

Au même titre que la texture, la forme est généralement une description très riche d'un objet.

De nombreuses solutions ont été proposées dans la littérature pour représenter une forme. Nous distinguons deux catégories de descripteurs de formes : les descripteurs basés sur les frontières (ou contours), et les descripteurs basés sur les régions.

Les premiers font référence aux descripteurs de Fourier et portent sur une caractérisation des contours de la forme. La seconde approche fait référence, entre autres, aux moments invariants qui sont utilisés pour caractériser l'intégralité de la forme région. Ces attributs sont robustes aux transformations géométriques comme la rotation et le changement d'échelle. La méthode que nous utilisons dans cette étude concerne une description de forme à base de région par le moyen de moments géométriques.

II.3.2 La représentation par les moments invariants de HU :

II.3.2.1 les moments géométriques :

Les moments géométriques [29] permettent de décrire une forme à l'aide de propriétés statistiques. Ils représentent les propriétés spatiales de la distribution des pixels dans l'image. Ils sont facilement calculés et implémentés, par contre cette approche est très sensible au bruit et aux déformations.

Les moments sont utilisés depuis longtemps pour calculer la position et le centre d'une distribution mais aussi sa variance. En vision par ordinateur, les moments permettent de calculer la position et l'orientation d'un objet.

A partir des moments géométriques, HU [28] a proposé un ensemble de sept moments invariants aux translations, rotations et changement d'échelle.

Une première utilisation des moments pour la reconnaissance de motifs géométriques a été proposée par HU [35].

Les moments offrent un cadre théorique puissant pour résoudre des problèmes rencontrés dans plusieurs applications d'imagerie y compris l'imagerie médicale, surtout lorsqu'il s'agit d'observer un objet rigide (comme les vertèbres) à différents

niveaux d'échelle, rotations et translations dans une image 2D (tomographie, projection, etc.).

Récemment, une application remarquable des moments a été présentée par [21] en combinaison avec la transformée en ondelettes pour la catégorisation des images médicales, en utilisant les cartes d'auto-organisation avec un taux de réussite de 81.8%.

II.3.2.2. Théorie des moments :

La notion de moment en mathématiques (notamment en calcul des probabilités) a pour origine la notion de moment en physique. Le moment m_{pq} d'ordre $n = p + q$ d'une fonction de distribution $f(x, y)$ est définie par l'équation (7) suivante :

$$m_{pq} = \int_{-\infty}^{-\infty} \int_{-\infty}^{-\infty} x^p y^q f(x, y) dx dy \quad (7)$$

Pour une image digitale $g(x, y)$ de taille $M \times N$, la formule ci-dessus devient :

$$m_{pq} = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} x^p y^q g(x, y) \quad (8)$$

Le moment centrale est défini par :

$$\mu_{pq} = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} (x - \bar{x})^p (y - \bar{y})^q g(x, y) \quad (9)$$

Les moments apportent différentes informations statistiques sur la forme :

- ✚ ordre 0, surface de la forme : m_{00} .
- ✚ ordre 1, centre de gravité de la forme, et est calculé par la formule (10) :

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (10)$$

Pour caractériser la forme des images, nous utilisons la famille de moments HU [35] qui sont invariants aux différents changements d'image :

Les moments de Hu sont 7 moments issus de produits et quotients des moments centrés normés d'ordre 3, définis comme suit :

Moments d'ordres 2 :

$$\Phi_1 = m_{20} + m_{02}$$

$$\Phi_2 = (m_{20} + m_{02})^2 + 4m_{11}^2$$

Moments d'ordres 3 :

$$\Phi_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2$$

$$\Phi_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2$$

$$\Phi_5 = (m_{30} - 3m_{12})(m_{30} + m_{12})[(m_{30} + m_{12})^2 + (m_{21} + m_{03})^2] - 6m_{11}(m_{30} + m_{12})(m_{21} + m_{03})$$

$$\Phi_6 = (m_{20} - m_{02})[(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] - 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03})$$

$$\Phi_7 = (3m_{21} - m_{03})(m_{30} + m_{12})[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] - 6m_{11}(m_{30} + m_{12})(m_{21} + m_{03})$$

Les six premiers moments sont invariants aux translations, aux changements d'échelles, et aux rotations ainsi qu'aux réflexions. Or, l'invariance aux réflexions peut être problématique quand il s'agit de reconnaître des images « miroirs ». C'est pourquoi M. K. Hu exploite aussi dans son système le 7^{ème} moment qui n'est pas invariant aux

réflexions. Ce dernier change de signe lorsqu'une telle transformation est appliquée à l'image et permet donc de détecter celle-ci.

II.4. Classification SVM :

L'enjeu essentiel de l'apprentissage artificiel est l'aptitude à généraliser des résultats obtenue à partir d'un échantillon limitée. Dans ce travail, nous choisissons la méthode à base des machines à vecteurs de support comme un moyen opérationnel pour ce problème.

