méthodes supposent que les pixels appartenant à une même région donnent naissance à un nuage de points couleur dans l'espace couleur. Chacun de ces nuages définit une classe de pixels de l'image. Les pixels sont alors assignés aux différentes classes construites. Une région est ainsi définie comme un ensemble de pixels connexes assignés a une même classe. Plusieurs régions peuvent donner naissance à une même classe ou seule une analyse en composantes connexe permet de les différencier.

II.12. Conclusion

Dans ce chapitre nous avons présenté d'une manière générale quelques notions essentielles sur l'image couleur ainsi que les différents espaces de représentation de celle-ci. L'image couleur étant à trois dimensions, son traitement et plus particulièrement la segmentation peuvent être réalisés de manière scalaire, marginale ou vectorielle dans les différents espaces de représentation.

L'une des stratégies possibles pour la segmentation est celle de la classification des pixels. Les réseaux de neurones se prêtent bien à une telle classification et feront l'objet de notre prochain chapitre.

II.1. Introduction

Le cerveau humain est un grand modèle dans le développement de l'intelligence artificielle, la biologie a apporté un grand nombre d'informations sur le fonctionnement de ce dernier. De simples enchevêtrements de neurones lui donnent toutes ses capacités, et c'est ce que tentent de reproduire les chercheurs.

Des mathématiciens ont alors tenté de reproduire le fonctionnement du cerveau en intégrant ces connaissances en biologie dans des programmes informatiques, et en leur donnant la possibilité d'apprendre. C'est ainsi que le premier réseau de neurones artificiel fût développé en 1943 par MC Culloch et Pitts. Des progrès ont depuis été réalisés. Nous allons dans ce chapitre présenter les principes de ces réseaux de neurones artificiels.

II.2. Historique

Les réseaux de neurones ont une histoire relativement jeune et les applications intéressantes des réseaux de neurones n'ont vu le jour qu'il à une vingtaine d'année avec le développement de l'informatique.

Le champ des réseaux neuronaux va donc démarrer par la présentation en 1943 par W. MC Culloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique. Le retentissement va être énorme. Par cette présentation, ils veulent démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques. La démonstration de Mc Culloch et Pitts sera un des facteurs importants de la création de la cybernétique.

En 1949, D. Hebb présente dans son ouvrage "The Organization of Behavior" une règle d'apprentissage et de nombreux modèles de réseaux aujourd'hui s'inspirent encore de cette règle là.

En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche lié à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience. Dans la même période, Le modèle de L'Adaline (ADaptive LINar Element) a été présenté par B. Widrow, chercheur américain à Stanford. Ce modèle sera par la suite le modèle de base des réseaux multicouches.

En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement

diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives, et propose des applications à la reconnaissance de formes.

C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines comme informatique, électronique, neurobiologie, et ce grâce à leurs différentes propriétés.

II.3. Définition [10]

Un réseau de neurones est un processus distribué d'une manière parallèle, qui a une prédisposition naturelle à mémoriser des connaissances de façon expérimentale et de les rendre disponibles pour l'utilisation. Il ressemble au cerveau en deux points :

- La connaissance est acquise au travers un processus d'apprentissage.
- Les poids des connexions entre les neurones sont utilisés pour mémoriser la connaissance.

C'est sur la base de cette définition que repose l'élaboration des réseaux de neurones artificiels.

II.4. Modèle d'un neurone

II.4.1. Le modèle neurophysiologique [10]

Le cerveau se compose d'environ 10^{12} neurones, avec 1000 à 10000 synapses (connexions) par neurone. Un neurone est caractérisé par un corps cellulaire (le soma), des extensions arborées qui recueillent les signaux électrochimiques (les dendrites) et un prolongement cellulaire (l'axone) qui est chargé de faire passer l'information nerveuse à d'autres cellules (au niveau des synapses) (voir figure II.1).

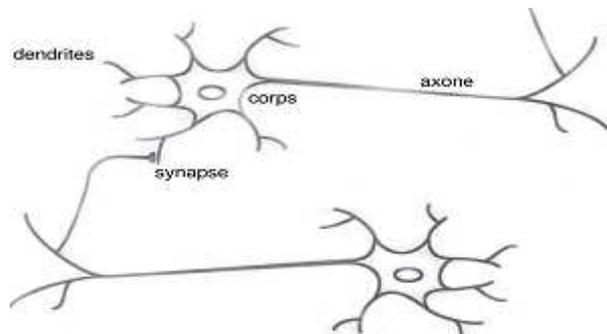


Figure II.1 Le neurone biologique.

Il existe au niveau de la jonction synapse un espace vide à travers lequel le signal électrique ne peut pas se propager. La transmission se fait alors par l'intermédiaire de substances chimiques, les neuromédiateurs. Quand un signal arrive au niveau de la synapse, il provoque l'émission de neuromédiateurs qui vont se fixer sur des récepteurs de l'autre côté de l'espace inter synaptique. (voir figure II.2). Quand suffisamment de molécules se sont fixées, un signal électrique est émis de l'autre côté et on a donc une transmission.

D'une façon simple, on peut dire que le soma du neurone traite les courants électriques qui lui parviennent de ses dendrites, et qu'il transmet le courant électrique résultant de ce traitement aux neurones auxquels il est connecté par l'intermédiaire de son axone.

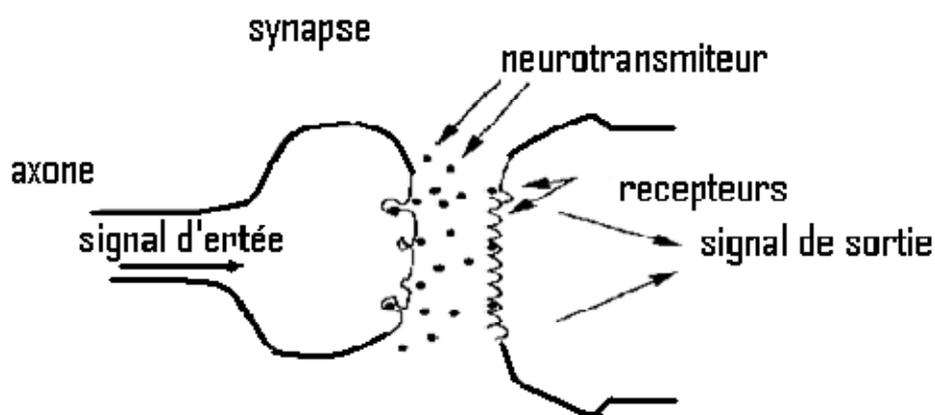


Figure II.2 Transmission synaptique.

III.4.2. Modèle artificielle (mathématique) [11]

Le modèle mathématique d'un neurone artificiel est illustré à la figure II.3.

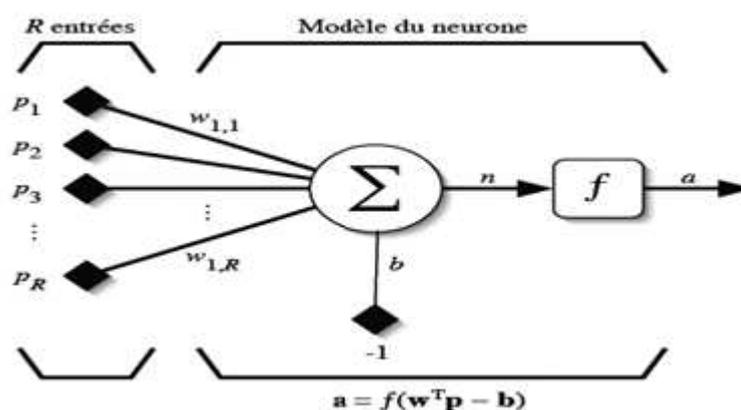


Figure II.3 : Modèle d'un neurone artificiel [12]

Un neurone est essentiellement constitué d'un intégrateur qui effectue la somme pondérée de ses entrées. Le résultat n de cette somme est ensuite transformé par une fonction

de transfert qui produit la sortie ‘a’ du neurone. En suivant les notations présentées à la figure II.3, les R entrées des neurones correspondent au vecteur $P = [P_1 P_2 P_3 \dots P_R]^T$ alors que $w = [w_{1,1} w_{1,2} \dots w_{1,R}]^T$ représente le vecteur des poids du neurone. La sortie n de l’intégrateur est donnée par l’équation suivante :

$$n = \sum_{j=1}^R w_{1,j} p_j - b$$

$$= w_{1,1} p_1 + w_{1,2} p_2 + \dots + w_{1,R} p_R - b \tag{II.1}$$

que l’on peut aussi écrire sous forme matricielle :

$$n = w^T p - b \tag{II.2}$$

Cette sortie correspond à une somme pondérée des poids et des entrées moins ce qu’on nomme le biais b du neurone. Le résultat n de la somme pondérée s’appelle le niveau d’activation du neurone. Le biais s’appelle aussi le seuil d’activation du neurone. Lorsque le niveau d’activation atteint ou dépasse le seuil b , alors l’argument de f devient positif (ou nul), sinon, il est négatif.

On peut faire un parallèle entre ce modèle mathématique et certaines informations que l’on connaît à propos du neurone biologique. Un poids d’un neurone artificiel représente donc l’efficacité d’une connexion synaptique. Un poids négatif vient inhiber une entrée, alors qu’un poids positif vient l’accentuer. Il importe de retenir que ceci est une grossière approximation d’une véritable synapse qui résulte en fait d’un processus chimique très complexe et dépendant de nombreux facteurs extérieurs encore mal connus.

Un autre facteur limitatif dans le modèle est son caractère discret. En effet, pour pouvoir simuler un réseau de neurones, nous allons rendre le temps discret dans nos équations. Autrement dit, nous allons supposer que tous les neurones sont synchrones, c’est-à-dire qu’à chaque temps t , ils vont simultanément calculer leur somme pondérée et produire une sortie $a(t) = f(n(t))$. Dans les réseaux biologiques, tous les neurones sont en fait asynchrones. Revenons donc à notre modèle tel que formulé par l’équation II.2 et ajoutons la fonction d’activation pour obtenir la sortie du neurone :

$$a = f(n) = f(w^T p - b) \tag{II.3}$$

En remplaçant w^T par une matrice $W = w^T$ d’une seule ligne, on obtient une forme générale :

$$a = f(Wp - b) \tag{II.4}$$

L'équation II.4 nous amène à introduire un schéma de notre modèle plus compact que celui de la figure II.3. La figure II.4 illustre celui-ci. On y représente les R entrées comme un rectangle noir (le nombre d'entrées est indiqué sous le rectangle). De ce rectangle sort le vecteur P dont la dimension matricielle est $R \times 1$. Ce vecteur est multiplié par une matrice W qui contient les poids (synaptiques) des neurones. Dans le cas d'un neurone simple, cette matrice possède la dimension $R \times 1$. Le résultat de la multiplication correspond au niveau d'activation qui est ensuite comparé au seuil b (un scalaire) par soustraction. Finalement, la sortie du neurone est calculée par la fonction d'activation f . La sortie d'un neurone est toujours un scalaire.

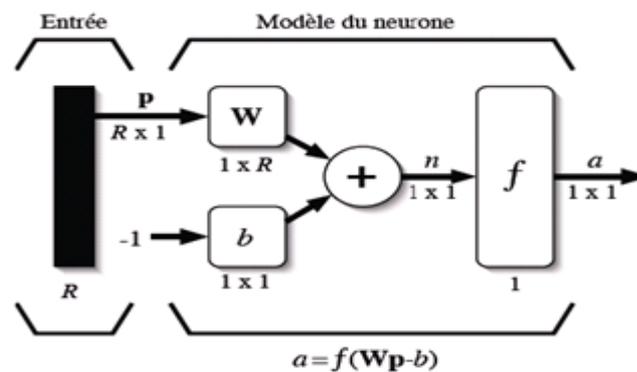


Figure II.4 : Représentation matricielle du modèle d'un neurone artificiel.

