Chapitre 1

État de l'art

« ...ce que nous voulons, c'est une machine qui peut apprendre de l'expérience ». A. Turing.

Introduction

Dans le but de faciliter la compréhension de notre solution dans le chapitre prochain, nous introduisons dans ce chapitre les differentes notions de base employé, ainsi, nous présentons les méthodes existantes pour les differents phases, de la détection du visage jusqu'à la classification non supervisé.

1.1 Méthodes de détection de visage

La détection de visage est un type d'application classé dans la technologie de vision par ordinateur. C'est le processus aux cours duquel des algorithmes sont développés et formés pour localiser correctement les visages ou les objets. Celles-ci peuvent être en tems réel à partir d'une caméra vidéo. Afin de détecter la position des visages dans les images de sorte à obtenir une région d'intérêt sur laquelle l'extraction des vecteurs de caractéristiques pourra être accomplie il y a bien de nombreuses méthodes, dans ce qui suit nous allons présenter les méthodes les plus utilisées dans la détection.

1.1.1 Détection avec la méthode Haar cascade

Les cascades de Haar est une méthode utilisée pour la détection d'objet de classificateurs en cascade basés sur les fonctionnalités Haar, c'est une méthode efficace de détection d'objets proposée par Paul Viola et Michel Jones en 2001. Il s'agit d'une approche basée sur l'apprentissage automatique.

La fonction cascade est formée à partir de nombreuses images positives et négatives. Il est par la suite utilisé pour détecter des objets dans d'autres images. Cette méthode travaille avec la détection de visage. Initialement, l'algorithme nécessite beaucoup d'images positives (image de visage) et d'images négatives (images sans visages) pour former le classifieur. Ensuite, extraire les caractéristiques, et pour cela, les fonctionnalités de Haar présentées dans la figure ci-dessous sont utilisées. Chaque caractéristique est une valeur unique obtenue par la soustraction de la somme des pixels sous le rectangle blanc de la somme des pixels sous le rectangle noir.

(a) Caractéristiques du bord (b) Caractéristiques de ligne (c) Caractéristiques quatre rectangles

FIGURE 1.1: Caractéristiques de Haar

Cette méthode consiste à parcourir l'intégralité de l'image pour extraire un certain nombre de caractéristiques comme ça permet de réduire le nombre de calculs relatif à un pixel donné à une opération ne nécessitant que quatre pixels. Parmi les caractéristiques

calculées, il y'en a celles qui sont sans importance, pour sélectionner que le nombre de caractéristiques importantes Haar Cascade utilise l'algorithme d'apprentissage Adaboost pour à la fin obtenir un résultat efficace des classifieurs.

Une fois le visage détecté, il faut en extraire les saillances. L'utilisation des caractéristiques rectangulaires de cette méthode permet de détecter les yeux ou la bouche sur un visage, et cette fonction est disponible dans OpenCV. Cette méthode a résolue beaucoup de problèmes rencontrés auparavant dans la détection de visage tel que, cette méthode fonctionne presque en temps réel sur le processeur, elle en dispose d'une architecture simple et elle détecte les visages à différentes échelles. Cependant cette méthode reste imprécise, elle donne beaucoup de fausses prédictions, elle ne fonctionne pas sur les images non frontales et elle ne fonctionne pas sous occlusion.

