

**UNE NOUVELLE APPROCHE POUR LA CARACTERISATION ET
LA CLASSIFICATION DE TEXTURES SUR IMAGES NATURELLES**

Gilles BUREL^{(*)(**)}, Jean-Yves CATROS^(*), Hugues HENOCQ^{(*)(**)}

^(*) Thomson CSF-LER, Avenue de Belle Fontaine, F-35510 CESSON-SEVIGNE

^(**) L.E.S.T., URA CNRS 1329, 6 Avenue Le Gorgeu, F-29200 BREST

RESUME :

La plus grande partie des méthodes de classification de textures existantes consiste à alimenter un classifieur par un ensemble de paramètres caractéristiques calculés localement sur l'image texturée. La mise en œuvre de ces méthodes dans le cadre d'applications opérationnelles suppose la prise en compte d'un élément important : le risque de confusion de classes dans l'espace paramétrique. Pour éviter ce problème, nous proposons d'exploiter la conjecture de Gagalowicz [12], qui nous fournit un ensemble de paramètres suffisants pour caractériser totalement la texture. Nous montrons qu'un classifieur connexionniste est capable d'exploiter efficacement ces paramètres.

ABSTRACT :

The existing texture classification methods are generally based on a parameter extraction stage followed by a classifier stage. Using this kind of method for an operational application requires to take into account the risk of classes mixture in the parameters space. We propose to take profit of Gagalowicz conjecture in order to minimise this risk. The conjecture provides us with a set of parameters which totally describe the texture. We show that a connectionnist classifier is able to deal efficiently with these parameters.

Mots clé : Apprentissage, Auto-covariance, Connexionnisme, Histogramme, Infra-Rouge, Rétro-propagation, Texture.

Keywords : Auto-covariance, Back-propagation, Connectionnism, Histogram, Infra-Red, Machine learning, Texture.

1 Introduction

La texture est une caractéristique importante pour la segmentation de divers types d'images, des images aériennes aux images médicales. Le système visuel humain est extrêmement performant dans ce domaine. Ainsi, sur une scène naturelle en extérieur on distingue sans difficulté les différentes textures : l'herbe, le feuillage, le sable, etc. Cependant, la segmentation automatique d'images par analyse de texture est toujours l'un des problèmes les plus difficiles dans le domaine du traitement d'images ; problème qui reste d'ailleurs ouvert car il n'y a pas de définition universellement acceptée de la notion de texture. Intuitivement, la notion de texture est liée à l'aspect homogène d'une surface. Une propriété essentielle de la perception texturale est son invariance par translation : une texture laisse la même impression au système visuel, quelle que soit la partie de la texture qui est observée.

Nous proposons une méthode de segmentation d'image par analyse de texture destinée à être mise en œuvre dans le cadre d'applications opérationnelles [20]. Pour ce type d'application, l'information texturale est utilisée en vue de dresser une carte des fonds (par exemple : ciel, végétation, habitations, etc). Cette carte doit pouvoir être élaborée indépendamment des conditions météorologiques, de l'éclairage (jour/nuit), etc.

Un bref rappel des techniques les plus couramment employées pour la classification de textures nous permettra de dégager un point commun à ces techniques : l'extraction d'un petit nombre de paramètres caractéristiques (quelques unités à quelques dizaines). Ceci est principalement dû au fait que la plupart des classifieurs disponibles nécessitent un bon choix de paramètres (de préférence non redondants, hautement discriminants, et tels qu'une distance euclidienne sur l'espace paramétrique soit significative). Le danger inhérent à ce type d'approche est le risque de confusion de classes dans l'espace paramétrique lorsque la base de données devient importante (ce qui est le cas pour les applications opérationnelles). Pour les applications opérationnelles, il est nécessaire de reconnaître la texture sur une grande gamme de distances, par diverses conditions météorologiques, etc. Ceci augmente la probabilité pour qu'il existe des configurations dans lesquelles deux textures différentes soient caractérisées par des vecteurs paramètres très voisins, voire confondus.

Nous proposons ici une approche différente qui, sous réserve de la validité d'une conjecture que nous exposerons plus loin, n'est pas sujette à ce problème, car les paramètres extraits seraient suffisants pour caractériser totalement la texture (i.e. la connaissance de ces paramètres suffit pour synthétiser une texture tout à fait semblable à l'originale). Ces paramètres sont les échantillons de l'histogramme et de l'autocovariance de la texture, et sont au nombre de quelques centaines. Nous montrerons qu'un perceptron multicouches se révèle suffisamment performant pour apprendre à reconnaître les fonctions d'autocovariance et les histogrammes des différentes textures à classifier. Une comparaison

avec d'autres classifieurs, alimentés par les mêmes paramètres, montrera l'avantage du perceptron multicouches. Enfin, nous validerons notre approche sur des images Infra-Rouge et des images du domaine visible.

La segmentation d'images est réalisée, comme dans la plupart des méthodes de segmentation par texture, par le biais d'une fenêtre d'analyse que l'on déplace sur l'image. Ceci dégrade la précision sur les frontières entre régions. Toutefois, dans le cadre d'applications opérationnelles, il ne s'agit là que d'un inconvénient mineur, du fait de la grande taille des images et des zones traitées (typiquement 36864x1024 pixels pour une image correspondant à 360° en gisement et 10° en site).

Les expérimentations dans le domaine Infra-Rouge ont été réalisées dans le cadre du programme Thomson CSF "Veille Panoramique Infra-Rouge et Réseaux de Neurones" [20] [35].

2 Méthodes classiques

Nous nous situons ici dans le cadre de la classification supervisée de textures, c'est-à-dire dans le cas où l'on a prédéfini les classes (par exemple : ciel, terre, habitations, ...), et où l'on dispose a priori d'échantillons représentatifs de ces textures. Il existe également des méthodes de classification non-supervisée, qui segmentent une image en zones, le regroupement des pixels de l'image dans une même zone se faisant sur un critère de proximité dans l'espace paramétrique. Le résultat obtenu (segmentation de l'image) dépend alors uniquement du choix des paramètres. Les méthodes non-supervisées peuvent être utiles pour découper une image en zones, et servir de prétraitement à des étages supérieurs (reconnaissance d'objets, ...). Mais les zones ne correspondent pas obligatoirement à des entités physiques (ciel, feuillage, ...). Des méthodes non-supervisées performantes exploitant des modèles des frontières ou des régions ont été publiées récemment ([13] par exemple).

Un grand nombre de méthodes a été proposé pour la discrimination de textures. Toutefois, il est possible de les regrouper en deux grandes familles :

1. Les méthodes statistiques
2. Les méthodes structurelles

De bonnes revues sont fournies par [14] [15] [16] et [38]. Les applications sont très variées, et couvrent divers types d'images, des images médicales aux images aériennes [18] [26]. Comme Kashyap [24] l'a très justement noté, le point fondamental dans toutes ces méthodes est le choix d'un ensemble de caractéristiques qui permettent de

réduire la dimension des données à une quantité acceptable pour le classifieur, tout en tentant de préserver une grande partie de l'information discriminante. De plus, pour la plupart des classifieurs, il est souhaitable que la loi de probabilité de ces caractéristiques conditionnellement à la classe soit simple (par exemple de type gaussien).