II.4.3. Fonctions de transfert [11]

Jusqu'à présent, nous n'avons pas spécifié la nature de la fonction d'activation. Il se trouve que plusieurs possibilités existent. Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone et sont énumérées au tableau II.1.

| Nom de la fonction | Relation d'entrée/sortie | Icône | Nom Matlab |
|-----------------------------|---|---|------------|
| seuil | $a = 0$ si $n < 0$ $a = 1$ si $n > 0$ |  | hardlin |
| seuil symétrique | $a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$ |  | hardlims |
| linéaire | $a = n$ |  | purelin |
| linéaire saturée | $a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$ |  | satlin |
| linéaire saturée symétrique | $a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$ |  | satlins |
| linéaire positive | $a = 0$ si $n < 0$ $a = n$ si $n \geq 0$ |  | poslin |
| sigmoïde | $a = \frac{1}{1+\exp^{-n}}$ |  | logsig |
| tangente hyperbolique | $a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$ |  | tansig |
| compétitive | $a = 1$ si n maximum $a = 0$ autrement |  | compet |

Tableau II.1 : Fonctions de transfert $a = f(n)$

Les trois fonctions les plus utilisées sont les fonctions seuil (en anglais hard limit), linéaire et sigmoïde.

Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, une entrée négative ne passe pas le seuil, la fonction retourne alors la valeur 0 (on peut interpréter ce 0 comme signifiant faux), alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction retourne 1 (vrai). Utilisée dans le contexte d'un neurone, cette fonction est illustrée à la figure III.5.a. On remarque alors que le biais b , dans l'expression détermine l'emplacement du seuil sur l'axe $w^T p$, ou la fonction passe de 0 à 1.

$$a = \text{hard lim}(w^T p - b) \tag{II.5}$$

La fonction linéaire est très simple, elle affecte directement son entrée à sa sortie :

$$a = n \tag{II.6}$$

Appliquée dans le contexte d'un neurone, cette fonction est illustrée à la figure II.5.b. Dans ce cas, la sortie du neurone correspond à son niveau d'activation dont le passage à zéro se produit lorsque $w^T p = b$

La fonction de transfert sigmoïde est quant à elle illustrée à la figure II.5.c. Son équation est donnée par :

$$a = \frac{1}{1 + \exp^{-n}} \quad (\text{II.7})$$

Elle ressemble soit à la fonction seuil, soit à la fonction linéaire, selon que l'on est loin ou près de b , respectivement. La fonction seuil est non linéaire car il y a une discontinuité lorsque $w^T p = b$. De son côté, la fonction linéaire est tout à fait linéaire. Elle ne comporte aucun changement de pente. La sigmoïde est un compromis intéressant entre les deux précédentes. Notons finalement, que la fonction «tangente hyperbolique» est une version symétrique de la sigmoïde.

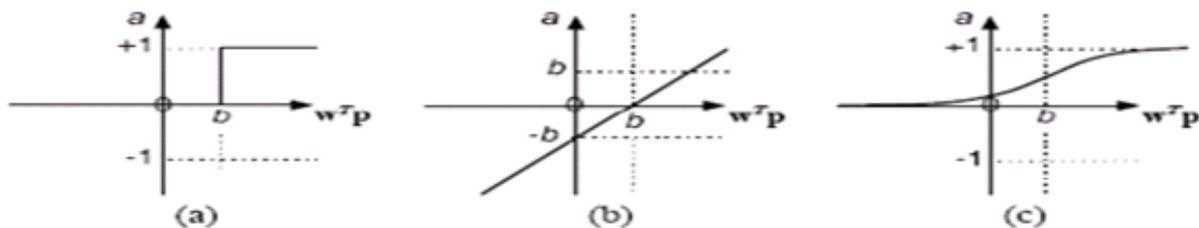


Figure II.5 : fonction de transfert : (a) neurone « seuil »; (b) neurone « linéaire »; (c) neurone « sigmoïde ».

II.5. Structure général d'un réseau de neurone formel [11]

Comme les neurones formels sont interconnectés entre eux, leur niveau d'interconnexion définit des couches. Nous distinguons une couche d'entrée contenant un ensemble de neurones qui vont recevoir les informations primitives, puis après un certain nombre de couches intermédiaires dites couches cachées, une couche de sortie qui contient les neurones finaux qui vont transmettre les informations de sortie traitées par la totalité du réseau. Les systèmes connexionnistes traitent les informations en parallèle. L'ensemble du réseau et des couches de neurones formels est alors modifié par les informations en entrée durant la phase d'apprentissage.

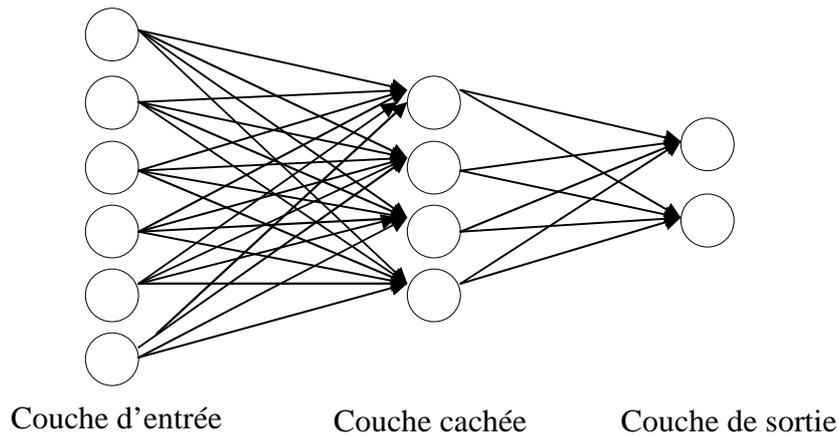


Figure II.6 : Structure générale d'un réseau de neurone formel.

II.6. Apprentissage

Pour un RNA (réseau de neurone artificiel), l'apprentissage peut être considéré comme le problème de la mise à jour des poids des connexions au sein du réseau, afin de réussir la tâche qui lui est demandée. L'apprentissage est la caractéristique principale des RNA et il peut se faire de différentes manières et selon différentes règles.

II.6.1. Type d'apprentissage

II.6.1.1. *Le mode supervisé* [10]

Dans ce mode, un professeur qui connaît parfaitement la sortie désirée ou correcte guide le réseau en lui apprenant à chaque étape le bon résultat. Donc l'apprentissage consiste à comparer le résultat obtenu avec le résultat désiré, puis à ajuster les poids des connexions pour minimiser la différence entre les deux.

II.6.1.2. *Le mode non supervisé* [10]

Dans l'apprentissage non supervisé, le réseau modifie ses paramètres en tenant compte des informations locales. Ces méthodes n'ont pas besoins de sorties désirées préétablies.

III.6.1.3. *Le renforcement*

Il est utilisé quand une information en retour sur la qualité de la performance est fournie, mais que la conduite souhaitée du réseau n'est pas complètement spécifiée par un professeur. Donc l'apprentissage est moins dirigé que l'apprentissage supervisé.

Contrairement à l'apprentissage non supervisé où aucun signal de retour n'est donné, le réseau à apprentissage renforcé peut utiliser le signal de renforcement pour trouver les poids les plus désirables quand c'est nécessaire.

III.6.2.4. *Le mode hybride*

Le mode hybride reprend en fait les deux autres approches, puisque une partie des poids va être déterminée par apprentissage supervisé et l'autre partie par apprentissage non supervisé.

II.6.3. Règles d'apprentissage

II.6.3.1. *Règle de correction d'erreurs [12]*

Cette règle s'inscrit dans le paradigme d'apprentissage supervisé, c'est-à-dire dans le cas où l'on fournit au réseau une entrée et la sortie correspondante. Si on considère Y comme étant la sortie calculée par le réseau, et D la sortie désirée, le principe de cette règle est d'utiliser l'erreur ($D-Y$), afin de modifier les connexions et de diminuer ainsi l'erreur globale du système. Le réseau va donc s'adapter jusqu'à ce que Y soit égal à D . Ce Principe est notamment utilisé dans le modèle du perceptron simple.

II.6.2.2. *Apprentissage de Boltzmann*

Les réseaux de Boltzmann sont des réseaux symétriques récurrents. Ils possèdent deux sous-groupes de cellules, le premier étant relié à l'environnement (cellules dites visibles) et le second ne l'étant pas (cellules dites cachées). Cette règle d'apprentissage est de type stochastique (c'est-à-dire qui relève partiellement du hasard) et elle consiste à ajuster les poids des connexions, de telle sorte que l'état des cellules visibles satisfasse une distribution probabiliste souhaitée.

II.6.2.3. *Règles de Hebb*

Cette règle, basée sur des données biologiques, modélise le fait que si des neurones, de part et d'autre d'une synapse, sont activés de façon synchrone et répétée, la force de la connexion synaptique va croître. Il est à noter ici que l'apprentissage est localisé, c'est à dire que la modification d'un poids synaptique $w_{i,j}$ ne dépend que de l'activation d'un neurone i et d'un autre neurone j .

II.6.2.4. Règle d'apprentissage par compétitions [12]

La particularité de cette règle, c'est qu'ici l'apprentissage ne concerne qu'un seul neurone. Le principe de cet apprentissage est de regrouper les données en catégories. Les patrons similaires vont donc être rangés dans une même classe, en se basant sur les corrélations des données, et seront représentés par un seul neurone, on parle de « winner-take all ».

Dans un réseau à compétition simple, chaque neurone de sortie est connecté aux neurones de la couche d'entrée, aux autres cellules de la couche de sortie (connexions inhibitrices) et à elle-même (connexion excitatrice). La sortie va donc dépendre de la compétition entre les connexions inhibitrices et excitatrices.

II.7. Architecture [12]

On peut classer les réseaux de neurones artificiels en deux grandes catégories, bouclée et non bouclée, comme l'illustre la figure suivante :

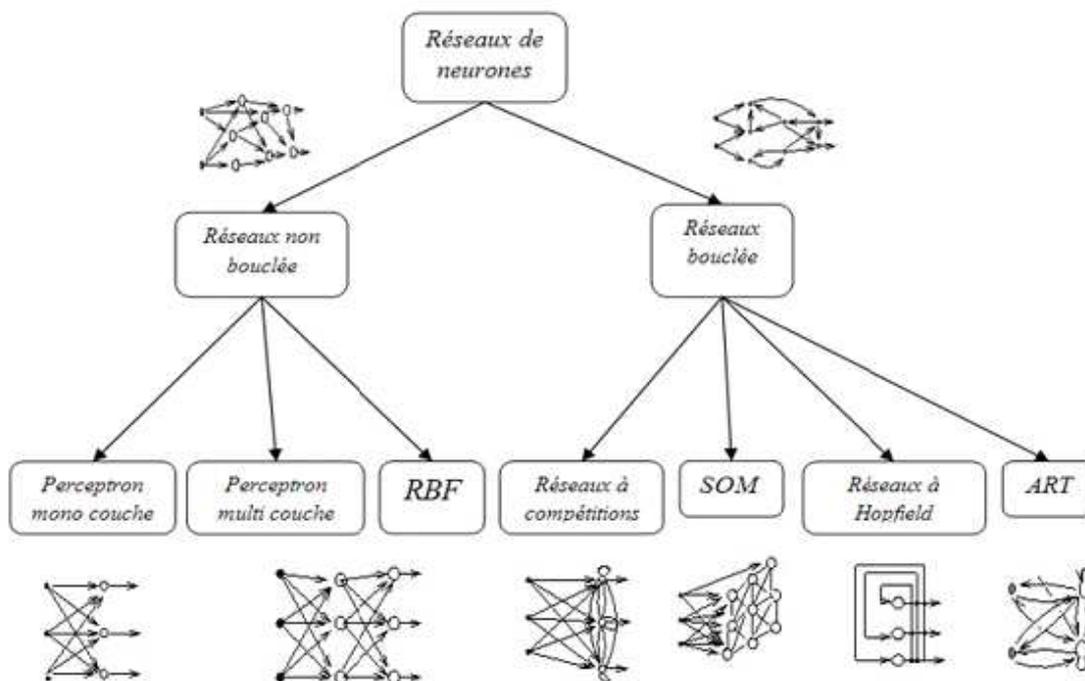


Figure II.7 les différents types de réseaux neurones.

II.7.1. Les réseaux non bouclés "FEED-FORWARD"

Appelés aussi "réseaux de type Perceptron, ce sont des réseaux dans lesquels l'information se propage de couche en couche sans retour en arrière possible.

Il existe deux types de perceptron, le mono couche et le multicouche.