1.1.2 Détection avec les réseaux de neurones profonds (DNN)

La méthode DNN ou Deep Neural Network (réseau de neurones profond) est une méthode de détection de visage plus performante basée sur l'apprentissage profond aussi appelé deep-learning [**?**]. Elle repose sur l'apprentissage, par un ordinateur d'un modèle de données servant à remplir une tâche. Pour ce faire, un réseau de neurones est modélisé dans l'ordinateur. Celui-ci reçoit des images les traite en vue de leur classification par le biais de modèle de l'apprentissage profond formé. Cette méthode donne en sortie une matrice à quatre dimensions telle que, la troisième dimension itère sur les visages détectés tandis que la quatrième dimension contient des informations sur le cadre de la sélection et le score de chaque visage. Les coordonnées de la sortie du cadre de sélection sont normalisées entre [0,1]. Ainsi, les coordonnées doivent être multipliées par la hauteur et la largeur de l'image d'origine pour obtenir le cadre de sélection correct sur l'image. La méthode DNN surmonte tous les défauts de la détection avec la méthode à cascade de Haar, sans compromettre aucun avantage fourni par Haar. En plus de ça, c'est la méthode la plus précise dans la détection, elle fonctionne en temps réel sur le processeur ainsi elle détecte les visages à différentes échelles (visage grand et petit) et elle fonctionne pour les différentes orientations du visage (haut, bas, gauche, droite, côté, etc) voire même sous occlusion importante. Actuellement le méthode DNN ne présente aucun inconvénient majeur d'après les dernières recherches faites en fin 2018 par rapport à toutes les méthodes de détection de visage dans OpenCV sauf qu'elle est plus au moins lente que celle basée sur Dlib HOG.

1.1.3 Détection avec la l'Histogrammes de gradients orientées (HOG)

Il s'agit d'un modèle de détection de visage largement utilisé, inventé en 2005, basé sur les fonctionnalités HOG (Histogram of Oriented Gradients) et SVM (Support Vector Machine). Dans le descripteur de caractéristiques HOG, la distribution (histogrammes) des directions de gradients (gradients orientés) est utilisé comme caractéristique. Le calcul de l'histogramme des gradients se déroule en différentes étapes.

Tout abord, nous allons mettre l'image en noir et blanc, nous enlevons toutes les autres couleurs car nous n'avons pas besoins de couleurs pour trouver un visage. Ensuite, sélectionner un patch d'image de l'image initiale et redimensionner ce patch de l'image à une

taille plus petite dans le but de regarder chaque pixel de notre image. Pour chaque pixel, nous voulons regarder les pixels qui l'entourent directement. Le but est de déterminer l'obscurité du pixel courant par rapport aux pixels qui l'entourent. Ensuite dessiner une flèche pour montrer dans quelle direction l'image devient plus sombre comme le montre la figure ci-dessous :

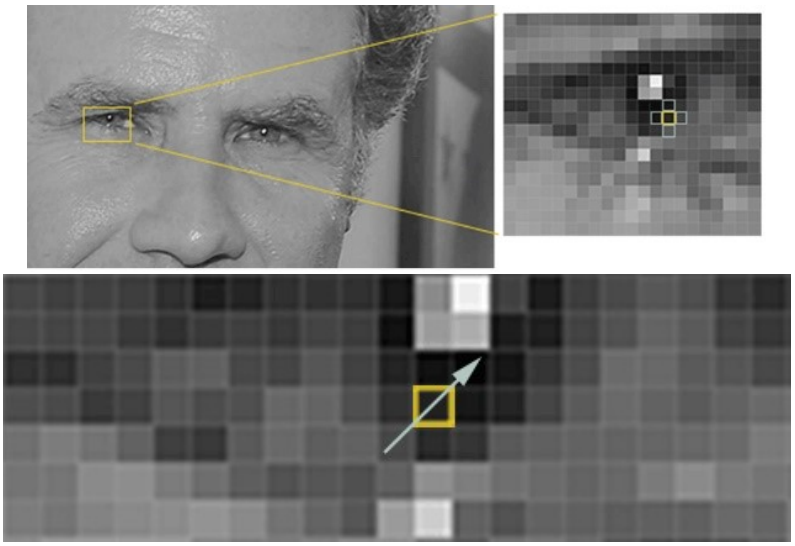


FIGURE 1.2: construction du gradient orienté

En répétant ce processus, chaque pixel sera remplacé par une flèche. Ces flèches sont appelées gradients et elles montrent l'écoulement de la lumière à l'obscurité sur toute l'image.