Les méthodes statistiques considèrent la texture comme un champ aléatoire à deux dimensions, et les caractéristiques requises sont obtenues par des approches telles que les matrices de co-occurrence [9] [14] [17] [31] [36], la fréquence spatiale [1] [5] [6] [19] [21] [28], la corrélation [4], ou les modèles paramétriques [8] [10] [11] [24] [39]. Quelques études comparatives non-exhaustives ont été proposées [7] [29]. Elles montrent que les méthodes basées sur les matrices de co-occurrence sont généralement les plus performantes, bien que des méthodes moins gourmandes en puissance de calcul peuvent donner des résultats comparables sur des ensembles réduits de textures [29].

Les méthodes structurelles caractérisent la texture par des primitives élémentaires appelés "texels", et par l'arrangement spatial de ces primitives [25] [33] [34] [37]. Bien que saisissant certains aspects du fonctionnement du système visuel humain, les méthodes structurelles sont généralement plus complexes que les méthodes statistiques, et réagissent assez mal en présence de textures faiblement structurées telles que l'herbe, le feuillage, la laine, et un grand nombre de textures naturelles.

On constate donc qu'aux deux grandes familles précédemment citées correspondent deux définitions de la texture : une définition statistique et une définition structurelle. Une définition structurelle semble mal adaptée au cas des images Infra-Rouge, qui sont souvent floues, bruitées, avec des contours mal définis. Ceci est moins vrai pour les images naturelles dans le domaine visible. Toutefois, un grand nombre de textures naturelles peuvent être vues comme des champs aléatoires, c'est pourquoi nous opterons ici pour une définition statistique.

Signalons enfin des travaux intéressants visant à segmenter des images selon la texture en utilisant des réseaux neuromimétiques qui imitent la structure du système visuel humain (voir [27] par exemple). Ces travaux à long terme sont pour l'instant validés sur des configurations simples. D'autres travaux dans le domaine des réseaux de neurones consistent à alimenter un classifieur neuronal avec des paramètres classiques (dans [32] par exemple, ces paramètres sont 6 moments invariants et 6 coefficients de prédiction linéaire).

3 Méthode proposée

3.1 Idées générales

En réalisant des expériences psychovisuelles, suite aux travaux de Julesz [22] [23], Gagalowicz [12] a obtenu un résultat particulièrement intéressant concernant la discrimination visuelle de textures aléatoires : “Deux textures naturelles possédant les mêmes histogrammes locaux et les mêmes fonctions d'autocovariance ne sont pas visuellement discriminables”. Sur la base de cette conjecture on peut définir la texture comme une image, ou une portion d'image, telle que son histogramme et son autocovariance soient invariants par translation.

Le qualificatif “naturelles” est important car un exemple de deux textures de synthèse visuellement discriminables bien que possédant même autocovariance et histogramme a été trouvé par Gagalowicz lui-même. Il s'agit de textures synthétiques d'autocovariance nulle en tout point (sauf à l'origine), alors qu'une texture naturelle est plutôt spatialement corrélée. Les expérimentations menées par Gagalowicz avec un grand nombre de textures naturelles semblent confirmer sa conjecture.

Ce résultat limite l'ensemble des paramètres à prendre en compte par un système qui vise à réaliser la discrimination de textures “comme l'œil le fait”. Cependant, la quantité de données à traiter est toujours énorme. Une première solution est d'extraire un faible nombre de paramètres de l'histogramme et de l'autocovariance (par exemple l'entropie de l'histogramme, etc). Mais le problème résultant est le risque de perte d'information significative.

La solution que nous proposons est d'utiliser un classifieur suffisamment puissant pour traiter directement l'histogramme et l'autocovariance. Un réseau de neurones multicouches (perceptron multicouches) semble être capable de réaliser une telle tâche. Cette approche diffère de la plupart des méthodes traditionnelles de par le fait qu'il n'y a pas de réduction draconienne des données (par exemple, les méthodes à base de matrices de co-occurrence n'utilisent jamais directement ces matrices, mais simplement un petit nombre de paramètres ad-hoc extraits de ces matrices).

Durant une phase d'apprentissage préalable, ce système aura appris, à partir d'exemples, à caractériser diverses textures. Le perceptron multicouches recevra en entrée l'autocovariance et l'histogramme de la texture (calculés sur une fenêtre d'observation), et fournira en sortie une indication de classe et de confiance (la classe et la confiance seront affectés au pixel central de la fenêtre d'observation). On met ainsi à la disposition du classifieur toute l'information pertinente pour la discrimination des fonds, ce qui n'est pas le cas avec les systèmes classiques qui n'exploitent généralement qu'un nombre restreint

de paramètres.

En contrepartie, l'apprentissage sera certainement plus long du fait de la complexité de la métrique pertinente sur l'espace paramétrique. En effet, la majorité des méthodes classiques exploitent des paramètres de texture qui sont tels que chaque paramètre (variance, énergie dans une bande de fréquence du spectre, etc) apporte une information significative indépendamment des autres, et respecte une loi de probabilité simple conditionnellement à la classe. Une distance euclidienne dans l'espace paramétrique est alors significative.

Ceci n'est pas le cas dans le traitement que nous proposons car, par exemple, un point de la fonction d'autocovariance apporte une information quasi-nulle s'il est pris indépendamment des autres. Mais par contre, l'information globale apportée par l'autocovariance et l'histogramme est très élevée car il est souvent possible de synthétiser une texture très semblable à l'originale à partir de cette donnée. Le paragraphe suivant a pour objectif de vérifier ce point.

3.2 Vérification de la pertinence des paramètres

Afin de vérifier la pertinence des paramètres choisis, nous combinons un système d'identification et un système de synthèse (fig 1). L'identification réside dans l'extraction de l'autocovariance et de l'histogramme d'un échantillon de texture naturelle. Puis on crée de la texture synthétique de telle sorte qu'elle ait même autocovariance et histogramme que la texture naturelle, en s'inspirant d'une méthode de synthèse décrite par Gagalowicz [12]. Si la texture synthétisée a même apparence que la texture d'origine, cela signifie que l'ensemble {autocovariance + histogramme} est suffisant pour caractériser une texture.

Notons \vec{H} l'histogramme et \vec{C} l'autocovariance de l'échantillon de texture naturelle. La méthode de synthèse est une méthode itérative, qui consiste à modifier progressivement l'échantillon de la nouvelle texture de sorte que ses paramètres se rapprochent de plus en plus de ceux de la texture naturelle. Notons \hat{H} et \hat{C} les paramètres de la nouvelle texture en cours de synthèse. On se définit une erreur qui est :

$$E = \| \hat{C} - \vec{C} \|^2 + \alpha \| \hat{H} - \vec{H} \|^2$$

Le principe de synthèse consiste d'abord à générer une image aléatoire dont l'histogramme est identique à celui de la texture naturelle (ceci peut être aisément réalisé en donnant aux

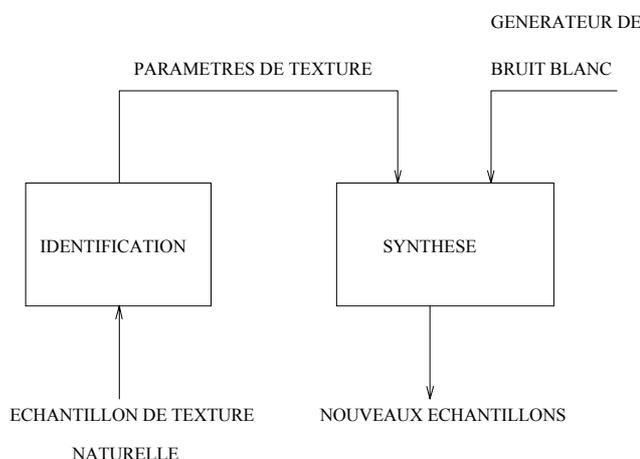


FIG. 1: Synthèse de textures

pixels des luminances aléatoires suivant une distribution de probabilité correspondant à l'histogramme). Puis on réitère un grand nombre de fois l'opération qui consiste à tirer un pixel au hasard et à lui donner la luminance qui améliore le mieux l'erreur. Il serait bien entendu coûteux de recalculer l'autocovariance à chaque itération. Mais si l'on prend α assez grand, l'histogramme \hat{H} est contraint à rester très proche de \vec{H} , ce qui autorise des approximations conduisant à une expression récurrente de l'autocovariance en fonction de sa valeur à l'itération précédente.