Le perceptron monocouche est le premier RNA, conçu en 1958 par Rosenblatt. Il est linéaire et monocouche. C'est un réseau simple, puisqu' il est composé d'une couche d'entrée et d'une couche de sortie. (Voir la **figure II.8**)

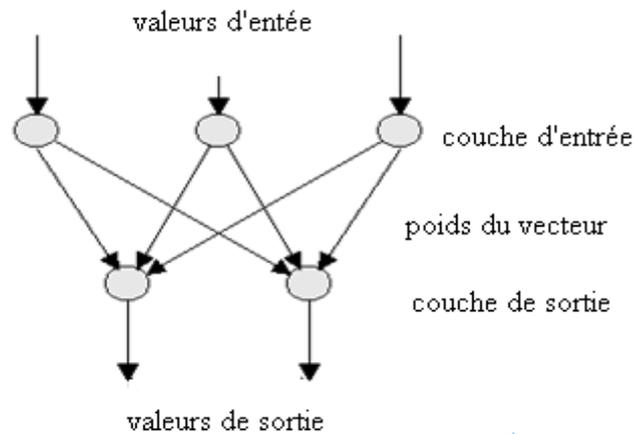


Figure II.8 le perceptron monocouche [14]

Le perceptron multicouche est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante et il n'y a pas de connexions entre les cellules d'une même couche.

Les réseaux à fonction radiale sont les réseaux que l'on nomme aussi RBF ("Radial Basic Functions"). L'architecture est la même que pour les perceptrons multi couche cependant, les fonctions de base utilisées ici sont des fonctions Gaussiennes. Les RBF seront donc employés dans les mêmes types de problèmes que les perceptrons multi couche à savoir, en classification et en approximation de fonctions, particulièrement. L'apprentissage le plus utilisé pour les RBF est le mode hybride et les règles sont soit, la règle de correction de l'erreur soit, la règle d'apprentissage par compétition.

II.7.2. Les réseaux bouclés "FEED-BACK"

Appelés aussi "réseaux récurrents", ce sont des réseaux dans lesquels il y a retour en arrière de l'information.

II.7.2.1. Les cartes auto organisatrices de Kohonen (SOM)

C'est probablement le type de réseau de neurones le plus utile. Ce type de réseau, est aussi appelé SOM « Self Organised Maps ». Ce sont des réseaux à apprentissage non supervisé qui établissent une carte discrète, ordonnée topologiquement, en fonction de

l'entrée. Le réseau forme ainsi une sorte de treillis dont chaque noeud est un neurone associé à un vecteur de poids. La correspondance entre chaque vecteur de poids est calculée pour chaque entrée.

Par la suite, le vecteur de poids ayant la meilleure corrélation, ainsi que certains de ses voisins, vont être modifiés afin d'augmenter encore cette corrélation.

II.7.2.2. Les réseaux de Hopfield [15]

Les réseaux de Hopfield sont des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à chaque autre neurone et il n'y a aucune différenciation entre les neurones d'entrée et de sortie. Ils fonctionnent comme une mémoire associative non-linéaire et sont capables de trouver un objet stocké en fonction de représentations partielles ou bruitées. L'application principale des réseaux de Hopfield est l'entrepôt de connaissances mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est le mode non supervisé.

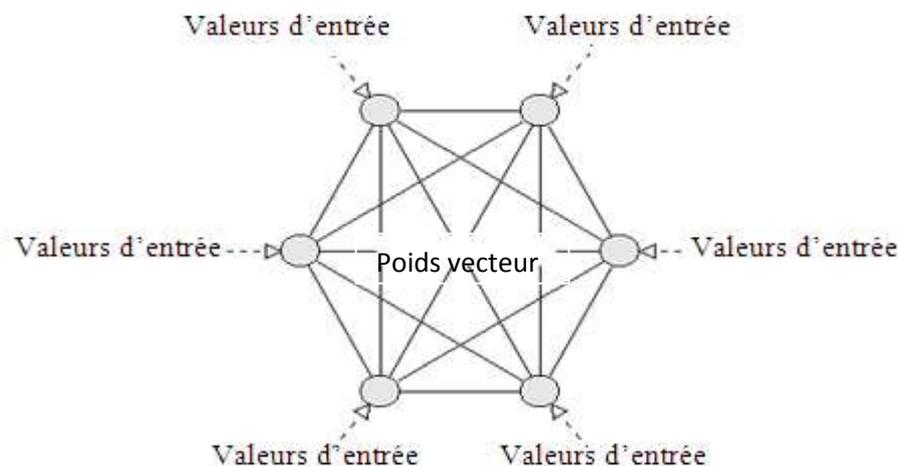


Figure II.9. Le réseau de Hopfield

II.7.2.3. Les ART [15]

Les architectures sont basées sur une théorie d'inspiration biologique assez complexe. Elles se manifestent par différentes implantations spécifiques dont celles nommées «ART1», «ART2», «ART3», etc. L'architecture ART1 possède la particularité de n'accepter que des entrées binaires, alors que l'ART2 accepte des entrées continues en incorporant aux mécanismes de l'ART1 différentes opérations complexes de normalisation. Quant à l'ART3, il développe l'ART2 d'avantage en lui ajoutant un nouveau mécanisme de réinitialisation

biologiquement inspiré. Ces trois architectures utilisent toutes un processus d'apprentissage non supervisé.

II.8. Application des réseaux de neurones [16]

Grâce à son application dans différents domaines (informatique, électronique, neurobiologie et même philosophie), l'étude des réseaux de neurones est une voie prometteuse de l'Intelligence Artificielle, qui a des applications dans de nombreux domaines :

- En industrie pour le contrôle qualité, le diagnostic de panne, les corrélations entre les données fournies par différents capteurs...etc.
- Dans les finances pour la prévision et la modélisation du marché, la sélection d'investissements, l'attribution de crédits...etc.
- En télécommunications et en informatique pour l'analyse du signal, l'élimination du bruit, la reconnaissance de formes (bruits, images, paroles)...etc.
- En environnement pour l'évaluation des risques, l'analyse chimique, la prévision et la modélisation météorologiques...etc.
- Dans la modélisation de l'apprentissage et l'amélioration des techniques de l'enseignement.
- Pour la classification, par exemple pour la classification d'espèces animales par espèce étant donné une analyse ADN.

II.9. Conclusion

L'apprentissage permet aux réseaux de neurones de réaliser des tâches complexes dans différents types d'application (classification, identification, reconnaissance de caractères, de la voix, vision, système de contrôle...). Ces réseaux de neurones peuvent souvent apporter une solution simple à des problèmes encore trop complexes ne pouvant être résolus rapidement par les ordinateurs actuels (puissance de calcul insuffisante) ou par notre manque de connaissances.

La méthode d'apprentissage dite supervisée est souvent utilisée mais des techniques d'apprentissage non supervisé existent pour des réseaux de neurones spécifiques. Ces réseaux peuvent, par exemple, identifier des classes dans une image, comme les cartes auto-organisatrices de Kohonen qui sera l'objet du prochain chapitre et que nous utiliserons par la suite dans la segmentation des images.

III.1. Introduction

La classification automatique et les cartes auto-organisatrice sont proches, puisque la plupart des méthodes de classification automatique cherchent à regrouper les données similaires. La notion d'ordre topologique constitue l'apport des réseaux de neurones à l'apprentissage non supervisé au domaine de la classification automatique, qui est un des grands thèmes abordés en analyse des données.

Ce chapitre est consacré à la classification non supervisée avec la carte auto-organisatrice de Kohonen introduite pour la première fois en 1981.

III.2. Notation et définition [17]

Nous notons D l'ensemble qui représente l'espace des observations de dimension n tel que $D \subset R^n$. Soit le sous-ensemble $A = \{z_i; i = 1, \dots, N\}$ de D , constitué de N individus qui formera l'ensemble d'apprentissage permettant d'estimer les différents paramètres du modèle.

Soit l'ensemble $W = \{w_c; c = 1, \dots, p\}$ de p vecteurs de D . Ces vecteurs de dimension n constituent l'ensemble des référents.

Soit X une fonction d'affectation qui est une application de D dans l'ensemble des indices $\{1, \dots, p\}$. Cette fonction permet de réaliser une partition $P = \{p_1, \dots, p_c, \dots, p_p\}$ de D en p sous-ensemble, $p_c = \{z \in D / X(z) = c\}$.

La figure III.1 montre le principe général de la modélisation. Une observation z est associée à un indice c choisi parmi p à l'aide de la fonction X ; cet indice permet de définir le référent w_c . On peut donc considérer que le vecteur référent w_c est un représentant de l'ensemble p_c dont il résume l'ensemble des observations. Les paramètres des différents modèles étant estimés à partir des observations de l'ensemble d'apprentissage A . La connaissance de l'ensemble des vecteurs référents w_c et de X détermine ce que l'on appelle une quantification vectorielle qui est utilisée pour affecter une observation $z \in D$ à son référent $w_{X(z)}$; tel que $X(z)$ représente l'indice du référent auquel est associée l'observation z . La connaissance de la fonction X permet donc, au-delà de la quantification vectorielle, de définir la partition P de D en p sous-ensembles.

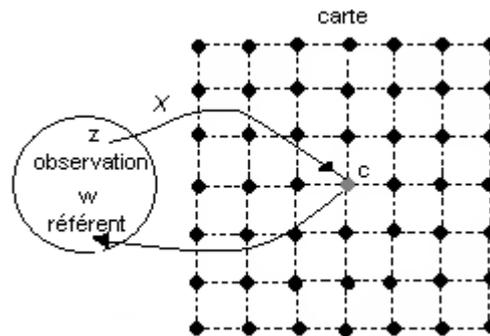


Figure III.1 : Principe général de la modélisation

III.3. Architecture neuronale et carte topologique [17]

Dans une carte auto-organisatrice, les vecteurs référents fournissent une représentation discrète de l'espace d'entrée. Ils sont positionnés de telle façon qu'ils conservent la forme topologique de l'espace d'entrée. De plus, ils gardent les relations de voisinage dans la grille, donc ils permettent une indexation facile, en utilisant les coordonnées dans la grille.

On peut représenter l'ensemble constitué par la carte et les référents sous la forme d'un réseau de neurones (comme l'indique la figure III.2) constitué de deux couches :

- **La couche d'entrée** : sert à la présentation des observations à classer ; les états de tous ces neurones sont forcés aux valeurs des observations. Cette couche contient donc exactement n neurones (n étant la dimension de l'espace des observations).

- **La couche d'adaptation** : est formée du treillis des neurones qui forment la carte. La structure du réseau employé peut être soit fixée a priori, soit évoluer lors de l'apprentissage. Les neurones utilisés à ce niveau sont des simples neurones « distances », chacun d'entre eux étant connecté à tous les éléments de la couche d'entrée. Soit φ la grille neuronale carée d'une carte auto-organisatrice. Une carte de neurone assigne à chaque vecteur d'entrée $z \in A$, un neurone $c \in \varphi$ désigné par son vecteur référent w_c , tel que le vecteur référent w_c est le plus proche de z .

Kohonen a montré qu'il existe entre les neurones émetteurs et récepteurs un mécanisme d'interaction latérale qui dépend de la distance entre les neurones concernés. Ces cartes sont utilisées pour cartographier un espace réel, c'est-à-dire pour étudier la répartition de données dans un espace à grande dimension. En pratique, cette cartographie peut servir à réaliser des tâches de discrétisation, quantification vectorielle, ou classification et regroupement.

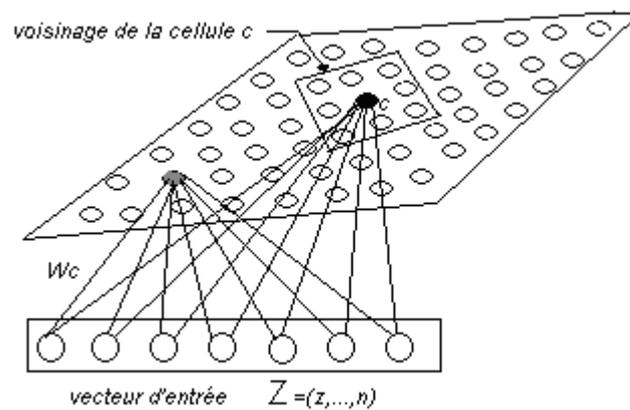


Figure III.2 : projection des observations sur la carte topologique.[11]

III.4. Notion de voisinage et de distance

La structure de graphe induit une distance δ sur φ , pour toute paire de neurones (c, r) de cette carte, $\delta(c, r)$ est définie comme étant la longueur du plus court chemin entre c et r sur le graphe φ . Pour chaque neurone c , cette distance discrète permet de définir la notion de voisinage d'ordre d de c : $V_c(d) = \{r \in \varphi, \delta(c, r) \leq d\}$.