Nous allons présenter dans cette section le fondement théorique de cette méthode ainsi que sa mise en œuvre dans notre problème de détection des régions vertèbres dans les images à rayons-X.

II.4.1 Notions sur l'apprentissage statistique :

2.4.1.1 Généralités :

D'après Mari & Napoli (1996) [1], effectuer une classification, c'est mettre en évidence, d'une part, des relations entre des objets et, d'autre part les relations entre ces objets et leurs paramètres. Il s'agit de construire une partition de l'ensemble des objets en un ensemble de classes qui soient les plus homogènes possible.

La classification, a donc deux objectifs à atteindre :

- Trouver un modèle capable de représenter la répartition des données (catégorisation).
- Définir de manière formelle l'appartenance à l'une ou l'autre des classes de toute nouvelle donnée (généralisation).

En effet la classification a pour but de réduire l'espace de recherche dans une base de données lors du processus d'identification.

En pratique, on peut rencontrer deux catégories de problèmes de classification, la classification supervisée et non-supervisée :

- **Apprentissage supervisé** : dans ce type d'apprentissage, on cherche à :
 - Estimer une fonction $f(x)$ qui est la relation entre les objets et leurs classes.
 - Les objets utilisés comme données d'apprentissages sont accompagnés par la classe à laquelle ils appartiennent.

- **Apprentissage non supervisé** : on ne cherche pas cette fois à estimer une fonction mais on cherche à regrouper les objets ayant des caractéristique commune, les objets utilisés comme données d'apprentissage sont présentés sans leur classes.

II.4.1.2 Formulation d'un problème de classification (supervisé) :

Le classificateur doit estimer une fonction $f(x)$ qui est l'estimation de la fonction qui représente la relation entre l'objet et sa catégorie. Cette fonction est appelée fonction de décision :

$$f: X \rightarrow Y$$

X: L'ensemble des objets à classifier (appelé espace d'entrée).

Y: L'ensemble des catégories (appelé espace d'arrivée).

2.4.1.3 Minimisation du risque structurel :

Deux types de données sont utilisés pour un problème d'apprentissage : les données d'entraînement (données d'origine pour calculer le modèle) et les données de test (pour évaluer la performance de généralisation du modèle).

La qualité de ce modèle est alors jugée par rapport à sa capacité à réduire l'erreur de test ou de « généralisation ». Cependant, comme le modèle n'est pas construit en utilisant l'ensemble de test, l'erreur de généralisation ne peut pas être évaluée exactement car elle dépend de la distribution de probabilité des données.

Suivant la théorie de Vapnik [2], nous supposons que les données sont générées selon une distribution de probabilité inconnue $P(x,y)$. De plus, nous supposons que les données sont indépendantes et identiquement distribuées (iid).

L'erreur moyenne commise sur toute la distribution $P(x,y)$ par la fonction $f(x)$ est donnée par:

$$R[f] = \int \frac{1}{2} Q(x) dP(x, y) \quad (11)$$

Où:

- Q : est la fonction d'erreur (erreur absolue dans le cas des SVM).
- x : est le vecteur d'entrée.
- y : est l'ensemble des classes.

Ainsi la fonction f devra être optimale : la fonction f^{opt} devra être calculée de sorte que l'erreur moyenne sur toute la distribution soit minimale.

$$f^{opt} = \arg \min_f (R[f]) \quad (12)$$

Le critère formulé dans (12) est malheureusement inutilisable en pratique. En effet, pour calculer le risque, nous devrions disposer d'une estimation de la distribution $P(x,y)$, ce qui n'est pas le cas. La seule information dont nous disposons comme évaluation de l'erreur est l'erreur d'entraînement appelée Risque empirique :

$$R_{emp} = \frac{1}{l} \sum_{i=1}^l \frac{1}{2} Q(x_i) \quad (13)$$

Où :

- l est le nombre d'objets d'entraînements.

Ceci n'est pas suffisant. La raison en est que l'on peut facilement trouver un modèle minimisant l'erreur d'entraînement mais pour lequel l'erreur de généralisation sera très grande. Donc cette dernière est liée à la famille de fonction utilisée comme modèle. Cette dépendance est nommée « risque structurel ».

Dans la théorie de l'apprentissage statistique, Vapnik et Chervonenkis ont prouvé qu'il est possible de définir une majoration du risque structurel en fonction de la famille de fonctions utilisée pour le modèle [2]. L'une de ces majorations peut être calculée en utilisant la dimension de Vapnik–Chervonenkis (dimension VC) qui représente le plus grand nombre de points pouvant être séparés de toutes les façons possibles par un membre de l'ensemble de fonctions de F . Cela veut dire qu'il doit exister une configuration de h ($=VC(F)$) points, telle que les fonctions $f \in F$ peuvent leur assigner les 2^h combinaisons des labels (classes) possibles.