Reste à savoir sur quelle extension on doit calculer l'autocovariance. Selon des expériences psycho-visuelles réalisées par Gagalowicz[12], l'œil humain est peu sensible aux corrélations entre des points vus sous un angle solide supérieur à 9° . Or l'observateur humain est capable de classifier sans trop de problèmes les fonds des images de notre base de données lorsque celles-ci sont présentées sur une station de travail (bien que la classification soit parfois assez difficile lorsque l'observateur n'est autorisé à voir qu'une zone limitée de l'image). Sur ce type de station, le pixel mesure environ 0.35mm de côté, et l'observateur est à une distance de l'écran de l'ordre de 50cm. Dans ces conditions, un calcul simple montre que 9° d'angle correspondent à 4 pixels. Compte tenu du fait que la mesure donnée par Gagalowicz est très approximative, et que la distance écran-observateur peut varier énormément, on s'accorde une bonne marge en prenant une extension de 16 pixels. Comme l'autocovariance est symétrique par rapport à l'origine, elle est calculée sur un demi plan seulement.

La synthèse d'images de Brodatz [2] a été réalisée en utilisant la méthode précédente (fig 5). Chaque texture fait 128x128 pixels, et l'identification a été réalisée, pour chaque

texture, sur une fenêtre de 64x64 pixels. Les résultats obtenus montrent que les textures synthétisées laissent la même impression d'ensemble que les textures naturelles (sauf pour les textures très structurées). La ressemblance décroît avec le niveau de structuration de la texture.

Le même processus a été appliqué aux textures Infra-Rouge "ciel", "terre", et "habitations" définies sur le site N. Les résultats de synthèse sont fournis sur la figure 6. Ces images sont de taille 64x64, et l'identification a été réalisée sur des échantillons 32x32 extraits d'une image naturelle. Les textures synthétiques "ciel" et "terre" sont tout à fait semblables aux textures naturelles. Cette ressemblance est moins vraie pour la texture "habitations", qui est plus structurée. On retrouve toutefois les principaux aspects de zones d'habitation : alignements, alternance de zones sombres (toits) et claires (murs), ...

En conclusion il n'y a donc pas de perte d'information significative pour la discrimination de texture dans le prétraitement que nous réalisons (du moins tant que les textures ne sont pas trop structurées). Remarquons que ceci n'est pas le cas dans les méthodes habituelles, où le nombre de paramètres extraits est limité et serait tout à fait insuffisant pour resynthétiser des textures de même apparence.

3.3 Prétraitement

La notion de texture étant par définition relative à une impression d'ensemble, elle n'a de sens que sur une certaine étendue (on ne peut pas parler de texture au niveau du pixel). C'est pourquoi la classification doit se faire par l'intermédiaire d'une fenêtre d'observation, que l'on déplacera sur l'image. La fenêtre d'observation est une fenêtre carrée, dont la taille est égale à $N \times N$ pixels. Les paramètres sont calculés sur cette fenêtre et la classe trouvée est affectée au pixel central. Notons $L(x,y)$ la luminance du pixel de coordonnées (x,y) . On définit ci-dessous les paramètres calculés.

Autocovariance :

$$C(\vec{\Delta}) = \frac{\sum_{y=y_a}^{y_b} \sum_{x=x_a}^{x_b} L^*(x, y)L^*(x + \delta_x, y + \delta_y)}{(y_b - y_a)(x_b - x_a)}$$

où

$$L^*(x, y) = L(x, y) - \eta$$

$$\eta = \frac{\sum_{y=0}^{N-1} \sum_{x=0}^{N-1} L(x, y)}{N^2} \quad (\text{moyenne})$$

Les bornes x_a, x_b, y_a, y_b doivent être déterminées pour chaque déplacement $\vec{\Delta}(\delta_x, \delta_y)$ afin que celui-ci ne sorte pas de la fenêtre :

$$\begin{aligned} y_a &= \text{sup}(0, -\delta_y) \\ y_b &= \text{inf}(N-1, (N-1) - \delta_y) \\ x_a &= \text{sup}(0, -\delta_x) \\ x_b &= \text{inf}(N-1, (N-1) - \delta_x) \end{aligned}$$

Le calcul se fait jusqu'à une extension de 15 pixels dans les 2 dimensions. En exploitant la symétrie de l'autocovariance par rapport à l'origine, $C(\vec{\Delta})$ sera calculé pour tous les $\vec{\Delta}$ tels que :

$$\begin{aligned} -15 \leq \delta_y \leq 15 & & 1 \leq \delta_x \leq 15 \\ 1 \leq \delta_y \leq 15 & & \delta_x = 0 \end{aligned}$$

Une normalisation en amplitude est ensuite réalisée en divisant les $C(\vec{\Delta})$ par $C(\vec{0})$. On se ramène donc à des valeurs entre -1 et +1.

Une résolution égale au pixel pour les points éloignés de l'origine n'étant pas justifiée, on réalise un sous échantillonnage comme indiqué sur la figure 2. En effet, ces valeurs éloignées de l'origine sont très sensibles à de légères dilatations de la texture. En sous-échantillonnant on s'assure une meilleure robustesse. Les six cases non grisées ne sont pas retenues du fait de la symétrie de l'autocovariance par rapport à l'origine, et de la normalisation en amplitude ($C(\vec{0})$ vaut toujours 1 après normalisation). Enfin, les points de l'axe horizontal éloignés de l'origine sont partagés entre deux échantillons.

Histogramme :

$$H(L) = \frac{\text{nombre de pixels de luminance } L}{N^2}$$

L'histogramme est ensuite centré sur sa valeur moyenne afin de ne pas être sensible à la luminance moyenne (par exemple, en Infra-Rouge, le ciel présente souvent un dégradé de luminance moyenne). Puis, il est sous-échantillonné avec un pas grandissant avec l'éloignement par rapport à la zone centrale (progression logarithmique), ce qui réduit l'influence du bruit, tout en conservant une bonne résolution dans la zone proche de la moyenne, qui est a priori la plus intéressante (fig 3).