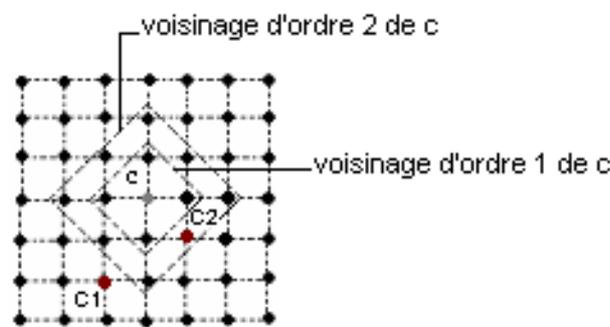


Figure III.3 : Représentation de la topologie discrète d'une carte topologique à 2D.

Dans la figure III.3 les distances entre c et c_1 est $\delta(c, c_1) = 3$, et c et c_2 est $\delta(c, c_2) = 2$. Cette figure présente l'ensemble de ces notions de voisinage et de distance pour une carte topologique constituée par un treillis à deux dimensions.

Les distances $\delta(c, r)$ qui lient le neurone c aux autres neurones (r) de la carte permettent de faire varier l'influence relative des différents neurones, cette importance est quantifiée par $K(\delta(c, r))$. La fonction de voisinage K force les neurones qui se trouvent dans

le voisinage de c à rapprocher leurs vecteurs référents du vecteur d'entrée Z . Moins un neurone est proche du vainqueur dans la grille, moins son déplacement est important.

La fonction de voisinage à seuil dans laquelle les neurones du voisinage ont la même influence est donnée par la relation :

$$K(\delta) = \begin{cases} 1 & \text{si } \delta < 1 \\ 0 & \text{sinon} \end{cases} \quad (\text{III.1})$$

La fonction de voisinage de type gaussien dont l'influence entre deux neurones dépend de la distance entre ces neurones est tel que :

$$K(\delta) = \exp(-|\delta|) \quad (\text{III.2})$$

Afin de gérer la taille de voisinage, on utilise la famille de fonction K^T paramétrée par T .

Les fonctions noyaux qui sont les plus utilisées dans la pratique sont :

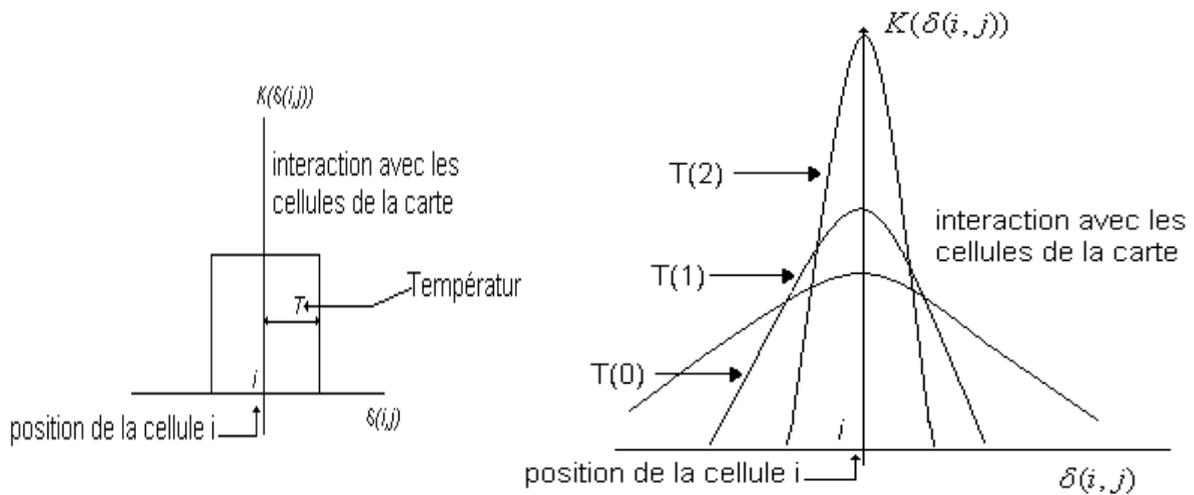
$$\bullet \quad K^T(\delta) = \begin{cases} 1 & \text{si } \delta < T \\ 0 & \text{sinon} \end{cases} \quad (\text{III.3})$$

$$\bullet \quad K^T(\delta) = \exp\left(-\frac{|\delta|}{T}\right) \quad (\text{III.4})$$

$$\bullet \quad K^T(\delta) = \exp\left(-\frac{\delta^2}{T^2}\right) \quad (\text{III.5})$$

La fonction (III.3) est représenté en figure (III.4.a), elle se confond avec la fonction (III.1) pour $T=1$.

La fonction (III.4) est représenté sur la figure (III.4.b) pour des valeurs de températures différentes.



a°) Fonction de voisinage à seuil

b°) Fonction de voisinage de type gaussien

III.4 : Exemples de fonctions de voisinage

III.5. Algorithme d'optimisation [17]

Soit J_{som}^T une fonction de coût dont le minimum fournit une partition formée de sous-ensembles qui sont suffisamment compacts, mais pour lesquels on est capable de définir un ordre induit à partir de la topologie de la carte. Les algorithmes des cartes auto-organisatrices minimisent donc cette fonction qui est donnée par la formule suivante :

$$J_{som}^T(X, w) = \sum_{Z_i \in A} \sum_{c \in C} K^T(\delta(c, X(z_i))) \|z_i - w_c\|^2 \tag{III.6}$$

La minimisation de cette fonction pour une valeur T fixée, est réalisée par itérations successives chacune se décomposant en deux phases. La première phase affecte l'ensemble des observations et la seconde minimise la valeur de la fonction coût associée à la partition.

III.5.1. Phase d'affectation

Cette phase minimise la fonction $J_{som}^T(X, W)$ par rapport à la fonction d'affectation. On suppose que l'ensemble W des référents est constant et égale à la valeur calculée précédemment. Ainsi, chaque observation z est affectée au référent le plus proche au sens d'une distance.

La fonction d'affectation utilisée par Kohonen est donnée par :

$$X(z_i) = \arg \min_c \|z_i - w_c\|^2 \tag{III.7}$$

III.5.2. Phase de minimisation

Cette phase minimise la fonction $J_{som}^T(X, W)$ par rapport à l'ensemble des référents W . Cette minimisation est effectuée en gardant la fonction d'affectation X fixée et égale à la fonction calculée durant la phase précédente. La minimisation est donc obtenue pour la valeur qui annule la dérivée de $J_{som}^T(X, W)$. Dans la version stochastique de Kohonen, il n'est pas obligatoire de trouver le minimum global de $J_{som}^T(X, W)$ pour X fixée, il suffit de faire décroître sa valeur. Ainsi à l'itération t et pour un neurone c le nouveau référent est donnée par :

$$w_c^t = w_c^{t-1} - \mu^t \frac{\partial J_{som}^T}{\partial w_c^{t-1}} \quad (\text{III.8})$$

Avec,
$$\frac{\partial J_{som}^T}{\partial w_c} = 2 \sum_{z_i \in A} K^T(\delta(c, X(z_i)))(z_i - w_c) \quad (\text{III.9})$$

Cette méthode non adaptative suppose que l'on dispose de toutes les observations de l'ensemble d'apprentissage A . La contribution d'une seule observation z_i à la correction de w_c et représentée par le terme de la somme $2K^T(\delta(c, X(z_i)))(z_i - w_c^{t-1})$.

D'où les nouveaux référents sont donnés par le terme :

$$w_c^t = w_c^{t-1} - \mu^t K^T(\delta(c, X_t(z_i)))(w_c^{t-1} - z_i) \quad (\text{III.10})$$

Où μ^t : représente le pas du gradient de l'itération t .

III.5.3. Les étapes de l'algorithme de Kohonen [17]

Donc l'algorithme de Kohonen est résumé de la manière suivante :

Algorithme de Kohonen

- choisir la structure et la taille de la carte et les p référents initiaux (en général, d'une manière aléatoire).

-fixer les valeurs de T_{\max} et T_{\min} et le nombre d'itération N_{iter} ; prendre $t=0$.

2. Etape itérative t

L'ensemble des référents w^{t-1} de l'étape précédente étant connus :

- choisir une observation z_i (en général d'une manière aléatoire).

- calculer la nouvelle valeur de T en appliquant la formule :

$$T = T_{\max} * \left(\frac{T_{\min}}{T_{\max}} \right)^{\frac{t}{N_{iter}}} \quad (\text{III.11})$$

Pour cette valeur du paramètre effectuer les deux phases suivantes :

- phase d'affectation : On suppose W^{t-1} connu ; on affecte l'observation z_i au neurone $X_t(z_i)$ défini par la relation III.7

- phase de minimisation : Calcul de l'ensemble des nouveaux référents w^t par la relation (III.10).

3. Répéter l'étape itérative en faisant décroître la valeur du paramètre T jusqu'à ce que l'on atteigne $t = N_{iter}$.

L'algorithme détermine le centre d'activité comme étant le voisinage de la carte associé au neurone dont l'état ($\|z_i - w_c\|$) est le plus petit. C'est l'utilisation de ce voisinage qui introduit les contraintes topologiques dans la représentation finale.

Dans la formule de modification des vecteurs référents, le pas du gradient μ^t décroît avec les itérations. Au début de l'algorithme, la valeur de μ^t est grande et la décroissance de la fonction J_{som}^T n'est pas strictement assurée. Par la suite, le pas du gradient devient suffisamment petit : la modification des référents à chaque itération est petite.

Lorsque le paramètre T est petit, la modification de la relation (III.11) ne concerne qu'un ensemble réduit de neurones. La fonction de voisinage $K^T(\delta(c,r))$ décroît lorsque la distance $\delta(c,r)$ croît.

III.6. Recherche d'une partition adaptée aux classes recherchées

Parmi les différentes applications réalisées à l'aide des cartes auto-organisatrices, un assez grand nombre sont des tâches de classification. L'auto-organisation, telle que nous venons de la décrire, ne permet pas de résoudre ce type de problèmes : le résultat de l'apprentissage non supervisé permet d'affecter une observation à un sous-ensemble d'une partition, indépendamment de toute notion de classe. On considère que l'on dispose d'un très grand nombre d'observation dont on connaît pas exactement la classe. La partition proposée dépend de la densité de probabilité qui est sous-jacente à l'ensemble des observations de l'ensemble d'apprentissage. Les régions ayant une forte densité d'observation permettent d'obtenir une information plus précise, elles sont décrites par une sous-partition fine, celle de faible densité par une sous-partition grossière. L'idée principale de l'algorithme des cartes auto-organisatrices est de faire apparaître des groupements cohérents (le sous-ensemble de la partition) en relation avec l'application traitée. Si l'application envisagée consiste en un problème de classification en S classes, les différents sous-ensembles doivent s'intégrer au mieux avec celles-ci. On cherche alors à affecter chaque sous-ensemble à l'une de ces classes.

Puisque chaque sous ensemble est associé à un neurone de la carte, le problème de classification se résume à celui de l'étiquetage de chaque neurone de la carte au moyen de l'une de S classes du problème. L'étiquetage peut se faire de trois manières différentes :

- Il est possible de demander à un expert du domaine d'application de reconnaître, grâce aux caractéristiques de cette observation, la classe à laquelle elle appartient.
- Regrouper les neurones de la carte d'une manière statistique, le recours à l'expertise ne se faisant qu'à l'issue de cette phase.
- On peut aussi les regrouper en utilisant une classification hiérarchique ascendante

III.7. Conclusion

Dans ce chapitre, nous avons détaillé la classification non supervisée par les cartes auto organisatrices, par l'algorithme de Kohonen. Nous l'utiliserons dans notre cas dans la segmentation des images couleurs qui est un exemple de classification. Une fois la carte topologique est organisée, nous regroupons les neurones en utilisant une classification hiérarchique ascendante. Cette classification, ainsi que différents résultats de la méthode seront détaillés au chapitre suivant.

IV.1. Introduction

Notre but est la segmentation des images couleur. N'ayant aucune connaissance a priori sur l'appartenance des pixels de l'image aux classes, nous avons opté pour une segmentation non supervisée en utilisant un réseau de neurones SOM (Self Organizing Maps). En effet, ce réseau permet l'auto-organisation de la carte topologique au fur et à mesure que les données (pixels de l'image) sont introduites. Par la suite, pour regrouper au mieux les référents afin d'obtenir le nombre de classe désiré, nous appliquons une classification ascendante hiérarchique qui est une méthode de classification automatique.

Nous avons donc implémenté sous Matlab 7.0 l'algorithme de Kohonen dont l'utilisation nécessite le choix de la fonction de voisinage K^T , le pas de gradient μ' et la taille de la carte topologique τ . Pour le regroupement des neurones, nous avons aussi implémenté l'algorithme de la classification hiérarchique ascendante qui nécessite la connaissance du nombre de groupes n_c à former dans le réseau.