Dans la section suivante, nous allons présenter une technique de classification automatique supervisée basée sur la minimisation de risque structurel : SVM.

II.4.2 Théorie des machines à vecteurs de support (SVM) :

2.4.2.1. Introduction :

Les Machines à Vecteurs de Support -SVM - [49,50] ont été largement utilisées et appliquées aux problèmes de classification [4, 8, 16, 17,44] et la régression non linéaire [36, 37, 48].

Nous nous intéressons ici à l'application des SVM dans le domaine bio-informatique et médical.

Les SVM ont été utilisées comme méthode de prédiction de cancer du sang (leucémie, lymphome, etc.) dans le contexte de la technologie des puces d'ADN (ou micro-array) [39, 42].

Liang et Lin ont prouvé l'efficacité des SVMs pour la classification des données ECG pour la détection d'une vidange gastrique tardive dans [50].

Une autre application en diagnostic médicale est présentée pour la détection des micro-calcifications dans les mammographies dans [3, 10].

II.4.2.2 Principe des machines à vecteurs de support :

Les SVM constituent une classe d'algorithmes basée sur le principe de minimisation de « Risque structurel » décrit par la théorie de l'apprentissage statistique qui utilise la séparation linéaire. Cela consiste à séparer par un hyperplan des individus représentés dans un espace de dimension égale au nombre de caractéristiques, les individus étant alors séparés en deux classes. Cela est possible quand les données à classer sont linéairement séparables. Dans le cas contraire, les données seront projetées sur un espace de plus grande dimension afin qu'elles deviennent linéairement séparables.

II.4.2.3 Cas de données linéairement séparables :

Considérons « l » points $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, $x_i \in R^N$

Avec $i=1 \dots L$ et $y_i \in \{\pm 1\}$

Ces points sont classés en utilisant une famille de fonctions linéaires définis par :

$$\langle w, x \rangle + b = 0 \quad (14)$$

Avec $w \in \mathbb{R}^N$ et $b \in \mathbb{R}$ de telle sorte que la fonction de décision concernant l'appartenance d'un point à l'une des deux classes soit donnée par :

$$f(x) = \text{sgn}(\langle w, x \rangle + b) \quad (15)$$

La fonction (14) représente l'équation de l'hyperplan H. La fonction de décision (15) va donc observer de quel côté de H se trouve l'élément de x.

On appelle la marge d'un élément la distance euclidienne prise perpendiculairement entre H et x. Si on prend un point quelconque t sur H, cette marge peut s'exprimer en :

$$M_x = \frac{w}{\|w\|} (x - t) \quad (16)$$

La marge de toutes les données est définie comme étant :

$$M = \min_{x \in E} M_x \quad (17)$$

L'approche de classification par SVM tend à maximiser cette marge pour séparer le plus clairement possible deux classes. Intuitivement, avoir une marge la plus large possible sécurise mieux le processus d'affectation d'un nouvel élément à l'une des classes.

Un SVM fait donc partie des classificateurs à marge maximale.

Un classificateur à marge maximale est un classificateur dont l'hyperplan optimal séparant deux classes est une solution du problème d'optimisation mathématique suivant (forme primale) :

$$\text{minimiser } \frac{1}{2} \|w\|^2 (\langle w, x \rangle + b) z \geq x \in E \quad (18)$$