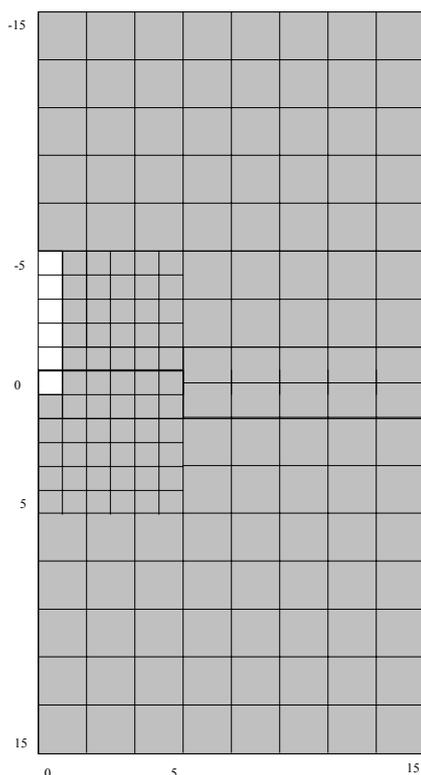


FIG. 2: sous échantillonnage de l'autocovariance

3.4 Le perceptron multi-couches

Nous utilisons un perceptron multi-couches (également nommé réseau de neurones multicouches) entraîné par l'algorithme de rétropropagation [30], auquel nous avons apporté des modifications qui permettent la stabilisation de l'apprentissage et le réglage automatique des paramètres de l'algorithme [3]. Le réseau de neurones est représenté figure 4. Le modèle du neurone est un sommateur pondéré suivi d'une non-linéarité en tangente hyperbolique, sauf pour les neurones de la couche d'entrée qui se contentent de recopier leur entrée sur leur sortie. Le réseau est entièrement connecté entre 2 couches successives, et est seuillé. Pour nos expérimentations en Infra-Rouge, le nombre total d'entrées est 210 (40 pour l'histogramme et 170 pour l'autocovariance, avec les sous-échantillonnages proposés au paragraphe précédent).

Le système fonctionne en 2 étapes :

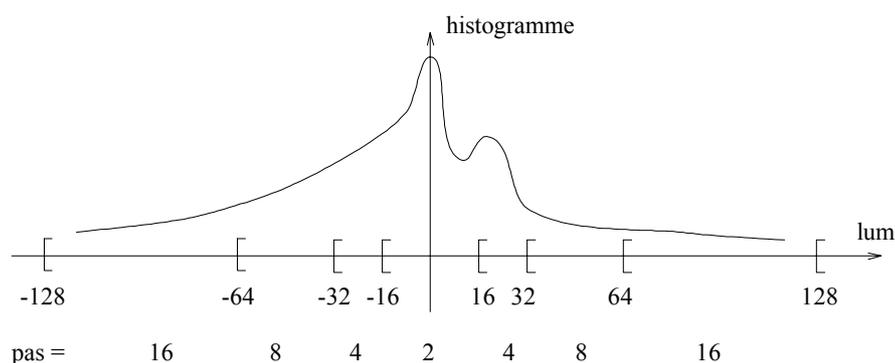


FIG. 3: sous échantillonnage de l'histogramme

Première étape : La phase d'apprentissage

Durant cette phase, le réseau apprend à classifier correctement des exemples de chaque texture. L'algorithme d'apprentissage est l'algorithme de rétropropagation, qui permet d'ajuster les coefficients de pondération des neurones pour obtenir le comportement désiré. Les exemples sont les paramètres (autocovariance + histogramme) calculés sur des fenêtres d'observation extraites des textures.

Seconde étape : La phase de reconnaissance

Le réseau peut maintenant classifier des images entières selon les textures qui y sont présentes. Ceci est réalisé par l'intermédiaire d'une fenêtre d'observation qui se déplace sur l'image. Pour chaque position, le réseau est alimenté par l'autocovariance et l'histogramme de la fenêtre. Les sorties du réseau sont calculées, et leur interprétation fournit une classe et une mesure de confiance. Lorsque toute l'image a été classifiée, un second passage permet de reclassifier les points dont la confiance est faible en effectuant un vote majoritaire pondéré dans leur voisinage.

Le réseau de neurones n'est pas directement alimenté par le contenu de la fenêtre d'observation car cela le forcerait à apprendre l'invariance en translation. Il est évident qu'une telle tâche est quasiment insurmontable à moins d'apprendre sur un nombre gigantesque d'exemples. L'histogramme et l'autocovariance présentent un double avantage : ils sont invariants en translation, et de plus ils caractérisent totalement la texture. Notons que l'autocovariance est par définition insensible à la valeur moyenne. De plus l'histogramme a également été normalisé par rapport à sa valeur moyenne. Ceci est souhaitable car la moyenne ne doit normalement pas être utilisée pour classifier la texture (si l'on change l'éclairage, la texture reste la même).

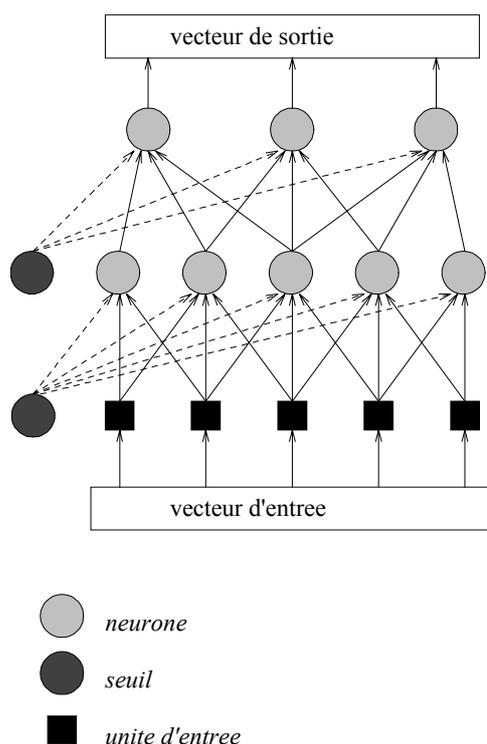


FIG. 4: Perceptron multicouches

En sortie du réseau de neurones, on a un neurone par classe, et on utilise le codage suivant pour l'apprentissage :

- classe 0 : $(1, -1, -1, \dots, -1, -1)$
- classe 1 : $(-1, 1, -1, \dots, -1, -1)$
- ... : ...
- classe n : $(-1, -1, -1, \dots, -1, 1)$

Il n'existe pas de méthode donnant a priori le choix optimal du nombre de couches et du nombre de neurones sur les couches cachées. La bonne démarche consiste à adopter une architecture aussi réduite que possible afin d'éviter une mauvaise généralisation. En effet, un réseau sur-dimensionné contient un trop grand nombre de paramètres libres (les poids du réseau), d'où un phénomène d'apprentissage "par cœur". Signalons toutefois que les performances en généralisation ne sont pas extrêmement sensibles au nombre de neurones cachés. C'est surtout l'ordre de grandeur qui est important. En infra-rouge, par exemple, nous avons obtenu de bons résultats avec un réseau à trois couches et six neurones cachés. Au delà de ces chiffres les performances en généralisation commencent à se dégrader lentement.

3.5 Calcul de la confiance et post-traitement

Pendant la phase de reconnaissance, le neurone dont la valeur de sortie est la plus forte détermine la classe. Notons j_1 ce neurone et O_{j_1} sa sortie. Une mesure de confiance est associée à la classification. La confiance est calculée comme :

$$CONFIANCE = \min_{j \in \text{sortie}, j \neq j_1} \left(\frac{O_{j_1} - O_j}{2} \right)$$

Par exemple, pour quatre classes de textures, et les valeurs de sortie $(-1, +0.95, -0.35, -0.98)$, nous aurons :

$$CLASSE = 1 \quad (\text{numéro du neurone qui a la plus forte sortie})$$

$$\begin{aligned} CONFIANCE &= \frac{1}{2} \min(0.95 + 1.0, 0.95 + 0.35, 0.95 + 0.98) \\ &= 0.65 \end{aligned}$$

La confiance est toujours une valeur entre 0 et 1, la valeur 1 correspondant à une forte confiance. Le premier passage de la phase de reconnaissance produit une image de classification et une image de confiance. Puis un deuxième passage reclassifie les points dont la confiance est faible en effectuant un vote majoritaire (pondéré par la confiance et par une fonction gaussienne de la distance) dans un voisinage du point central. La nouvelle classe est obtenue en analysant les pixels voisins distants de d_{max} au plus (étendue de reclassification, mesurée en pixels) afin de déterminer la classe la plus crédible.