Dans ce chapitre, nous appliquerons notre programme sur différentes images couleur en faisant varier un à un les éléments cités précédemment à savoir la fonction de voisinage K^T , le pas de gradient μ' , la taille de la carte topologique τ et le nombre de groupes n_c . Nous afficherons et commenterons les résultats de la segmentation à chaque fois.

IV.2 La classification hiérarchique ascendante (CAH)

Il s'agit d'une méthode qui calcule une hiérarchie de partitions, chaque partition permet de regrouper de manière différente les neurones de la carte. Les différentes partitions de la hiérarchie sont déterminées d'une manière itérative, en commençant par la partition la plus fine qui est composé de l'ensemble des neurones.

La classification hiérarchique utilise cette partition initiale, et procède à des regroupements successifs en fusionnant à chaque itération deux sous ensembles de neurones. Le choix des deux sous ensembles qui vont fusionner à une étape donnée est effectué à l'aide d'une mesure de similitude, définie entre deux sous ensembles. On choisit, parmi tous les couples de sous ensembles qui constituent la partition à cette étape, les deux sous ensembles de neurones les plus semblables, au sens de la mesure choisie.

La classification hiérarchique permet d'engendrer un nombre variable de sous ensembles, car le processus de regroupement peut être arrêté à tout moment. Pour une mesure

de similarité donnée le nombre d'éléments de la partition que l'on choisie dépend du nombre S de classe recherchées.

L'algorithme de CAH se résume comme suit :

Algorithme de classification hiérarchique

- 1. Initialisation.** Considérer la partition formée par les neurones ; chaque neurone est alors affecté à un sous ensemble distinct. Choisir le nombre K de groupement que l'on souhaite former.
- 2.** Pour une partition donnée trouver les deux sous-ensembles les plus proches au sens du critère de similitude choisi, et les fusionner de manière à former un seul sous ensemble.
- 3.** Si le nombre de groupement de la partition courante est inférieure à K , revenir a l'étape **2**, sinon l'algorithme se termine.

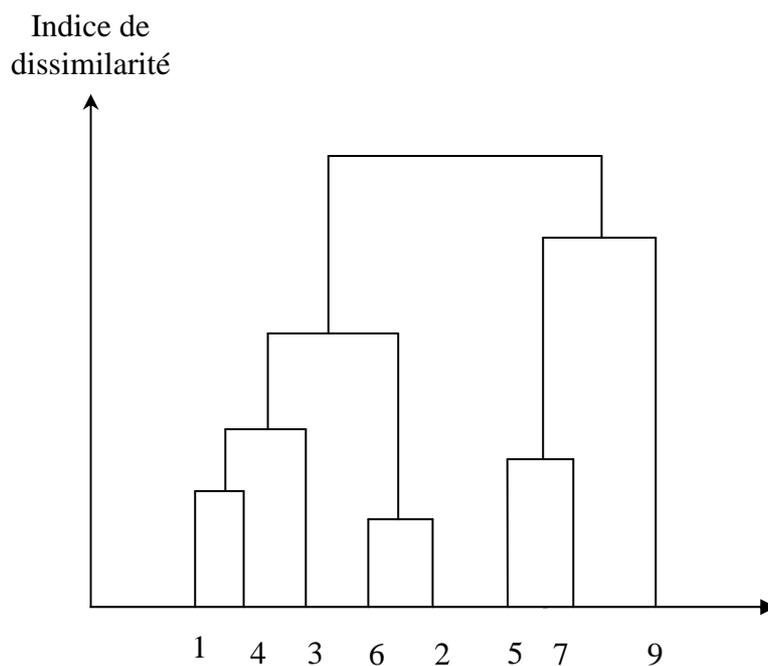


Figure IV.1 : Regroupement des neurones de la carte par classification ascendante hiérarchique

IV.3. Influence de la fonction de voisinage K^T sur la segmentation des images couleur

Les tests sont effectués sur plusieurs images couleur de mêmes tailles 512x512 pixels². Dans ce paragraphe, nous proposons d'étudier l'influence de la fonction de voisinage K^T sur la segmentation des différentes images. Pour ce faire, nous fixerons les valeurs des autres paramètres. Ainsi, la carte topologique, dans ce cas, sera de taille $\tau=5 \times 5$, le nombre de groupes n_c prendra la valeur 5 et le choix du pas du gradient s'est porté sur la fonction $\mu^t = \frac{1}{t}$. Nous allons faire varier la fonction de voisinage en considérant les trois exemples suivant :

$$K^T(\delta) = \begin{cases} 1 & \text{si } \delta < T \\ 0 & \text{sinon} \end{cases} \quad (\text{IV.1})$$

$$K^T(\delta) = \exp\left(-\frac{|\delta|}{T}\right) \quad (\text{IV.2})$$

$$K^T(\delta) = \exp\left(\frac{\delta^2}{-T^2}\right) \quad (\text{IV.3})$$

Des résultats de la segmentation des images sont présentés par les figures ci-après.

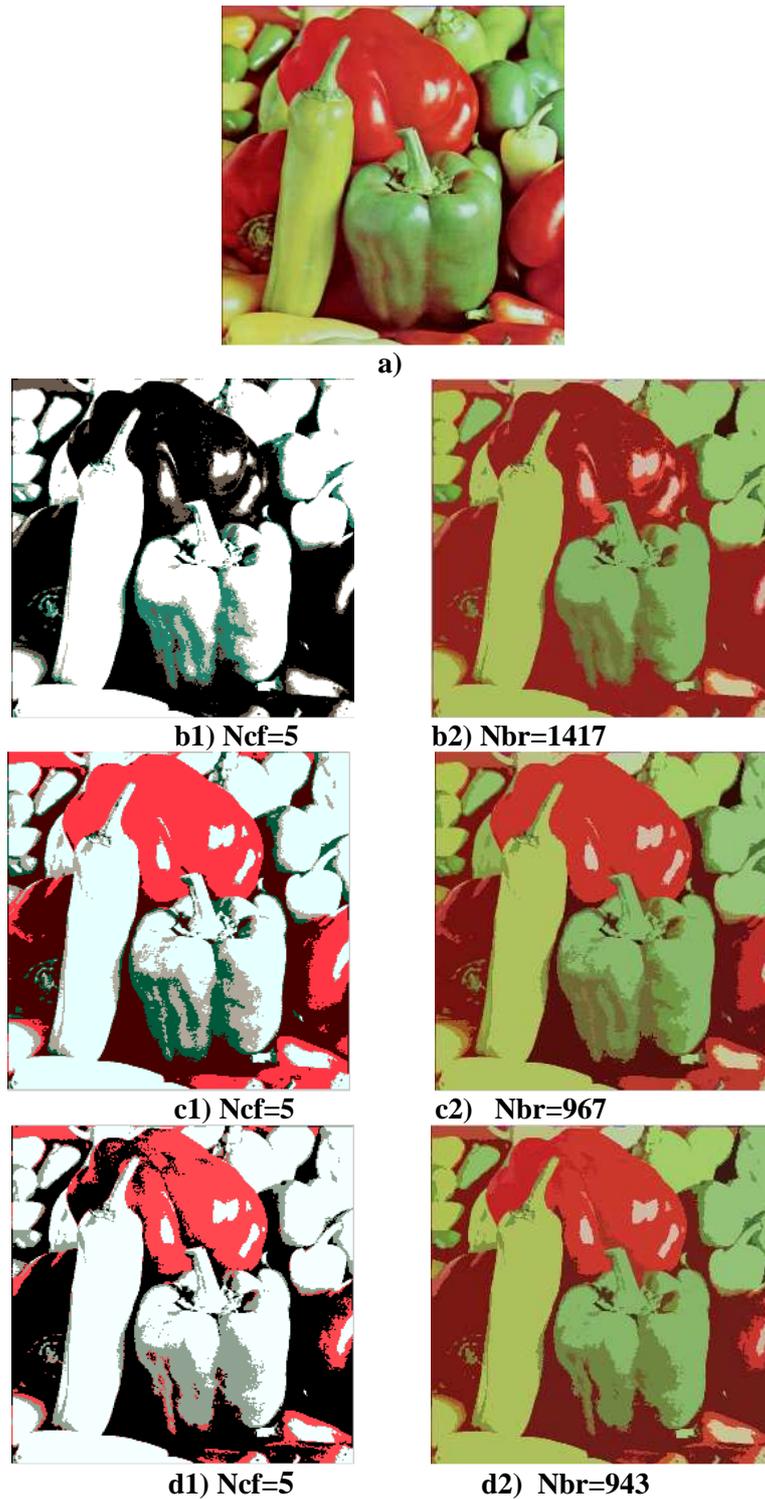


Figure IV.2. : Segmentation de l'image peppers pour $\tau=5 \times 5$, $nc=5$ et $\mu' = \frac{1}{t}$

- a) Image initiale peppers b) résultats de la segmentation en considérant la fonction (IV.1)
 c) résultats de la segmentation en considérant la fonction (IV.2) d) résultats de la segmentation en considérant la fonction (IV.3)
 1) affichage des résultats en fausses couleurs après classification 2) affichage des résultats en régions connexes.

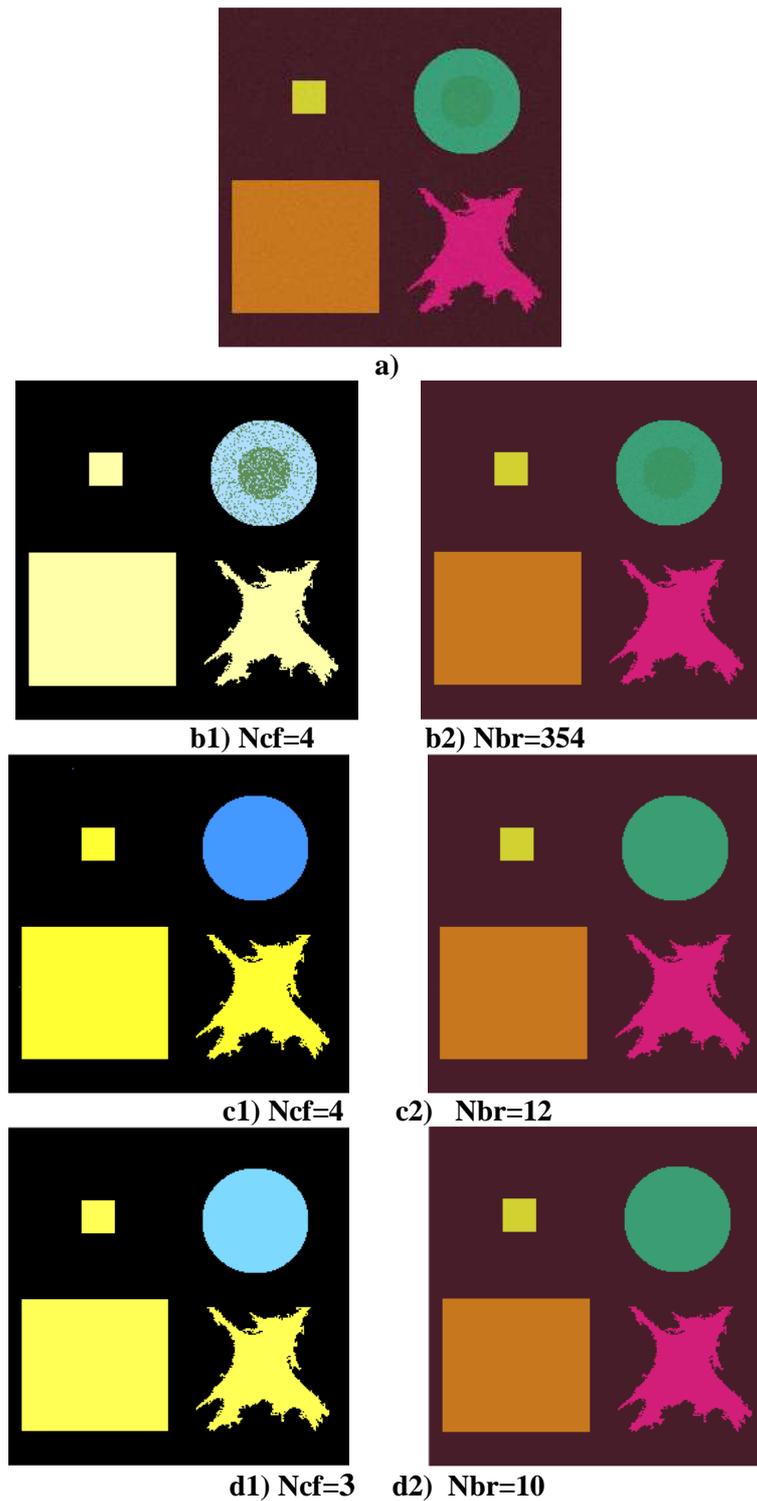


Figure IV.3. : Segmentation de l'image synthétique pour $\tau=5 \times 5$, $nc=5$ et $\mu^t = \frac{1}{t}$

- a) Image initiale synthétique b) résultats de la segmentation en considérant la fonction (IV.1)
 c) résultats de la segmentation en considérant la fonction (IV.2) d) résultats de la segmentation en considérant la fonction (IV.3)
- 1) affichage des résultats en fausses couleurs après classification 2) affichage des résultats en régions connexes.