La raison majeure pour laquelle cette méthode remplace les pixels par des flèches est pour avoir une représentation exacte des images. En analysant directement les pixels, pour la même personne, les images vraiment sombres et les images vraiment claires auront des valeurs de pixels totalement différentes. En revanche, en considérant seulement la direction dans laquelle la direction change, nous arrivons à une représentation exacte.

Cependant, sauvegarder le gradient pour chaque pixel nous donne beaucoup de détails et cela fait beaucoup de données, pour cela, nous allons découper l'image en petits carrés de 16x16 pixels chacun. Et pout chaque case, on comptera combien de pentes dans chaque direction principale (vers le haut, haut droit, haut gauche, etc.. . .). Ensuite, nous remplaçons ce carré dans l'image par la direction des flèches. A la fin, on arrive à transformer l'image originale en une représentation très simple qui capture la structure de base d'un visage de manière très simple.

A la fin, on calcule le vecteur de caractéristiques HOG, ce dernier est un géant vecteur qui concatène plusieurs vecteurs de caractéristique de l'ensemble de l'image pour chaque carré de cette image comme il est illustré dans la figure 1.3 ci-aprés.

La dernière étape consiste à calculer le vecteur de caractéristique HOG, pour calculer le dernier vecteur de caractéristique pour l'ensemble du bloc d'image, les vecteurs de 36 x 1 sont concaténés en un seul vecteur géant.

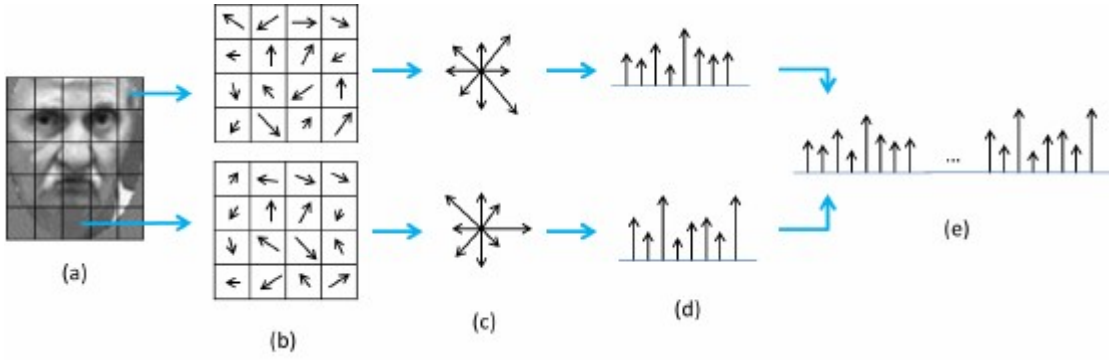


FIGURE 1.3: Processus de la construction de l'histogramme des gradients

La méthode HOG est la méthode la plus rapide sur le processeur, elle fonctionne très bien pour mes faces frontales et légèrement non frontales, son modèle est léger par rapport aux méthodes précédentes et elle fonctionne sous petit occlusion. Cependant, l'inconvénient majeur de cette méthode est qu'elle ne détecte pas les petits visages, car elle est conçue pour une taille minimale de 80 x 80, la boite englobante exclut souvent une partie du front et même une partie menton parfois, elle ne fonctionne pas très bien sous occlusion importante et elle ne fonctionne pas pour les faces latérales et les faces extrêmes non frontales, comme par exemple regarder vers le bas ou vers le haut.