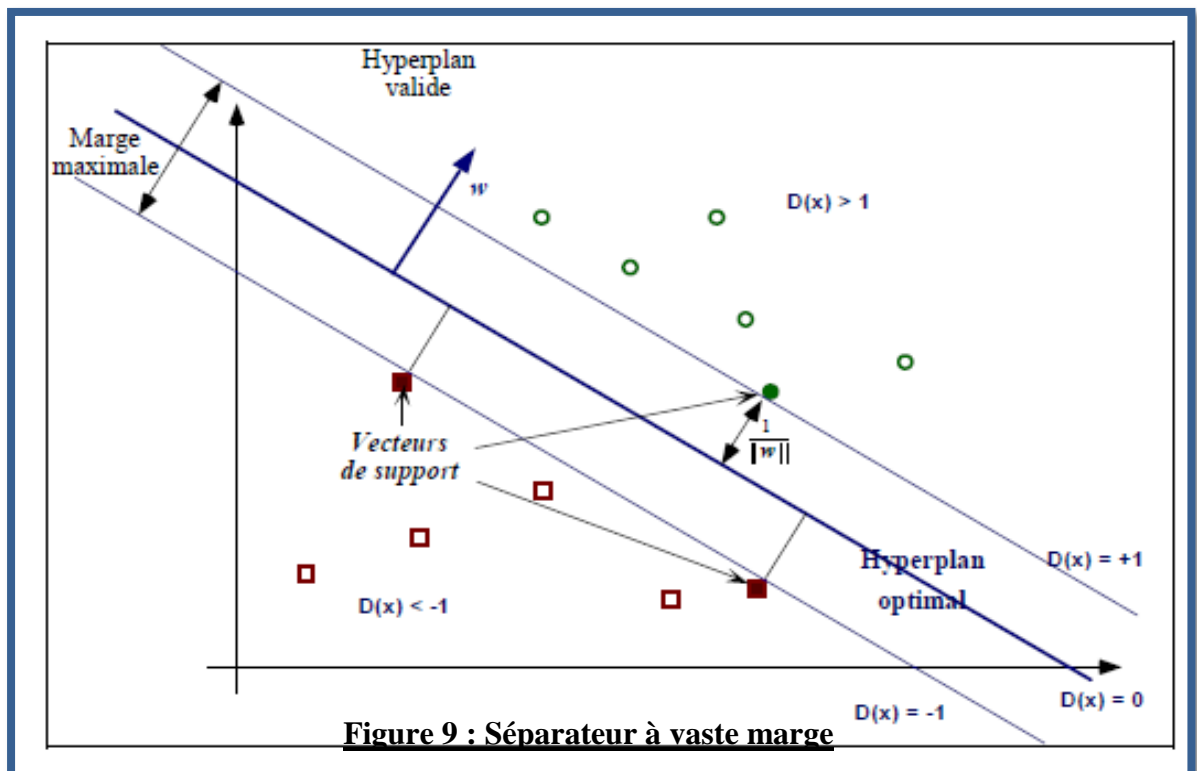
La fonction objective de ce problème est le carré de l'inverse de la double marge qu'on veut maximiser. La contrainte unique correspond au fait que les éléments x doivent être bien placés.

La résolution de ce problème nécessite de fixer les paramètres w et b qui constituent les variables α_i de la machine d'apprentissage.

Les classificateurs à marge maximale donnent de bons résultats lorsque les données sont linéairement séparables.

La tâche de discrimination est de trouver un hyperplan qui sépare deux (ou plus) ensembles de vecteurs (Figure 9).

Pour la détection et la reconnaissance des vertèbres, ces deux classes peuvent être région de vertèbre ou non.



Forme duale :

La formulation primale peut être transformée en formulation duale en utilisant les multiplicateurs de Lagrange. L'équation (18) s'écrit alors sous la forme suivante :

$$L(\mathbf{w}, \mathbf{b}, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + \mathbf{b}) - 1) \quad (19)$$

II.4.2.4 Cas des données non-linéairement séparables :

En pratique, il est assez rare d'avoir des données linéairement séparables. Afin de traiter également des données bruitées ou non linéairement séparables, les SVM ont été généralisées grâce à deux outils : la marge souple (soft margin) et les fonctions noyau (kernel functions).

Le principe de la marge souple est d'autoriser des erreurs de classification. Le nouveau problème de séparation optimale est reformulé comme suit :

$$\begin{aligned} \text{MIN}_{\mathbf{w}, \mathbf{b}, \epsilon} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{W} + C \sum_{i=1}^l \epsilon_i, \quad C \geq 0 \quad \text{sous contraintes} \quad (20) \\ & y_i (\langle \mathbf{w}, \mathbf{x} \rangle + \mathbf{b}) \geq +1 - \epsilon_i \\ & \epsilon_i \geq 0 \quad \text{pour } i = 0, \dots, l \end{aligned}$$

Un terme de pénalité est introduit dans la formule (20), Le paramètre C est défini par l'utilisateur. Il peut être interprété comme une tolérance au bruit de classificateur.

Remarque :

L'idée de base pour les données non linéairement séparables, est de projeter l'espace d'entrée (espace des données) dans un espace de plus grande dimension appelé espace de caractéristiques (feature space) afin d'obtenir une configuration linéairement séparable (à l'approximation de la marge souple près) de nos données, et d'appliquer alors l'algorithme SVM.

II.4.3 Architecture d'un classificateur SVM :

II.4.3.1 la fonction noyau :

Afin de résoudre le problème de données non linéaires, la fonction noyau joue le rôle central de liaison des vecteurs d'entrées à l'espace de caractéristiques de grande dimension. (Voir figure 10).