En appliquant notre programme sur les images nous obtenons une images des classes et non pas une image segmentée. An effet, apres organisation de la carte topologique et regroupement des neurones en nc groupes, nous affectons chaque pixel au groupe qui lui est le plus proche, formant ainsi les differentes classes. Ces résultats sont affichés sur les colonnes noté 1). Nous avons remarqué que certains groupes de la carte n'ont reçu aucun pixel lors de l'affectation, ce sont des classes vides. Ainsi le nombre de classes reel detectés dans l'image n'est pas nc mais Ncf . Dans le cas de l'exemple de la figure (VI.2), le nombre de groupe nc est égale au nombre de classes réel Ncf , il n'y a donc aucun groupe vide. Dans le cas de la figure (VI.3), le nombre de classes réel Ncf est inferieur au nombre de groupes nc , dans le cas b) et c) nous avons un groupe vide et dans le cas d) nous avons deux groupes vides. En colonne 2), nous avons visualisé les images segmenté en utilisant les vraies couleurs, et Nbr , dans ce cas, correspond au nombre de regions connexes.

Les résultats des traitements en utilisant des fonctions de voisinages différentes sont affiché sur les lignes b), c) et d) des figures.

Pour l'images peppers, les résultats pour les différentes fonctions sont assez semblables, par contre pour l'image synthétique la fonction (IV.2) fournit de meilleurs résultats. Nous optons donc pour le choix de cette fonction.

IV.4. Influence du pas du gradient μ^t sur la segmentation des images

Dans cette partie, nous proposons d'étudier l'influence de μ^t sur la segmentation des différentes images. Nous considérons trois exemples :

$$\mu^t = \frac{1}{t} \quad (\text{IV.4})$$

$$\mu^t = \frac{1}{\sqrt{t}} \quad (\text{IV.5})$$

$$\mu^t = \frac{1}{t^2} \quad (\text{IV.6})$$

La carte topologique est fixée à $\tau=5 \times 5$, le nombre de groupe à $nc=5$ et la fonction de voisinage est donnée par : $K^T(\delta) = \exp\left(-\frac{|\delta|}{T}\right)$

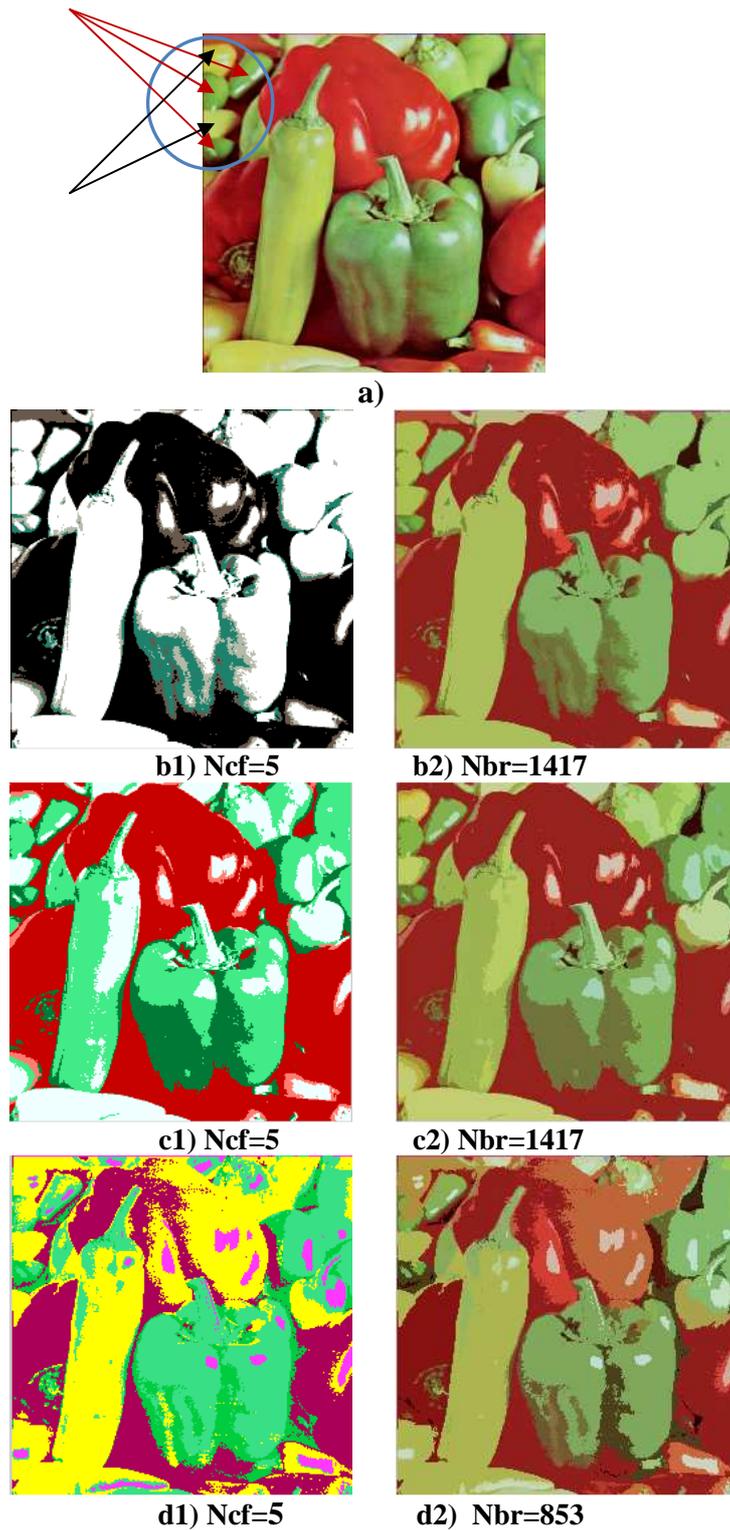


Figure IV.4. : Segmentation de l'image peppers pour $\tau=5 \times 5$, $nc=5$ et $K^T(\delta) = \exp(-\frac{|\delta|}{T})$

- a) Image initiale peppers b) résultats de la segmentation en considérant la fonction (IV.4)
 c) résultats de la segmentation en considérant la fonction (IV.5) d) résultats de la segmentation en considérant la fonction (IV.6)
- 1) affichage des résultats en fausses couleur après classification 2) affichage des résultats en régions connexes.

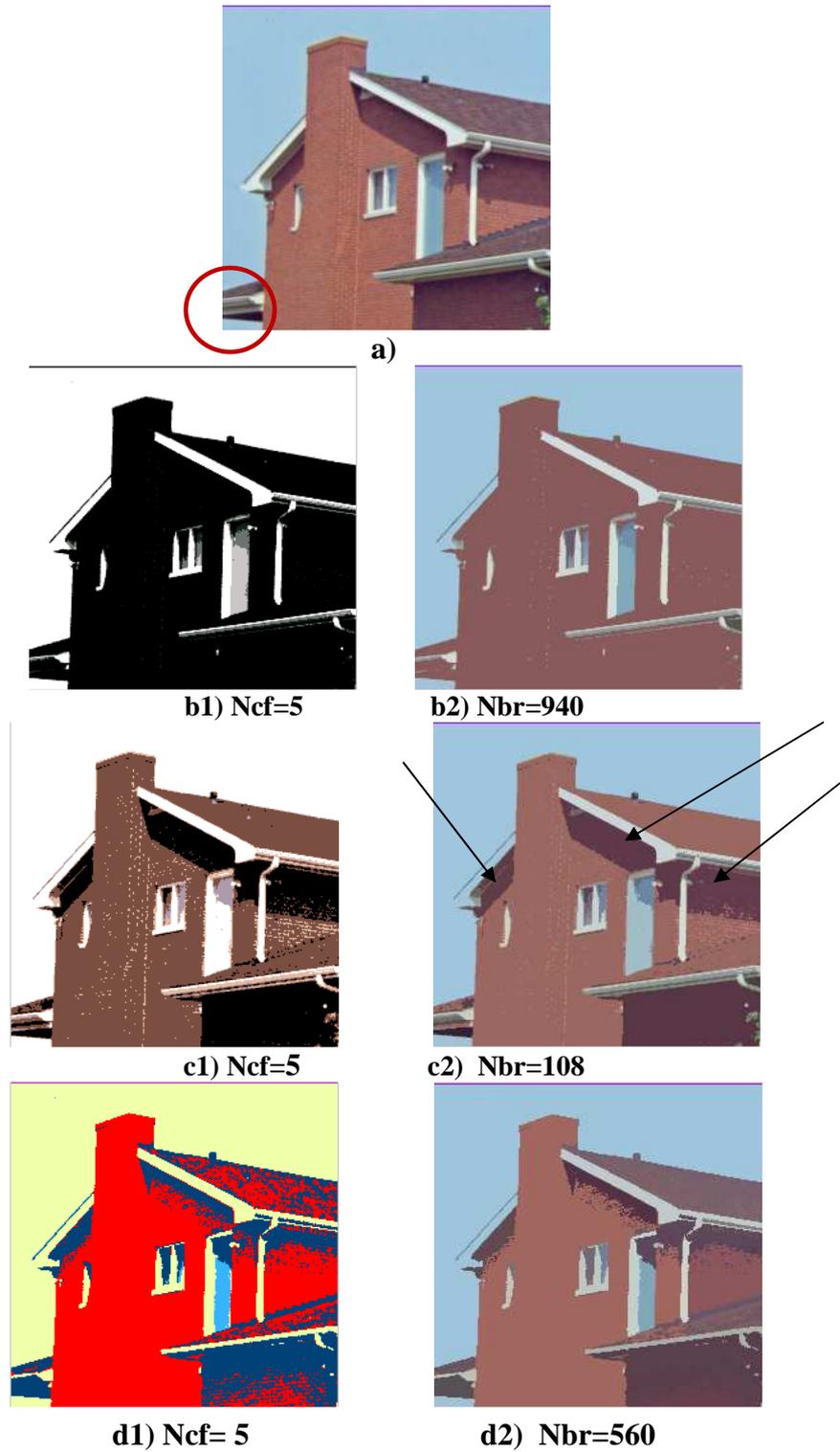


Figure IV.5. : Segmentation de l'image house pour $\tau=5 \times 5$, $nc=5$ et $K^T(\delta) = \exp(-\frac{|\delta|}{T})$

- a) Image initiale house b) résultats de la segmentation en considérant la fonction (IV.4)
 c) résultats de la segmentation en considérant la fonction (IV.5) d) résultats de la segmentation en considérant la fonction (IV.6)
 1) affichage des résultats en fausses couleurs après classification 2) affichage des résultats en régions connexes.

Les trois fonctions (pas du gradient) fournissent des résultats appréciables néanmoins la fonction $\mu^t = \frac{1}{\sqrt{t}}$ donne à vue d'œil de meilleurs résultats. En effet si nous observons la région encerclée en rouge sur l'image house, nous remarquons bien qu'elle est mieux segmentée dans le cas c2). Les ombres de la toiture sont aussi nettement mieux détectées dans ce cas, elles sont indiquées par des flèches sur la figure.

Nous constatons aussi que pour l'image peppers, la fonction $\mu^t = \frac{1}{\sqrt{t}}$ fournit de meilleurs résultats. Par exemple, la segmentation de la région encerclée en bleu est meilleur en c2) de la figure (IV.4) car dans ce cas les poivrons indiquée par les flèches rouges sont bien séparés des poivrons indiqués par les flèches noires, ce n'est pas le cas avec les autres fonctions.

Pour ces différentes raisons notre choix se porte sur la fonction $\mu^t = \frac{1}{\sqrt{t}}$ pour le pas du gradient.