1.1.4 Détection avec les réseaux de neurones convolutionnel (CNN)

C'est une méthode qui est aussi très utilisée dans cette dernière décennie, à cause de sa capacité et la perfection dans l'extraction qui a permet aux utilisateurs de se libérer des tâches fastidieuses telles que l'extraction minière négative. Cette méthode repose sur un algorithme de détection d'objets à marge maximale MMOD (Max Margin Object Detection en anglais) Pour Dlib. Ce modèle produit des détecteurs de haute qualité à partir de quantités relativement faibles de donnée d'entrainement (à partir de 4 images seulement) en utilisant un ensemble de données étiqueté manuellement par son auteur. Cependant, l'implémentation MMOD utilise l'extraction de caractéristiques HOG suivie d'un filtre linéaire unique. Cela signifie qu'il capable d'apprendre à détecter des objets présentant une variation complexe de pose ou une grande variété d'apparences, ce qui a conduit à utiliser au cours des dernières années, les réseaux de neurones convolutifs CNN [**?**] capable de traiter tous ces problèmes dans un seul modèle. L'idée est donc d'ajouter une implémentation de MMOD avec l'extraction de la fonctionnalité HOG remplacée par un réseau de neurones convolutifs. Le résultat obtenu avec la version CNN de MMOD était inattendu en travaillant seulement avec 4 images étant donné que d'autres méthodes de Deep-learning nécessitent généralement plusieurs milliers d'images. Les images suivantes montrent la différence de capacité dans la détection entre le nouveau détecteur CNN et le détecteur de Dlib par défaut HOG telles que, les cases rouges correspondent aux détections CNN et les cases bleues aux détections HOG.

Une différence notable entre les deux modèles de détection, HOG fait un excellent travail sur les visages faciles en regardant la caméra, mais seulement en étant directement