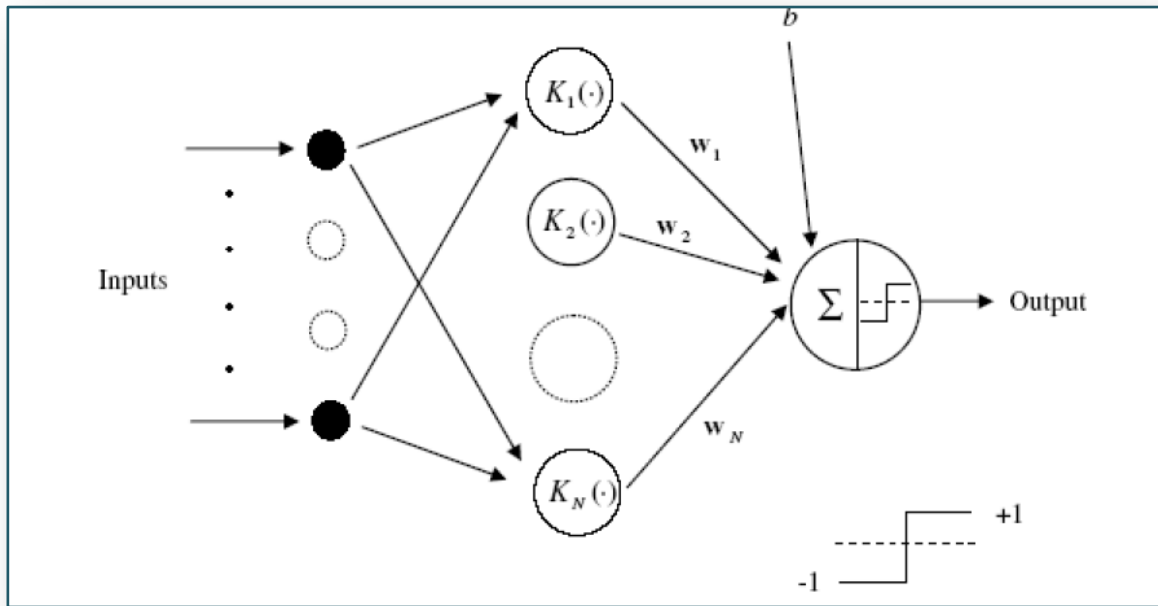


Figure 10 : Architecture d'une machine à vecteurs de support (d'un nombre N)

Les choix typiques pour la fonction noyau sont :

Noyau gaussien à base radiale (ou RBF : Radial Basis Function) :

$$K(x, x_i) = \exp[\gamma |x - x_i|] \quad (21)$$

Noyau polynômiale :

$$K(x_i - x_j) = (x_i^T x_j + 1)^p \quad (22)$$

Avec p : une constante qui spécifie le degré du polynôme.

Remarque :

Si $p=1$: la formule (22) devient la fonction d'un noyau linéaire.

Dans cette étude nous avons utilisé le noyau Gaussien, linéaire, et polynômial de degré 4.

II.4.3.2 Sélection de modèle SVM :

Une machine à vecteur de support binaire sépare les exemples positifs des exemples négatifs dans la phase d'apprentissage.

Les multiplicateurs de Lagrange α_i pour chaque machine binaire sont déterminés par

la minimisation de la fonction coût donnée par la formule (23) suivante :

$$P(w) = \frac{1}{2} \|w\|^2 \quad (23)$$

Ce problème est résolu à partir de la forme duale qui est exprimée par :

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \quad (24)$$

Cela revient à chercher : $\sum_i \alpha_i y_i = 0$ et $0 \leq \alpha \leq C$

Si la valeur du paramètre de régularisation « C » qui contrôle la tolérance aux erreurs de classification dans la phase d'apprentissage est élevée, donc plus de pénalité sera donnée à l'erreur.

Le vecteur d'apprentissage x_i qui a une valeur de α_i non nul est appelé « *vecteur de support* ».

II.4.3.3 Estimation de l'erreur de généralisation :

La technique la plus populaire pour l'estimation de l'erreur de généralisation est la validation croisée (en anglais cross validation ou Leave One Out (LOO)) qui est utilisée indépendamment de la nature de la machine d'apprentissage utilisée. Le principe de cette méthode consiste à séparer en N sous-ensembles les échantillons de la base d'apprentissage, à apprendre sur N - 1 sous-ensembles, à valider sur le sous-ensemble restant, puis à faire tourner les sous-ensembles de façon à ce que chacun ait pu être testé. Au final, on note le nombre d'erreurs de classification de la procédure LOO par $l(X_1, Y_1, \dots, X_n, Y_n)$.

Il a été démontré que dans [7] que cette procédure donne une estimation presque non biaisée de l'espérance de l'erreur de généralisation.

En effet, l'espérance $E(\cdot)$ de la probabilité p_{err}^{n-1} de l'erreur de test par une machine entraînée à partir de $n-1$ exemples est donnée par la formule (25):

$$E(p_{err}^{n-1}) = \frac{1}{n} E(l(X_1, y_1, \dots, X_n, y_n)) \quad (25)$$

Cependant la procédure est coûteuse en calcul, car nécessitant « n » apprentissages.

Une procédure simple d'estimation d'erreur de généralisation est la validation croisée dite K -fold. Elle consiste à diviser l'ensemble des données en k sous-ensembles mutuellement exclusifs de taille approximativement égale.