IV.5. Influence de la taille τ de la carte topologique sur la segmentation des images

Dans cette partie, nous proposons d'étudier l'influence de τ sur la segmentation des différentes images. Nous considérons quatre exemples : $\tau=5 \times 5$, $\tau=7 \times 7$ et $\tau=10 \times 10$. Ainsi nous aurons respectivement une carte topologique de 25, 49 et 100 neurones.

La fonction de voisinage est donnée par : $K^T(\delta) = \exp\left(-\frac{|\delta|}{T}\right)$

Le pas du gradient est : $\mu^t = \frac{1}{\sqrt{t}}$

Le nombre de groupe est : $nc=5$.

Quelques résultats sont illustrer sur les figures suivantes.



Figure IV.6. : Segmentation de l'image portrait pour $K^T(\delta) = \exp(-\frac{|\delta|}{T})$, $nc=5$ et $\mu^t = \frac{1}{\sqrt{t}}$

- a) Image initiale portrait b) résultats de la segmentation en considérant $\tau=5 \times 5$ c) résultats de la segmentation en considérant $\tau=7 \times 7$.d) résultats de la segmentation en considérant $\tau=10 \times 10$
 1) affichage des résultats en fausses couleur après classification 2) affichage des résultats en régions connexes.

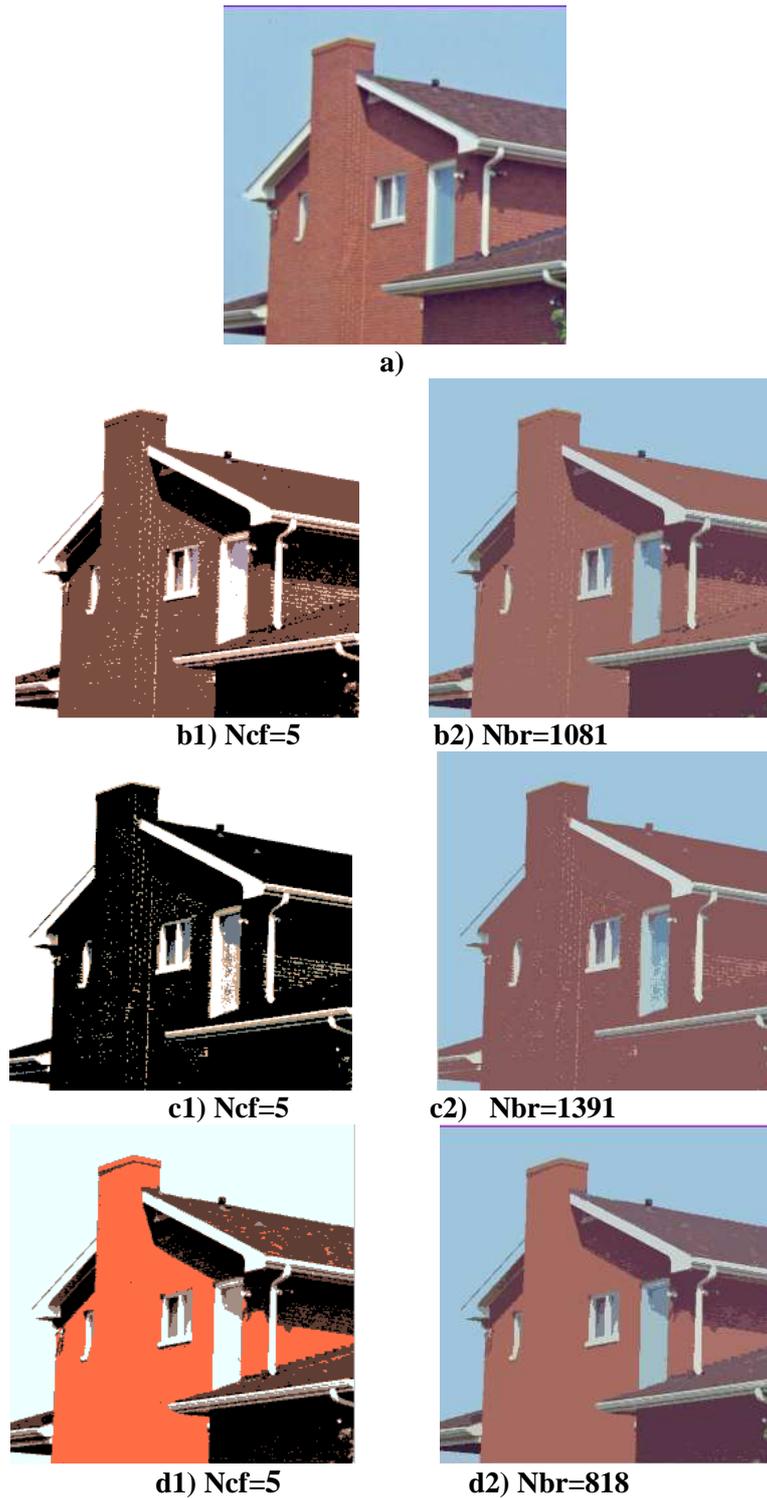


Figure IV.7. : Segmentation de l'image house pour $K^T(\delta) = \exp(-\frac{|\delta|}{T})$, $nc=5$ et $\mu^t = \frac{1}{\sqrt{t}}$

- a) Image initiale house b) résultats de la segmentation en considérant $\tau=5 \times 5$ c) résultats de la segmentation en considérant $\tau=7 \times 7$.d) résultats de la segmentation en considérant $\tau=10 \times 10$
 1) affichage des résultats en fausses couleur après classification 2) affichage des résultats en régions connexes.

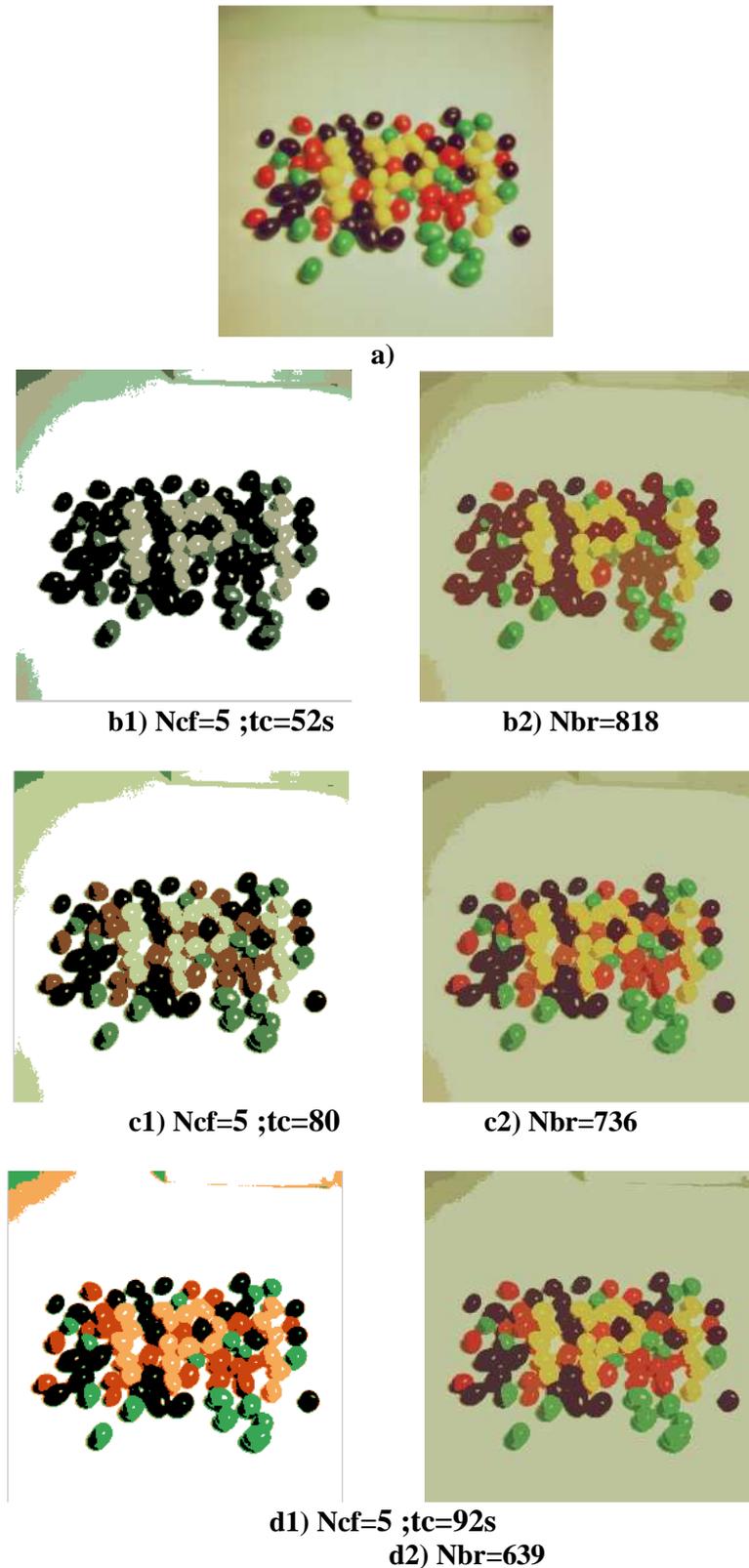


Figure IV.8. : Segmentation de l'image house pour $K^T(\delta) = \exp(-\frac{|\delta|}{T})$, $nc=5$ et $\mu' = \frac{1}{\sqrt{t}}$

a)Image initiale jelly b) résultats de la segmentation en considérant $\tau=5 \times 5$ c) résultats de la segmentation en considérant $\tau=7 \times 7$ d) résultats de la segmentation en considérant $\tau=10 \times 10$
 1) affichage des résultats en fausses couleur après classification 2) affichage des résultats en régions connexes.

En observant les figures IV.6, IV.7 et IV.8 nous constatons que plus la taille de la carte topologique augmente, meilleurs sont les résultats. En effet, par exemple pour la figure IV.6, lorsque la taille de la carte est de $\tau=5 \times 5$, une partie des cheveux du personnage est confondue avec le fond de la scène. Ce problème ne se pose pas pour des tailles plus élevées. Pour les images house et jelly, nous obtenons de bons résultats pour une taille de 10×10 . Ce résultat nous paraît prévisible car les différentes nuances dans l'image seront mieux séparées lorsque le nombre de neurones augmente. Nous pouvons augmenter ce nombre indéfiniment mais au détriment du temps de calcul. En effet pour une taille $\tau=5 \times 5$ le temps de calcul est en moyenne de 50 s, pour une taille $\tau=7 \times 7$, ce temps est de 85 s, pour $\tau=10 \times 10$, il est de 94 s et pour $\tau=50 \times 50$, il est de 30mn. En augmentant la taille de la carte, nous augmentons le temps de calcul sans pour autant apporter de nettes améliorations aux résultats au delà de la taille 10×10 . Afin d'illustrer cela, nous donnons un exemple pour l'image portrait sur la figure IV.9 pour une taille de la carte topologique de 50×50 . Pour faire un compromis entre temps de calcul et résultats, nous choisissons une taille de $\tau=10 \times 10$, ce qui correspond à 100 neurones dans le réseau.

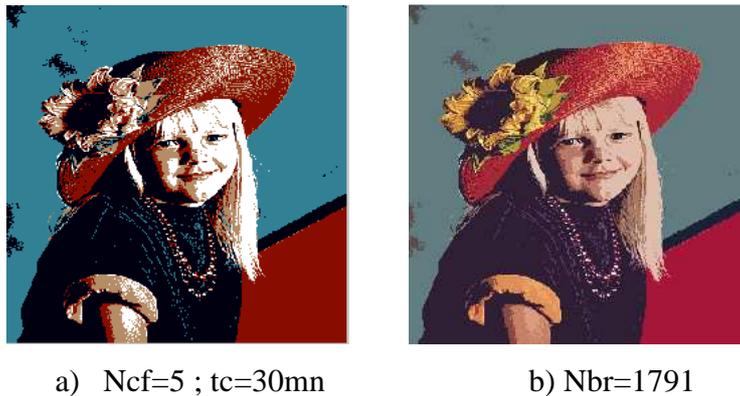


Figure IV.9 : Segmentation de l'image portrait pour $K^T(\delta) = \exp\left(-\frac{|\delta|}{T}\right)$, $nc=5$ et $\mu^t = \frac{1}{\sqrt{t}}$

a) résultats en fausses couleur après classification pour $\tau=50 \times 50$ b) résultats en régions connexes.