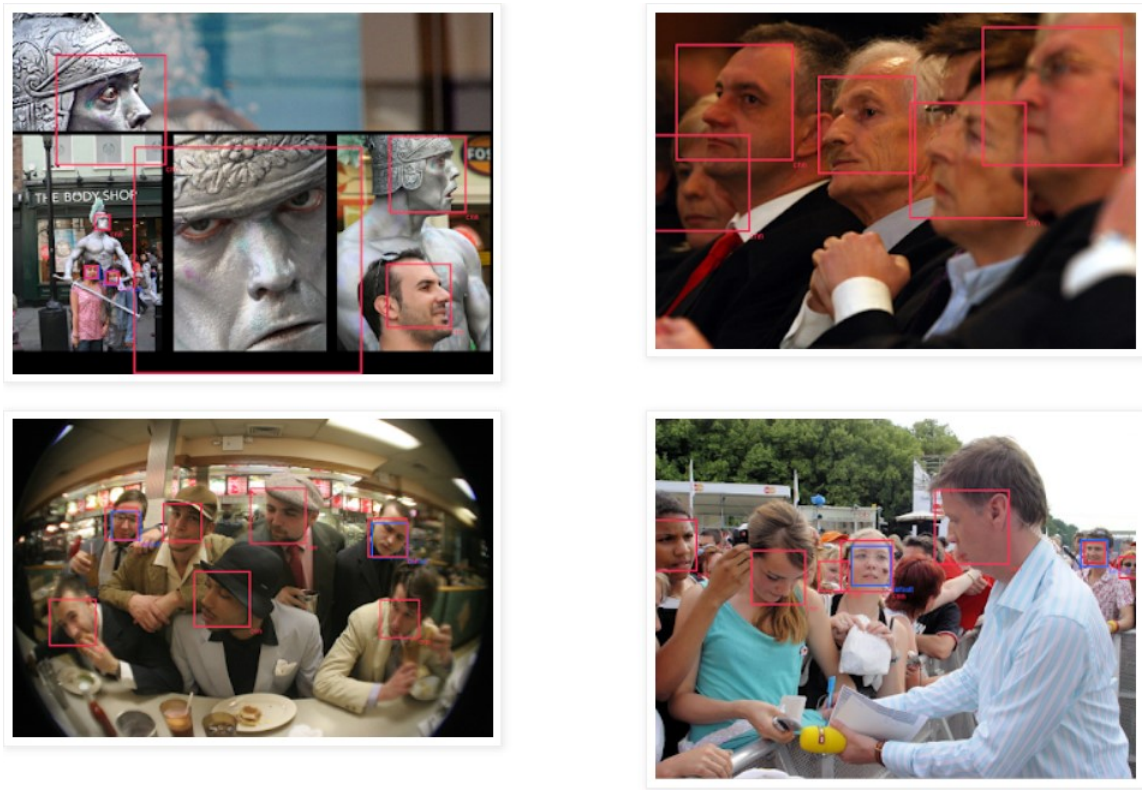


FIGURE 1.4: visages détectés par le détecteur HOG et CNN

face à la caméra, cependant e détecteur CNN est bien meilleur non seulement pour traiter les cas facile mais tous les visages en général. Il est aussi robuste à l'occlusion, rapide sur les GPUs (45 ms par images) et son processus de formation est très facile. En revanche, ce modèle reste lent sur mes processus (370 ms pour traiter une seule image), il entrainé pour une taille minimale de 80 x 80, ce qui fait qu'il ne détecte pas les petits visages et sa boite englobante est encire plus petite que le détecteur HOG.

1.2 Methode d'extraction des points de saillances

Le processus qui est capable d'explorer un ensemble de points clés à partir d'une image de visage donnée, est appelée Localisation du repère du visage ou Face Landmark Localization en anglais (alignement du visage).

Les repères (points clés) qui nous intéressent sont ceux qui décrivent la forme des attributs du visage comme : les yeux, les sourcils, le nez, la bouche, et le menton. Ces points ont donné un excellent aperçu de la structure faciale analysée, qui peut être très utile pour un large éventail d'applications.

De nombreuses méthodes permettent de détecter ces points : certaines d'entres elles atteignent une précision et une robuste supérieures en analysant un modèle de visage 3D extrait d'une image 2D, d'autres s'appuient sur la puissance des CNN et d'autres utilisent

des fonctions simples (mais rapides) pour estimer l'emplacement des points.

L'Algorithme de détection des repères de visage reposé par Dlib est l'implémentation de l'Ensemble of Regression Trees (ERT) présenté en 2014 par Kazemi et Sullivan [**?**]. Cette technique utilise une fonction simple et rapide (différence d'intensité des pixels) pour estimer directement la position des points de repère. Ces positions estimées sont ensuite affinées au moyen d'un processus itératif effectué par une cascade de variables explicatives. Les régresseurs produisent une nouvelle estimation à partir de la précédente, en essayant de réduire l'erreur d'alignement des points estimés à chaque itération. L'algorithme est très rapide, en fait, il faut environ 1 à 3 ms pour détecter un ensemble de 68 points de repère sur un visage donné. La figure suivante présente l'ensemble des points de visage extrait par l'algorithme :