Exemple de traitement pour 4 classes :

Soit un pixel(x,y) classifié 1 avec une confiance faible (inférieure à un seuil de 0.7 par exemple). On utilise un tableau d'accumulateurs C, la sommation étant réalisée sur tous les pixels voisins à une distance $d \leq d_{max}$ du pixel (x,y).

$$C(\text{classe } 0) = \sum_{\text{pixel} \in \text{classe } 0} k(d) \text{ confiance}(\text{pixel})$$

$$C(\text{classe } 1) = \text{confiance}(x,y) + \sum_{\text{pixel} \in \text{classe } 1} k(d) \text{ confiance}(\text{pixel})$$

$$C(\text{classe } 2) = \sum_{\text{pixel} \in \text{classe } 2} k(d) \text{ confiance}(\text{pixel})$$

$$C(\text{classe } 3) = \sum_{\text{pixel} \in \text{classe } 3} k(d) \text{ confiance}(\text{pixel})$$

$k(d)$ pondère l'influence de la confiance en fonction de la distance séparant le pixel à reclassifier et son voisin. Cette pondération suivra par exemple une loi gaussienne centrée ($\sigma = 7.5$ pixels pour nos expérimentations) :

$$k(d) = e^{-(d^2/2\sigma^2)}$$

A l'issue, de ce traitement on recherche le maximum du tableau C. Soit C(classe i) ce maximum. On a alors :

nouvelle classe = classe i

4 Résultats statistiques

4.1 Conditions expérimentales

Les résultats statistiques sont fournis pour des images Infra-Rouge provenant de sites que nous désignerons par N (site du Nord de la France) et S (site du Sud de la France). Sur le site N, 3 classes ont été définies : ciel (0), terre (1), et habitations (2). Sur le site S, 2 classes ont été définies : ciel (0), et terre (1).

La fenêtre d'observation utilisée a une taille de 31x31 pixels et l'autocovariance dans cette fenêtre est calculée avec une extension de 15 pixels. L'autocovariance est sous-échantillonnée d'un facteur linéaire 2 pour les zones éloignées du centre. L'histogramme est également sous-échantillonné.

La base d'apprentissage contient 320 exemples par classe, soit un total de 960 exemples pour le site N et 640 exemples pour le site S. La base d'évaluation (qui sert à mesurer le taux de généralisation) contient 160 exemples par classe, soit un total de 480 exemples pour N et 320 exemples pour S. Les exemples sont des imasettes 31x31 extraites de plusieurs images panoramiques des sites correspondants (ces images ont été prises par différentes conditions météorologiques et différentes heures du jour et de la nuit). Les imasettes des bases d'évaluation et d'apprentissage ont été extraites sur des zones différentes du panorama.

4.2 Résultats statistiques et comparaisons

Nous donnons ci-dessous les taux de reconnaissance obtenus (sans post-traitement) sur les bases d'apprentissage et d'évaluation. A titre de comparaison, les résultats obtenus avec d'autres classifieurs (k-Plus Proches Voisins) alimentés par les mêmes paramètres

sont également fournis. Les méthodes sont classées par ordre de taux de généralisation décroissant. Les matrices de confusion correspondant au meilleur classifieur sont indiquées.

SITE DU NORD DE LA FRANCE

Méthode	Apprentissage	Généralisation	Nb de multiplications
Réseau 3 couches	96.6%	86.5%	1278
Réseau 2 couches	92.0%	84.0%	630
Plus_Proche_Voisin	100.0%	72.9%	201600
5-Plus_Proches_Voisins	97.0%	68.5%	201600
3-Plus_Proches_Voisins	97.5%	68.3%	201600

Le réseau à 2 couches contient 210+3 neurones, et le réseau à 3 couches 210+6+3 neurones (neurones seuils non compris).

Matrice de confusion pour le réseau à 3 couches (210+6+3 neurones)

Classe	Effectif	Matrice de confusion		
0	320 (33.3%)	86.9	9.4	3.7
1	320 (33.3%)	13.8	82.5	3.7
2	320 (33.3%)	0.6	9.4	90.0
Taux moyen de généralisation : 86.5%				

SITE DU SUD DE LA FRANCE

Méthode	Apprentissage	Généralisation	Nb de multiplications
Réseau à 3 couches	97.5%	93.8%	848
Réseau à 2 couches	97.5%	93.4%	420
5-Plus-Proches-Voisins	94.7%	88.1%	134400
3-Plus-Proches-Voisins	97.3%	87.5%	134400
Plus-Proche-Voisin	100.0%	85.6%	134400

Le réseau à 2 couches contient 210+2 neurones, et le réseau à 3 couches 210+4+2 neurones (neurones seuils non compris).

Matrice de confusion pour le réseau à 3 couches (210+4+2)

Classe	Effectif	Matrice de confusion	
0	160 (50.0%)	90.6	9.4
1	160 (50.0%)	3.1	96.9
Taux moyen de généralisation : 93.8%			

5 Traitement d'images complètes

Afin d'illustrer les résultats obtenus sur le traitement d'images complètes nous présentons la classification d'une image Infra-Rouge (extraite de la base d'images du site N). Le réseau utilisé est un réseau à 3 couches (210+6+3 neurones). Notre méthode est également valide dans le domaine "visible", comme le prouvent les résultats obtenus sur les textures de Brodatz [2] et l'image aérienne que nous présentons ci-dessous.

– **Textures de Brodatz :**

La figure 7 illustre le résultat obtenu pour le traitement d'images de Brodatz, qui servent souvent de référence dans le domaine de la discrimination de textures. La fenêtre d'observation pour l'analyse est de 32x32 pixels. L'apprentissage a été réalisé sur les textures de synthèse (dont on rappelle qu'elles ont été créées à partir des

paramètres extraits sur des fenêtres 64x64 de chaque texture). On a pris 100 exemples par texture.

Sur l'image on a représenté la source (de taille 256x256, en haut à gauche), la classification intermédiaire (en haut à droite), la confiance (en bas à gauche, la luminance étant proportionnelle à la confiance), et la classification définitive (en bas à droite).

– **Image aérienne :**

La figure 8 illustre le résultat obtenu pour le traitement d'une image aérienne (de taille 256x256). Les fenêtres incrustées dans l'image source indiquent les exemples d'apprentissage. La taille de ces fenêtres est de 25x25 pixels pour obtenir une meilleure précision sur les frontières. Des exemples supplémentaires ont été obtenus en déplaçant ces fenêtres de 1 à 4 pixels dans leur voisinage. Le nombre total d'exemples par texture est donc de 81. L'autocorrélation n'a ici été calculée que sur une extension de 7 pixels.

– **Image Infra-Rouge :**

La figure 9 illustre la classification d'images Infra-Rouge. L'apprentissage n'a pas été réalisé sur cette image, mais sur d'autres zones du panorama pour diverses conditions météorologiques. Sur la figure 9, on distingue à gauche l'image source (de taille 512x512, et détramée pour éviter le flou dû au mouvement du capteur), et à droite la classification définitive.