L'apprentissage de la machine est effectué en utilisant $k-1$ sous ensemble et le test est effectué sur le sous-ensemble restant. Cette procédure est utilisée une fois pour le test. La moyenne des k taux d'erreur obtenus estime l'erreur de généralisation.

Nous avons montré dans notre travail que la procédure de la validation croisée donne de bonnes estimations de l'erreur de généralisation. Le nombre d'ensembles utilisé dans notre travail est égal à 5 (5fold).

II.5 Détection par SVM:

Notre approche de détection est basée sur l'extraction automatique des fenêtres rectangulaires. Le contenu de ces fenêtres est transformé en vecteurs de caractéristiques en appliquant les méthodes d'extraction citées précédemment. Ces vecteurs sont ensuite injectés dans notre classificateur SVM afin de déterminer si une fenêtre correspond ou pas à une région de vertèbres.

Les fenêtres extraites sont de taille fixe, avec un chevauchement pour parcourir la totalité de l'image, respectant l'algorithme suivant (voir figure 11):

1. Mettre une fenêtre de taille fixe $W_{sub} * H_{sub}$ dans le coin supérieur gauche de l'image.
2. Extraire une région de l'image dans cette fenêtre.
3. Décaler la fenêtre à droite avec un pas de S_x ($S_x < W_{sub}$).

4. Extraire une nouvelle région de l'image dans la fenêtre. Refaire l'étape 3 et 4 jusqu'à ce que la fenêtre soit au bord droit de l'image.
5. Décaler la fenêtre au bord gauche de l'image.
6. Décaler la fenêtre en bas avec un pas de S_y ($S_y < H_{sub}$).
7. Extraire une nouvelle région de l'image dans la fenêtre.
8. Refaire l'étape 3 à 8 jusqu'à ce que la fenêtre soit au bord droit de l'image.

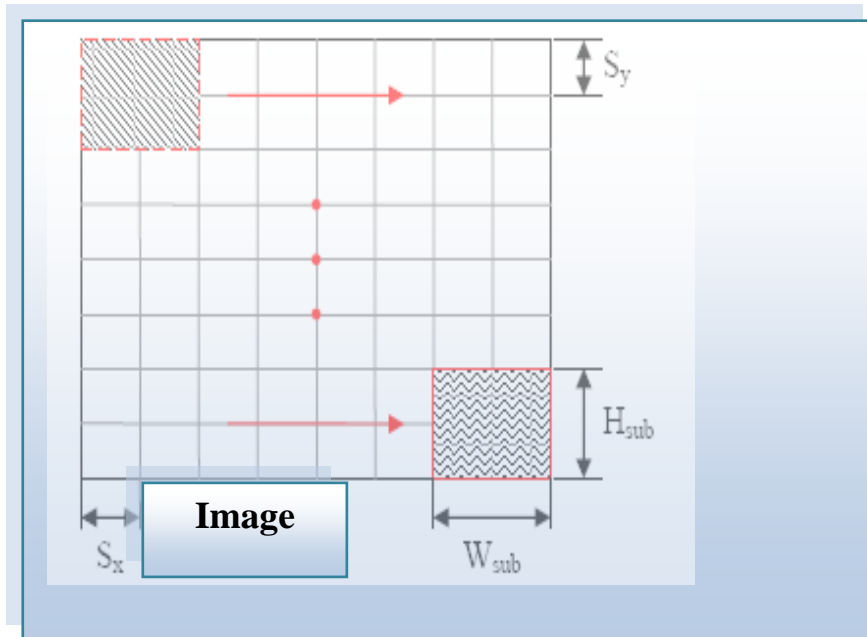


Figure 11 : Processus d'extraction de fenêtres à la phase de détection.

La figure (12) présente ce processus d'extraction de différentes fenêtres et de balayage appliqué à un exemple de notre base :