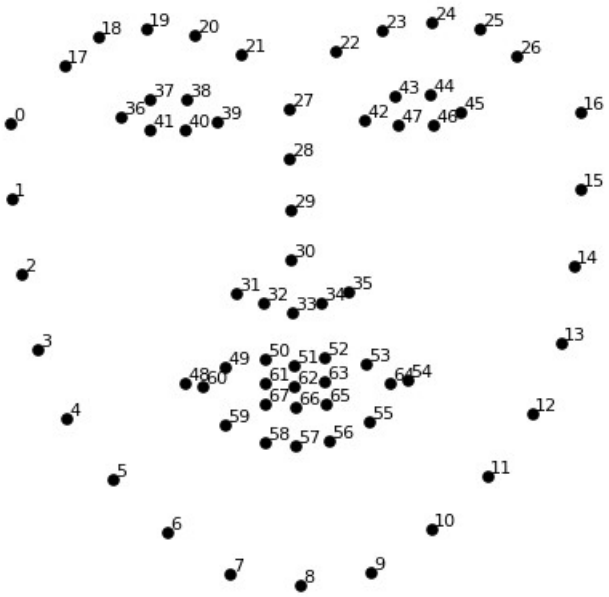


FIGURE 1.5: L'ensemble des 68 points détectés par Dlib pré-entraîné

Fondamentalement, un prédicteur de forme peut être généré à partir d'un ensemble d'images, d'annotations et d'options de formation. Une seule annotation se compose de la région du visage et des points marqués que nous voulons localiser. La région du visage peut être facilement obtenue par n'importe quel algorithme de détection de visage (OpenCV Haar Cascade, Dlib HOG Detector, CNN, ...), au lieu de cela les points doivent être marqués manuellement ou détectés par des détecteurs et modèles déjà disponible. Enfin, les options de formation sont un ensemble de paramètres qui définissent les caractéristiques du modèle formé. Ces paramètres peuvent être correctement ajustés afin d'obtenir le comportement souhaité du modèle généré.

1.3 Clustering

Le clustering ou la classification non supervisée en français, est une technique de classification très parlante en apprentissage automatique, en analyse et en fouille de donnée ainsi qu'en reconnaissance de formes. Il fait une partie intégrante de tout un processus d'analyse exploratoire de données permettant de produire ses outils de synthétisation, de prédiction, de visualisation et d'interprétation d'un ensemble d'individus (personnes, objets, processus, etc.). L'objectif est, à partir de données constituées d'un ensemble d'individus ou d'objets et d'une relation de proximité entre ceux-ci, de construire des groupes d'individus homogènes dans les sens où :

— Deux individus proches doivent appartenir à un même ensemble (groupe).
— Deux individus éloignés doivent appartenir à des groupes différents.

Afin de définir l'homogénéité d'un groupe d'observation, il est nécessaire de mesurer une ressemblance entre deux observations. D'où la notion de similarité et de dissimilarité [**?**].

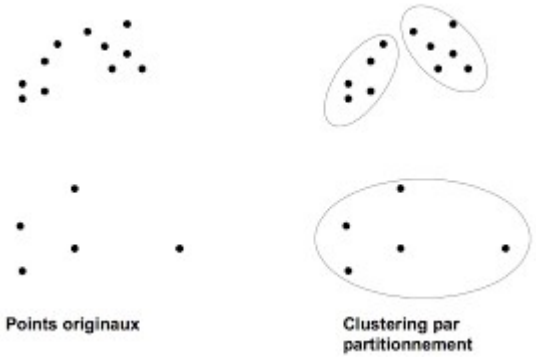


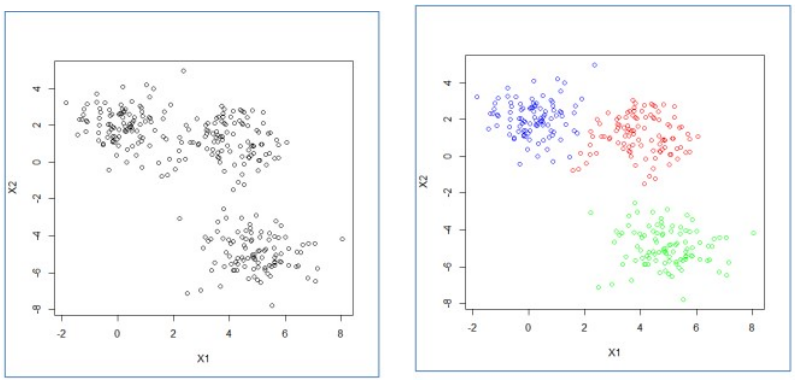
FIGURE 1.6: Clustering

Ici, les points originaux ce sont les données constituées par exemple un ensemble d'individus, le clustering les partitionne dans des groupes où chaque groupe représente un ensemble d'individus qui partagent une même caractéristique. Le clustering comprend plusieurs algorithmes de classifications pour classifier chaque point de données dans un groupe spécifique tels que : k-Means le plus algorithme connu dans la classification non supervisée, Mean-Shift un algorithme basé sur une fenêtre glissante qui tente de trouver des zones senses de points de données, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), un algorithme basé sue la densité similaire au décalage moyen, Expectation-Maximization (EM) à l'aide de Modèles de Mélange Gaussiens (GMM) et Self Organisation Map (SOM). Dans ce qui suit nous allons nous intéressés aux deux algorithmes suivent k-Means et l'algorithme Safe Organizen Map (SOM).