Cette classification est globalement satisfaisante, hormis sur les frontières entre zones (une approche multi-résolution avec plusieurs tailles de fenêtres permettrait d'améliorer la précision aux frontières). Il y a peu d'erreurs de classification sur le ciel, malgré le fort dégradé de luminance moyenne (dû à la variation de la température avec l'altitude) et la présence de nuages sur l'horizon. Certaines erreurs de classification sur les bords de l'image sont dues à des problèmes de numérisation de l'image, et non pas au système de classification.

Il est intéressant de remarquer que les non-stationarités de l'image (dus à la variation de distance) n'induisent pas d'erreur de classification notable. La texture "terre" par exemple apparaît sur une grande gamme de distance dans la figure 9. Le problème de dégradation des performances en présence de non-stationarités est caractéristique des méthodes basées sur une modélisation explicite de chaque classe de texture (voir les expérimentations présentées dans [11] par exemple). Toutefois, il est clair que nous serions confrontés au même problème si l'apprentissage était réalisé sur un base d'exemples insuffisamment représentative.

6 Conclusion

Nous avons montré la faisabilité d'une approche basée sur la conjecture de Gagalowicz pour la classification de fonds. L'étude comparative avec des classifieurs selon les k-Plus-Proches-Voisins montre un net avantage pour le classifieur connexionniste (86.5% contre 72.9% avec le plus proche voisin sur le site du Nord de la France). Nos expérimentations en Infra-Rouge ont montré une bonne robustesse face aux non-stationarités, sous réserve d'une bonne représentativité de la base d'apprentissage.

Il resterait à comparer le prétraitement proposé avec d'autres prétraitements. Il est possible que, sur une base de données réduite, d'autres attributs soient aussi performants que l'histogramme et l'autocovariance. Mais l'avantage de la méthode présentée est de conserver toute l'information pertinente, ainsi que nous l'avons montré par synthèse. Il n'y a donc pas de risques de confusion de classes dans l'espace des paramètres lorsque la base de données devient importante. Cet élément ne doit pas être négligé car un système opérationnel devra travailler sur un volume important de données, et être insensible aux conditions météorologiques, à l'éclairage, etc.

La mise en œuvre du système est très simple car il fonctionne par apprentissage. De plus, les opérations réalisées, tant au niveau du prétraitement que du réseau de neurones sont simples, ce qui conduit à penser qu'une réalisation matérielle ne devrait pas poser de problème majeur. Il serait toutefois souhaitable d'étudier la possibilité de réduire la charge de calcul en exploitant diverses redondances, notamment au niveau du calcul de l'autocovariance.

De nombreuses améliorations restent possibles. Par exemple, l'utilisation de poids partagés et de connexions locales dans le réseau devrait améliorer les performances en généralisation. En effet, la structure bi-dimensionnelle de l'autocovariance n'est pas du tout traduite dans les connexions du réseau utilisé actuellement. D'autre part, une approche multi-résolution, avec plusieurs tailles de fenêtre permettrait d'améliorer la précision sur les frontières.

Références

- [1] A.C. BOVIK, M. CLARK, W.S. GEISLER
“Multichannel texture analysis using localized spatial filters”
IEEE PAMI, vol 12, n° 1, January 1990
- [2] P. BRODATZ
“Textures - A Photographic Album for Artists and Designers”
Dover Publications Inc. , New York, 1966
- [3] Gilles BUREL, Dominique CAREL, Jean Yves CATROS
“A connectionist system for recognition of 2D workpieces”
Revue Technique THOMSON-CSF, vol.22, n°4, Décembre 1990
- [4] P.C. CHEN, T. PAVLIDIS
“Segmentation by texture using correlation”
IEEE PAMI, vol 5, n° 1, January 1983
- [5] M. CLARK, A.C. BOVIK, W.S. GEISLER
“Texture segmentation using Gabor modulation/demodulation”
Pattern recognition Letters 6 (1987) 261-267, September 1987
- [6] J.M. COGGINS, A.K. JAIN
“A spatial filtering approach to texture analysis”
Pattern Recognition Letters 3 (1985) 195-203, May 1985
- [7] R.W. CONNERS, C.A. HARLOW
“A theoretical comparison of texture algorithms”
IEEE PAMI, vol. PAMI-2, n° 3, May 1980
- [8] G.R. CROSS, A.K. JAIN
“Markov Random Field Texture Models”
IEEE PAMI , vol 5, n° 1, January 1983
- [9] L.S. DAVIS, S.A. JOHNS, J.K. ARGAWAL
“Texture analysis using generalized co-occurrence matrices”
IEEE PAMI, vol 1, n° 3, July 1979
- [10] H. DERIN, H. ELLIOTT
“Modeling and segmentation of noisy and textured images using Gibbs Random Fields”
IEEE PAMI, vol 9, n° 1, January 1987
- [11] Z. FAN, F.S. COHEN
“Textured image segmentation as a Multiple Hypothesis Test”
IEEE Trans. Circuits and Systems, vol 35, n°6, June 1988
- [12] A. GAGALOWICZ
“Vers un modèle de textures”
Thèse de doctorat d'état ès sciences mathématiques

Université de PARIS VI , 1983

- [13] D. GEMAN, S. GEMAN, C. GRAFFIGNE, D. PONG
“Boundary Detection by Constrained Optimization”
IEEE PAMI, vol 12, n^o7, July 1990
- [14] R.M. HARALICK, K. SHANMUGAM, I. DINSTEIN
“Textural features for image classification”
IEEE Trans. Syst.,Man,Cybern., vol SMC-3, pp.610-621, Nov. 1973
- [15] R.M. HARALICK
“Statistical and structural approaches to texture”
Proc. IEEE, vol 67, n^o5, May 1979
- [16] R.M. HARALICK
“Statistical Image Texture Analysis”
Handbook of Pattern Recognition,
T.Y. Young and K.S. Fu eds., Academic Press, 1986
- [17] H. HILLION, P. MASSON, C. ROUX
“Une méthode de classification de textures par extraction linéaire non paramétrique de caractéristiques”
Traitement du Signal, vol 5, n^o4, 1988
- [18] Q.A. HOLMES, D.R. NÜESCH, R.A. SHUCHMAN
“Textural analysis and real-time classification of sea-ice types using SAR data”
IEEE Trans. on Geoscience and Remote Sensing, vol GE-22, n^o2, March 1984
- [19] J.Y. HSIAO, A.A. SAWCHUK
“Supervised textured image segmentation using feature smoothing
and probabilistic relaxation techniques”
IEEE PAMI, vol 11 , n^o 12, December 1989
- [20] F. JACQUET, H. NOEL, N. DERYCKE, J. DESMOUCEAUX, G. BUREL
“Application des réseaux de neurones à la Veille Panoramique Infra-Rouge”
Revue Technique THOMSON CSF, vol. 23, n^o 1, Mars 1991
- [21] M.E. JERNIGAN, F. D’ASTOUS,
“Entropy-based texture analysis in the spatial frequency domain”
IEEE PAMI, vol 6, n^o 2, March 1984
- [22] B. JULESZ
“Experiments in the visual perception of textures”
Scientific American, n^o232, April 1975
- [23] B. JULESZ, R. BERGEN
“Textons, the fundamental elements in preattentive vision”
The Bell System Technical Journal, vol 62, Jul-Aug. 1983, pp 1619-1645