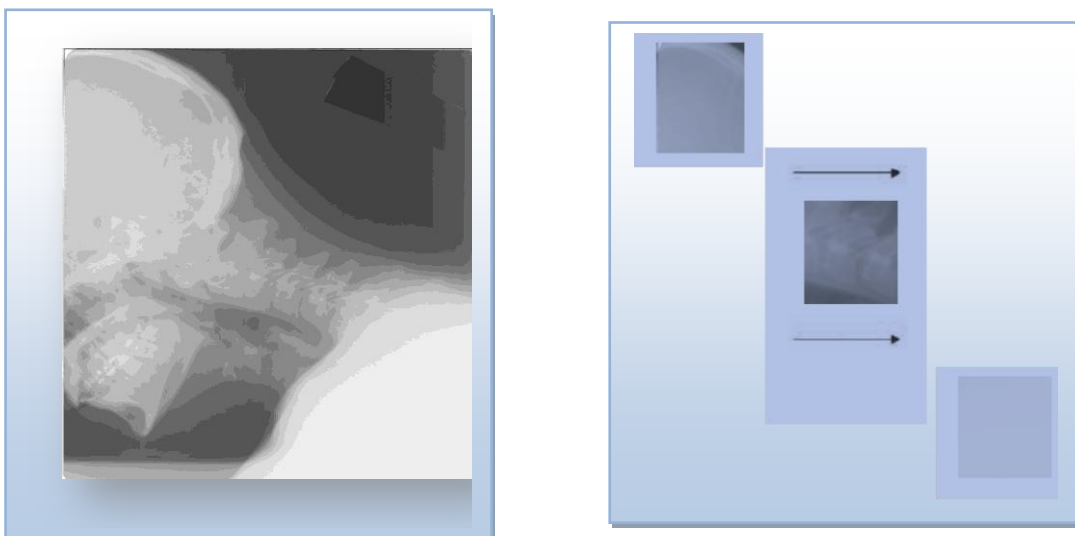


Figure 12 : Exemple de processus d'extraction de fenêtres à la phase de détection.

Post traitement : Afin d'éliminer les fausses réponses du classifieur SVM une méthode d'analyse à base de coefficients de corrélation est utilisée.

II.5.1 Le coefficient de corrélation : est un indice de mesure de l'intensité d'un lien qui peut exister entre deux variables. Le coefficient de corrélation peut prendre une valeur comprise entre -1 et +1. S'il est égal à 0, cela signifie qu'il n'existe aucun lien entre ces 2 variables. Il est très généralement utilisé dans le cadre de l'analyse de variables quantitatives.

Dans notre cas les deux variables représentent la fenêtre de détection qui a été mal classé et un modèle d'une région positive contenant la vertèbre.

Notons A la matrice représentant la fenêtre de balayage et B la matrice représentant notre modèle moyen.

La formule utilisée pour calculer ce coefficient 'r' est :

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2) (\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (26)$$

La valeur du Coefficient de Corrélation est toujours comprise entre -1 et +1

Plus il est proche de 1 ou de -1, plus les points sont proches d'une droite. S'il est égal à ± 1 , les points sont strictement alignés.

II.6 Estimation par RANSAC :

Dans le but du raffinement des résultats de la phase précédente, une méthode d'estimation « robuste » est utilisée afin d'éliminer les fausses détections du classifieur et encore retrouver les vertèbres manquantes.

Un algorithme d'estimation est dit « robuste » s'il garde ses propriétés malgré les incertitudes sur le modèle, les erreurs de mesure et les changements de l'environnement.

La méthode RANSAC est utilisée afin de répondre aux besoins de raffinement.

Nous l'utilisons dans notre travail pour estimer une courbe qui passe par les vertèbres détectées.

II.6.1 L'algorithme RANSAC :

La méthode RANSAC [25] (en anglais Random Sample Consensus) est une méthode de vote probabiliste. Le principe de cette dernière est l'utilisation d'un minimum de données nécessaires pour l'estimation.

Chaque estimation, avec un jeu de données particulier, correspond à un "vote" pour les paramètres obtenus. Le jeu de paramètres élu, i.e. le plus "voté", est retenu comme résultat de l'estimation.

La formulation basique de l'algorithme RANSAC, permet d'estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données observées qui contient des valeurs aberrantes (en anglais « outliers ») illustrés dans la figure 13.