1.3.1 K-means

K-Means est probablement l'algorithme de classification le plus connu. C'est un algorithme facile à comprendre et à implémenter.

La figure ci-dessus montre un exemple de classification par l'algorithme k-Means, tels que les points noirs à droites sont les points de données de départs et à la fin on affecte



(a) avant le clustering (b) aprés le clustering

FIGURE 1.7: K-means clustering

chaque point dans son groupe, ici nous avons trois groupes (groupes des points bleus, verts et rouges).

Pour commencer, nous avons d'abord sélectionnés un certain nombre de classes (groupes) à utiliser en initialisant aléatoirement leurs points centraux respectifs. Dans notre exemple nous avons trois classes. Les points centraux sont des vecteurs de la même longueur que chaque vecteur de points de données.

Chaque point de données (points noirs) est classé en calculant la distance entre ce point et chaque centre de groupe, puis en classant le point dans le groupe dont le centre est le plus proche. Sur la base de ces points classés, nous recalculons le centre du groupe en prenant la moyenne de tous les vecteurs du groupe.

Nous répétons cette étape pour un nombre défini d'itérations ou jusqu'à ce que les centres de groupe ne changent plus beaucoup d'une itération à une autre. Comme il est possible de choisir d'initialiser plusieurs fois le centre de groupe de manière aléatoire, puis de sélectionner le cycle qui donne les meilleurs résultats.

K-Means a l'avantage d'être assez rapide, car nous ne faisons que calculer les distances entre les points et les centres de groupe. Très peu de calculs, sa complexité linéaire est O (n).

D'autres parts, K-Means présente quelques inconvénients. Tout d'abord, nous devons sélectionner le nombre de groupes (classes). Ce n'est pas toujours évident et idéalement avec un algorithme de classification, nous aimerions qu'il soit mieux compris pas ceux-ci, car il s'agit là d'obtenir un aperçu des données. K-Means commence également par un choix aléatoire de centre de grappes et peut donc donner différents résultats de frappes dur différentes exécutions de l'algorithme. Ainsi, les résultats peuvent ne pas être reproductible et manquer de cohérences.

1.3.2 Carte auto-organisatrice (SOM)

Self Organizen Map ou la carte auto organisatrice (SOM), est un type de réseau de neurones artificiels (RNA) formé à l'aide d'un apprentissage non supervisé afin de pro-

duire une représentation discrète, généralement bidimensionnelle, de l'espace d'entrée des échantillons d'apprentissage. La carte est donc une méthode pour faire la réduction de dimensionnalité. Les carte auto organisatrices différent des autres réseaux de neurones artificiels par le fait qu'elles appliquent l'apprentissage compétitif par opposition à l'apprentissage avec correction d'erreur et qu'elles utilisent une fonction de voisinage pour préserver les propriétés topologiques de l'espace d'entrée.