- [24] R.L. KASHYAP, A. KHOTANZAD,
"A model-based method for rotation invariant texture classification"
IEEE PAMI, vol 8, n^o 4, July 1986
- [25] J.G. LEU, W.G. WEE
"Detecting the spatial structure of natural textures based on shape analysis"
Computer Vision, Graphics, and Image Processing 31 , 67-88 (1985)
- [26] R. LUMIA, R.M. HARALICK, O. ZUNIGA, L. SHAPIRO
T.C. PONG, F.P. WANG
"Texture analysis of aerial photographs"
Pattern Recognition, vol 16, n^o 1, pp 39-46, 1983
- [27] E. MESROBIAN, J. SKRZYPEK
"Discrimination of natural textures : a neural network architecture"
Proc. of the IEEE ICNN, San Diego, june 21-24th 1987, pp IV.247 to IV.258
- [28] T.R. REED, H. WECHSLER,
"Segmentation of textured images and gestalt organization
using spatial/spatial-frequency representations"
IEEE PAMI, vol 12, n^o 1, January 1990
- [29] J. RONSIN, D. BARBA, S. RABOISSON
"Comparison between co-occurrence matrices, local histograms
and curvilinear integration for texture characterization"
SPIE Symposium vol 596, Cannes 1985
- [30] D.E. RUMELHART, G.E. HINTON, R.J. WILLIAMS
"Learning internal representations by error backpropagation"
Parallel Distributed Processing, D.E. RUMELHART and J.L. Mc CLELLAND
Chap8, Bradford book - MIT Press - 1986
- [31] C. SUN, W.G. WEE
"Neighboring gray level dependence matrix for texture classification"
Computer Vision, Graphics, and Image Processing 23, 341-352 (1983)
- [32] A. TIRAKIS, L. SUKISSIAN, S. KOLLIAS
"An adaptative technique for segmentation and classification of textured images"
Proc. of the ICNN90, Paris, july 9-13th, 1990, pp 31-34
- [33] F. TOMITA, Y. SHIRAI, S. TSUJI
"Description of textures by structural analysis"
IEEE PAMI, vol 4, n^o2, March 1982
- [34] M. TÜCERYAN, A.K. JAIN
"Texture segmentation using Voronoi polygons"
IEEE PAMI, vol 12, n^o 2, February 1990
- [35] F. VALLET, E. PERNOT
"NeuroClass : Manuel d'utilisation (1.1)"

Thomson CSF/LCR, 3 avril 1990

- [36] A.L. VICKERS, J.W. MODESTINO
“A maximum likelihood approach to texture classification”
IEEE PAMI, vol 4, n^o 1, January 1982

- [37] F.M. VILNROTTER, R. NEVATIA, K.E. PRICE
“Structural analysis of natural textures”
IEEE PAMI, vol 8, n^o 1, January 1986

- [38] H. WECHSLER
“Texture Analysis - A survey”
Signal Processing 2 (1980) 271-282

- [39] Xia XIE, André SMOLARZ
“Un nouveau modèle Markovien pour la segmentation de textures”
7^e congrès AFCET, Paris, 29 nov. 1989, pp 683-692

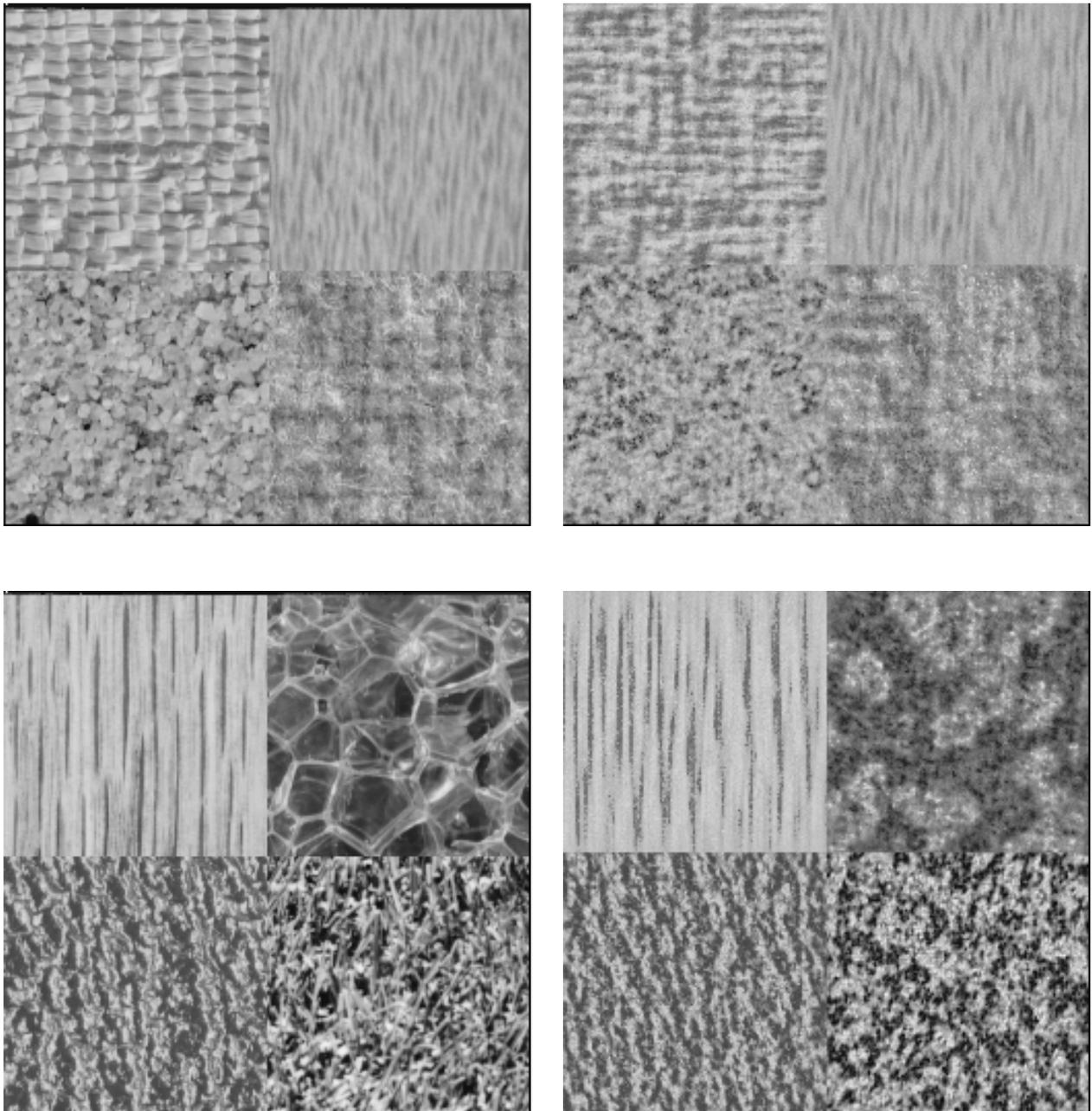


FIG. 5: Textures de Brodatz (à gauche) et leur synthèse (à droite)