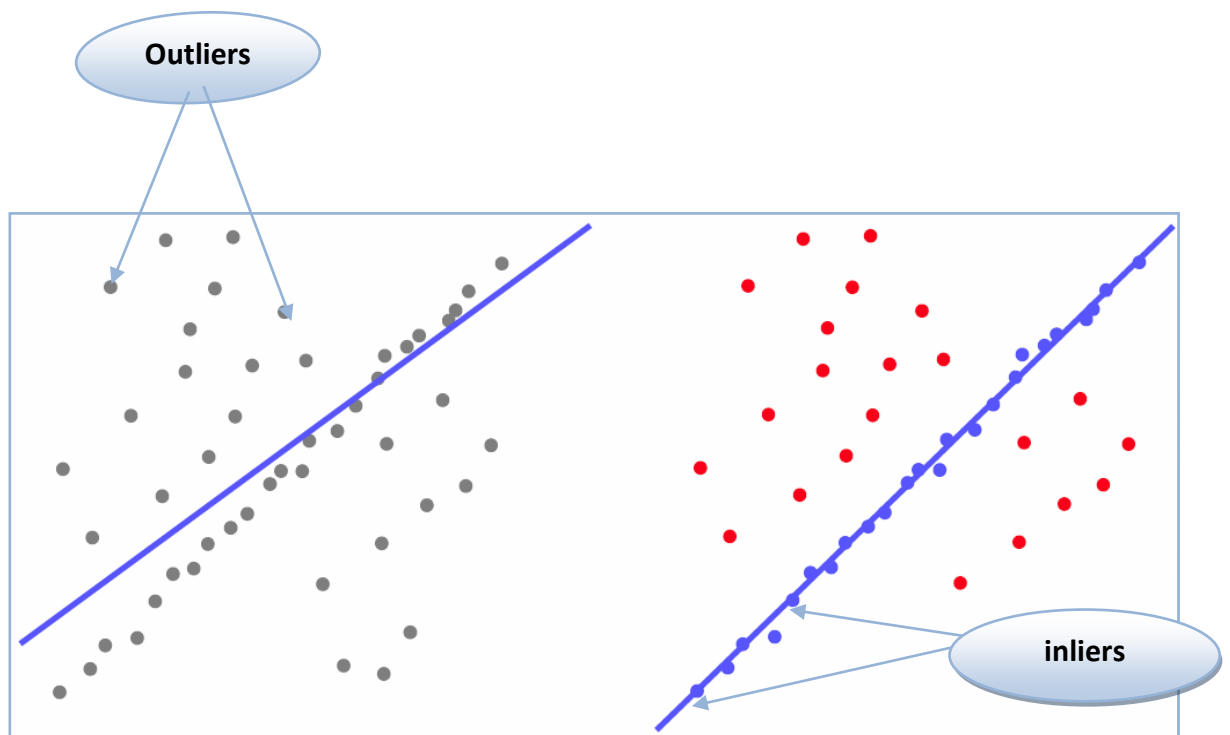


Figure 13 : Exemple d'estimation d'une droite avec RANSAC

Son principe de fonctionnement est le suivant :

L'algorithme effectue d'une façon itérative, un tirage aléatoire d'un échantillon dont la taille est suffisante pour estimer les paramètres d'un modèle mathématique.

Après l'estimation du modèle, le nombre de points « proches » du modèle, est comptabilisé. On parlera dans ce cas des points valables (« en anglais « inliers » »).

Le modèles ayant le plus grand nombre de points est choisi et considéré comme étant le meilleur modèle présent dans le jeu de données. La taille de l'échantillon correspond au nombre de points minimum requis pour initialiser le modèle.

Ensuite, l'ensemble des points valables est déterminé via le calcul de la distance qui sépare chaque point du model estimé. Tout point se trouvant à une distance inférieur à une tolérance donnée du modèle initialisé est considéré comme un point valable.

L'algorithme RANSAC de base peut se résumer par les étapes suivantes :

Données : ensemble S d'éléments.

- **Répéter N fois :**
 1. Sélectionner au hasard un sous ensemble de S (s_k)
 2. Instancier le modèle à partir de s_k .
 3. Déterminer l'ensemble S_k contenu dans S qui sont plus près du modèle instancié qu'une distance t (c'est-à-dire pour chaque donnée de S , calculer sa distance du modèle, et si elle est plus petite que t , on le sélectionne pour S_k).
 4. S_k est un ensemble consensus et définit les données régulières de S
 5. Si $|S_k| > T$, on sort de la boucle
- **Retenir $\max|S_k|$ pour tout k , et ré-estimer le modèle à partir de tous les éléments du S_k retenu.**

II.6.2 Sélection des paramètres :

Après « N » itérations de l'algorithme RANSAC, les paramètres retenus sont ceux qui minimisent le nombre d'erreurs, c'est à dire le nombre de points distants du modèle estimé.

Le nombre d'itérations, N , est choisie suffisamment élevé pour faire en sorte que la probabilité p (normalement fixé à 0.99) fait qu'au moins un des ensembles d'échantillons aléatoires S_k ne comprend pas une valeur aberrante (« outliers »).

Supposons ω , la probabilité qu'une donnée soit régulière. Alors $\varepsilon = 1 - \omega$. On tire un échantillon de s données ($|s_k| = s$).

Ainsi, ω^s est la probabilité d'avoir s données régulières, et $1 - \omega^s$ est la probabilité d'avoir au moins une donnée aberrante.

'N' échantillons sont tirés à partir de s données, alors :

$(1 - \omega^s)^N$: Avoir N fois au moins une données aberrante.

$1 - (1 - \omega^s)^N$: Avoir au moins 1 échantillon avec aucune données aberrante.

Ainsi :

Le nombre N peut être calculé par la formule (27) suivante :

$$N = \frac{\log(1 - p)}{\log(1 - \omega^s)} \quad (27)$$

Le paramètre T est posé :

$T =$ nombre présumé de données régulières dans $S = \omega * |s|$ (règle heuristique).

Dans le chapitre suivant nous allons présenter les résultats de classification des régions de vertèbres par SVM en utilisant la procédure de 5 - fold pour le réglage des différents paramètres de notre modèle, suivie des résultats de test de détection et de raffinements sur différents images par l'algorithme RANSAC.