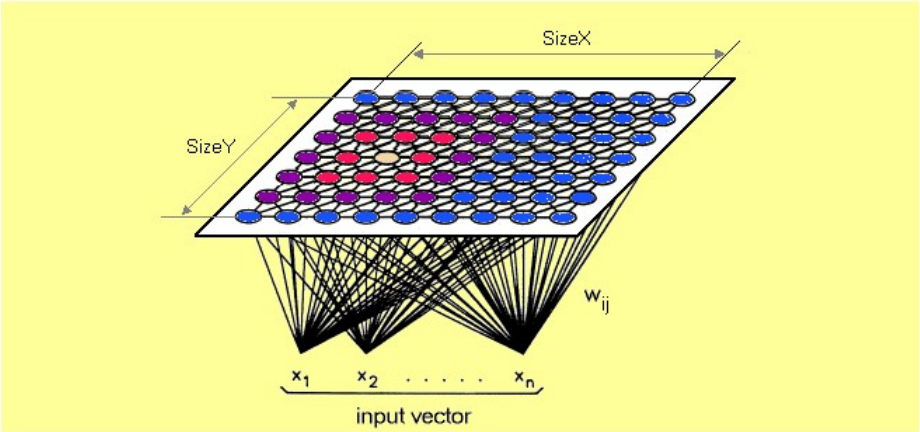


FIGURE 1.8: Réduction de la dimensionnalité dans la SOM

SOM a été introduite par le professeur finlandais Teuvo Kohonen dans les années 1980, elle est aussi appelée carte de Kohonen [**?**].

Chaque point de données dans l'ensemble des points de données se reconnait en compétition pour la représentation. Les étapes de mappage SOM commencent par l'initialisation des vecteurs de pondération. A partir de là, un vecteur d'échantillon est sélectionné de manière aléatoire et la carte des vecteurs de poids est explorée pour trouver quel poids représente le mieux cet échantillon. Chaque vecteur de poids a des poids voisins qui lui sont proches. Le poids choisi est récompensé par sa capacité de ressembler davantage à cet échantillon de vecteur sélectionné au hasard. Les voisins de ce poids sont également récompensés par leur capacité à ressembler davantage au vecteur échantillon choisi. Cela permet à la carte de s'agrandir et de former différentes formes. Plus généralement, ils forment des formes carrées, rectangulaires, hexagonales et L dans un espace de fonction 2D.

L'algorithme SOM :

1. Initialiser chaque poids de chaque nœud.
2. Choisir un vecteur au hasard dans l'ensemble des données d'apprentissages.
3. Chaque nœud est examiné pour calculer les poids qui ressemblent le plus au vecteur d'entrée. Le nœud gagnant est généralement appelé unité de meilleure correspondance ou Best Matching Unit (BMU)
4. Le poids gagnant est récompensé par le fait de ressembler davantage au vecteur échantillon. Les voisins deviennent également davantage comme le vecteur échan-

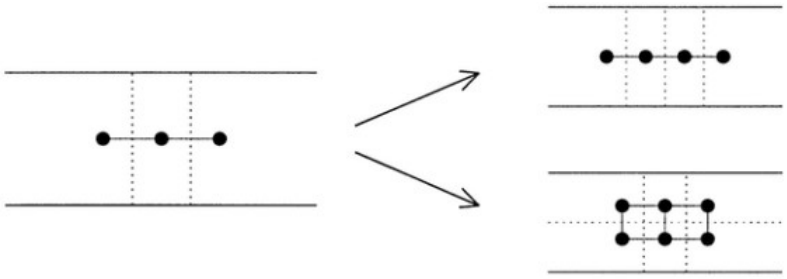


FIGURE 1.9: Illustration de la décision de base à prendre lors de la croissance

tillon. Plus le nœud est proche du BMU, plus des poids sont modifiés et plus le voisin est éloigné du BMU, moins il en apprend.

5. Répéter l'étape 2 pour N itération.

BMU est technique qui calcul la distance entre chaque poids et le vecteur échantillon en parcourant tous les vecteurs de poids. Le poids avec la distance la plus courte est le gagnant. Il existe nombreuse façon de déterminer la distance, cependant la méthode la plus couramment utilisée est la distance euclidienne.

Conclusion

Nous avons exploré, dans ce chapitre, les différents notions, existantes, liées à notre solution. Cela nous permet d'entamer les détails de notre conception dans le chapitre suivant.