IV.6. Influence du nombre de groupes nc sur la segmentation des images

A la fin de l'exécution de l'algorithme de Kohonen, nous obtenons sur la carte topologique τ référents finaux. Il faut alors opérer un groupement de ces référents pour réduire le nombre de classes dans l'image. Nous regroupons tous les référents adjacents (c'est-à-dire ayant une distance égale à 1) de manière hiérarchique ascendante. Le nombre de groupes finaux obtenus est noté nc et il est imposé par l'utilisateur.

Dans cette partie, nous proposons d'étudier l'influence du nombre de groupes choisies sur la segmentation des différentes images. Nous considérons trois exemples : $nc=5$, $nc=10$ et $nc=20$.

La fonction de voisinage est donnée par : $K^T(\delta) = \exp(-\frac{|\delta|}{T})$.

Le pas du gradient est : $\mu^t = \frac{1}{\sqrt{t}}$.

La taille de la carte est de : $\tau=10 \times 10$.

Quelques résultats en faisant varier nc sont illustrés sur les figures ci-dessous.



Figure IV.10.: Segmentation de l'image portrait pour $K^T(\delta) = \exp(-\frac{|\delta|}{T})$, taille $\tau=10 \times 10$ et

$$\mu^t = \frac{1}{\sqrt{t}}$$

a) Image initiale portrait b) résultats de la segmentation en considérant $n_c=5$ c) résultats de la segmentation en considérant $n_c=10$ d) résultats de la segmentation en considérant $n_c=20$
 1) affichage des résultats en fausses couleurs après classification 2) affichage des résultats en régions connexes.

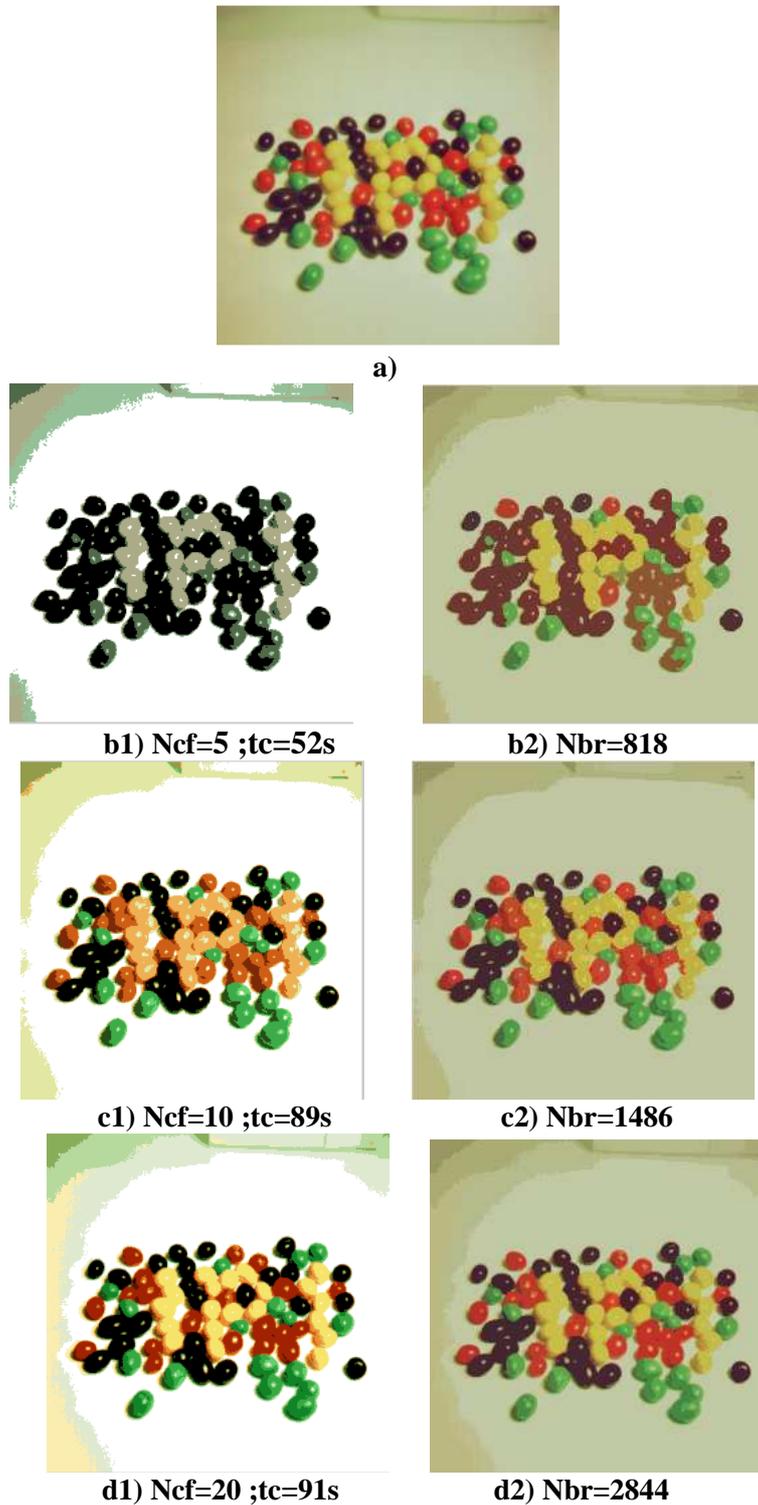


Figure IV.11. : Segmentation de l'image jelly_beans pour $K^T(\delta) = \exp(-\frac{|\delta|}{T})$, taille $\tau=10 \times 10$

$$\mu^t = \frac{1}{\sqrt{t}}$$

a) Image initiale jelly b) résultats de la segmentation en considérant $nc=5$ c) résultats de la segmentation en considérant $nc=10$.d) résultats de la segmentation en considérant $nc=20$
 1) affichage des résultats en fausses couleur après classification 2) affichage des résultats en régions connexes.

Les résultats obtenus sont prévisibles. En effet plus la valeur de n_c est élevée, plus le nombre de classes réels détectées est grand (il est au plus égale à n_c) et meilleurs seront les résultats. Si ce nombre est faible, c'est-à-dire s'il est inférieur au nombre de classes présentes dans l'image, plusieurs classes différentes seront considérées comme étant la même. Il est aussi inutile de prendre n_c très élevé car cela risque de sur-segmenter l'image. Il faut donc bien choisir ce paramètre.

IV.7. Conclusion

Dans ce chapitre nous avons présenté différents résultats de l'application des réseaux SOM, à la classification des images couleur. Il nous a paru utile d'afficher les images classifiées ainsi que les images segmentées afin de mieux évaluer les dissimilarités et ce en variant les différents paramètres.

Nous avons ainsi sélectionné la fonction de voisinage K^T , le pas du gradient μ^t , la taille de la carte τ et le nombre de groupes n_c qui fournissent relativement à nos expériences, de meilleurs résultats.

Conclusion générale

Le travail qui nous a été proposé est l'utilisation d'une méthode de classification non supervisée pour la segmentation des images couleur.

Ce travail nous a permis de nous familiariser avec plusieurs domaines à savoir le traitement d'images, les réseaux de neurones et la programmation sous Matlab.

Nous avons donc implémenté sous Matlab un algorithme de segmentation d'image par les cartes auto-organisatrice de Kohonen qui fait partie de la famille des réseaux de neurones.

Rappelons que cette méthode cherche d'abord à affecter les observations (les différentes valeurs des pixels) aux différents neurones de la carte topologique puis regrouper ces derniers dans des classes après avoir trouver les valeurs des référents. Nous avons choisie de réaliser le regroupement des neurones à l'aide la classification ascendante hiérarchique.

La méthode que nous avons implémenté, nous paraît assez robuste puisque la modification de certains paramètres n'influe que faiblement sur les résultats. Le choix de quelques paramètres est néanmoins primordial pour avoir de bons résultats de segmentation. Par exemple, le nombre de groupes doit être supérieure ou au moins égale au nombre de classes réelles présentes dans l'image. Dans le cas contraire, nous classerons dans une même classe des objets appartenant à des classes différentes.

Nous avons appliqué la méthode implémentée sur des images couleur en utilisant les trois composantes colorimétriques RVB. La segmentation réalisée est alors vectorielle, prenant ainsi en considération les interactions colorimétriques des composantes couleurs. Les résultats obtenus sont appréciables, néanmoins il nous paraît intéressant de refaire les tests sur les composantes des différents systèmes de représentation des images couleur tel que les systèmes $L^*u^*v^*$, $L^*a^*b^*$, XYZ, HSV, etc.

Bibliographie

- [1] J. P. COCQUEREZ et S. PHILIPP, Analyse d'images, filtrage et segmentation, Masson, Paris 1995.
- [2] http://tecfa.unige.ch/~lombardf/CPTIC/couleurs/couleur_ERAG/Base.htm).
- [3] A. TREMEAU, C. FERNANDEZ-MALOIGNE et P. BONTON, Image Numérique Couleur De L'acquisition Au Traitement, Dunod, Paris 2004
- [4] I.T Young, Fundamentals of image processing, Université de Delft.
- [5] R.DERICHE et O.FAUGERA, Les edp en traitement d'images et vision par ordinateur, tech. Rep,2697 , INRIA Technical Report, France, 1995 .
- [6] Y .I.OHTA, T.KANADE, AND T.SAKAI, Color information for region segmentation, Computer Graphics and Image Processing,vol,13,1980.
- [7] N.VANDERBROUKE : Segmentation d'image couleur par classification de pixel dans des espaces d'attributs colorimétriques adaptés, Ph. D. thesis, université de Lille1, France, Décembre 2000.
- [8] O.LOZORAY : Segmentation d'images couleur par morphologie mathématique en classification de données par réseaux de neurones, application à la classification de cellules d'objets colorés, Ph. D. thesis, université de Caen, France, Janvier 2000.
- [9] S. OUADFEL : Contributions à la segmentation d'images basées sur la résolution collective par colonies de fourmis artificielles, thèse de Doctorat en informatique, Université Hadj Lakhdar de Batna, Faculté des Sciences de l'Ingénieur, 2006
- [10] L. IFTICENE, M. BEN CHABANE et L. CHERFA : Application des réseaux de neurones à fonction radiale de base pour la classification d'arythmies cardiaques, Thèse d'ingénieur en électronique, Université Mouloud Mammeri de Tizi-Ouzou, promotion 2004-2005.
- [11] B. VIROLE : Réseaux de neurones et psychométrie, Editions du Centre de Psychologie Appliquée, Juin 2001
- [12] M. PARIZEAU : Réseaux de neurones, Université LAVAL, 2004.
- [13] R. AMAOUCHE, M. AGOUN et R. LOUDA : Utilisation des réseaux de neurones pour la prévision météorologique : application aux données satellitaires, Thèse d'ingénieur en électronique, Université Mouloud Mammeri de Tizi-Ouzou, promotion 2006-2007.
- [14] www.lesrobots.com/robots/neurones_artificiels.htm
- [15] www.techno-science.net

- [16] F. MOUTARDE, CAOR, MINES ParisTech : Introduction aux réseaux de neurones
- [17] G. DREYFUS, J. M. MARTINEZ, M. SAMUELIDES, M. B. GORDON, F. BARDRAN, S. THIRIA et L. HERAULT, Réseaux de neurones : méthodologie et application, Edition EYROLLES, 2004
- [18] K. HAMMOUCHE, Méthodes de classification automatique utilisant la notion de texture, Thèse de doctorat d'état en automatique, Université Mouloud Mammeri de Tizi-Ouzou, promotion , Novembre 2007.
- [19] Z. ABBAS, K. BOUSSAAD, Algorithme génétique, Thèse d'ingénieur en automatique, Université Mouloud Mammeri de Tizi-Ouzou, promotion 2006-2007. en classification automatique.
- [20] A. DEBIANE, Approche connexionniste pour la classification automatique, appliquée à la segmentation d'images, Thèse d'ingénieur informatique, Université Mouloud Mammeri de Tizi-Ouzou, promotion 2000-2001.
- [21] T.CARRON, segmentation d'images couleur dans la base Teinte-Luminance-Saturation, approche numérique et symbolique, Ph. D. thesis, LAMII, université de Savoie, Annecy, France, Décembre 1995 ;
- [22] D. AMMOUR. A. BELHADJ, prétraitement des images couleur, Thèse d'ingénieur en automatique, Université Mouloud Mammeri de Tizi-Ouzou, promotion 2001-2002