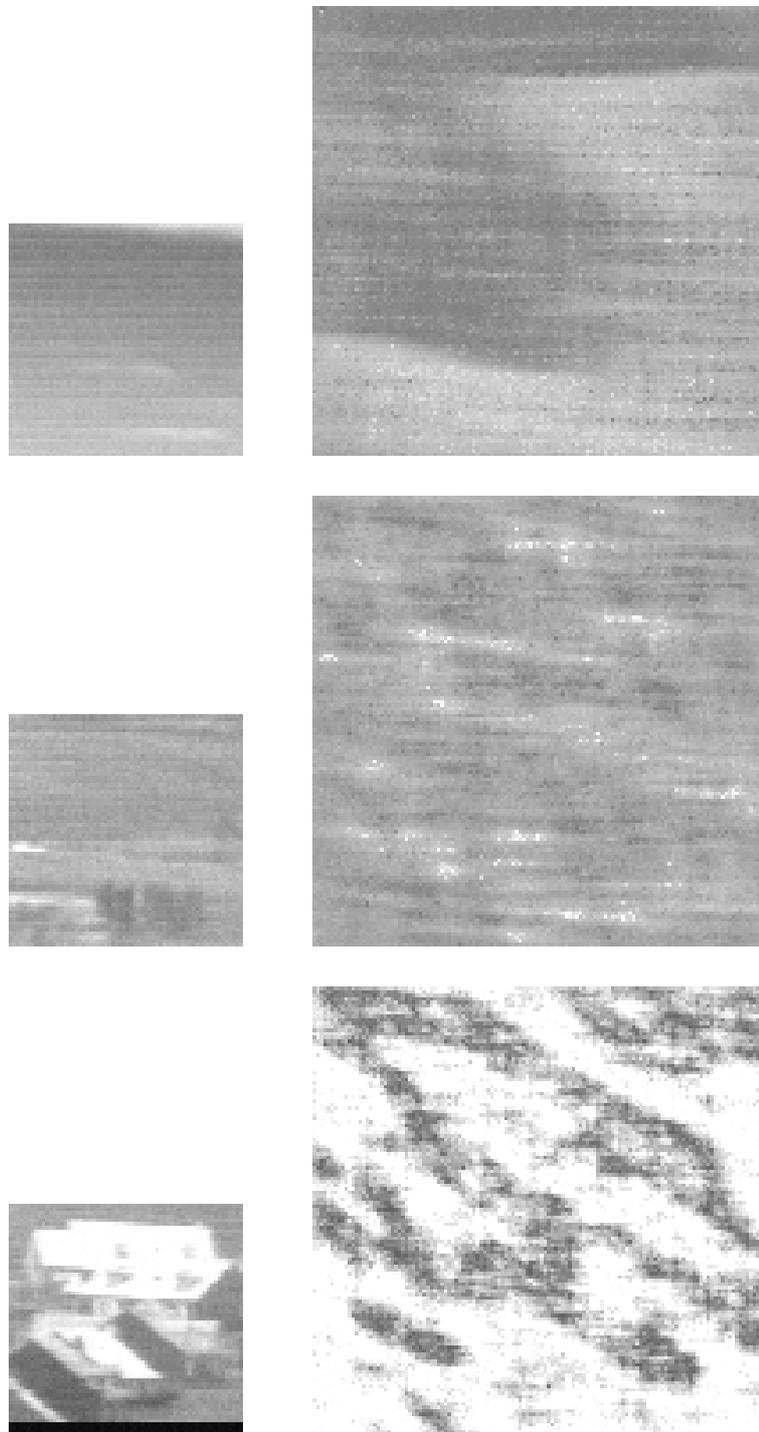


FIG. 6: Textures Infra-Rouge ciel, terre, et habitations (à gauche) et leur synthèse (à droite)

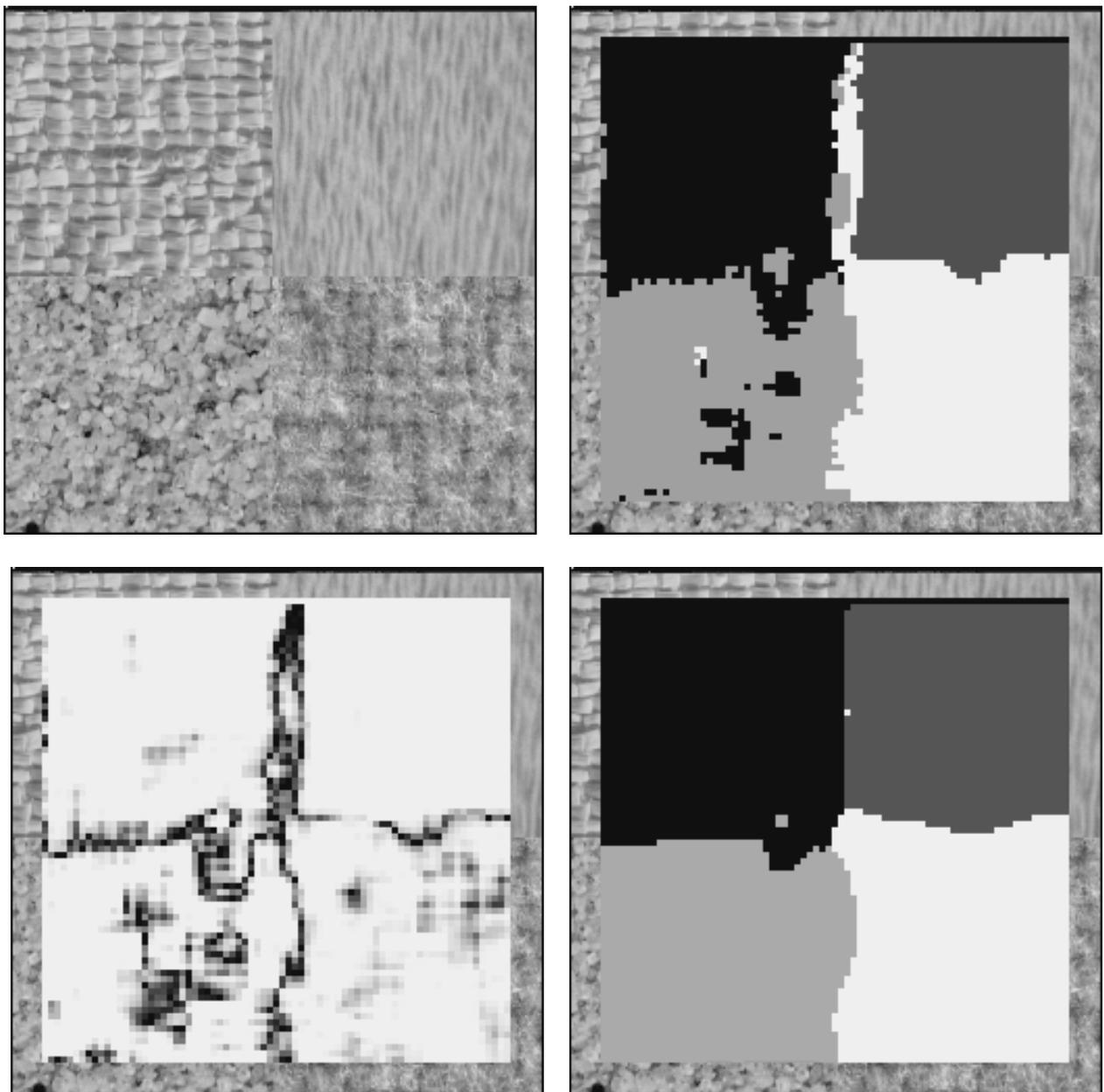


FIG. 7: Classification de textures de Brodatz

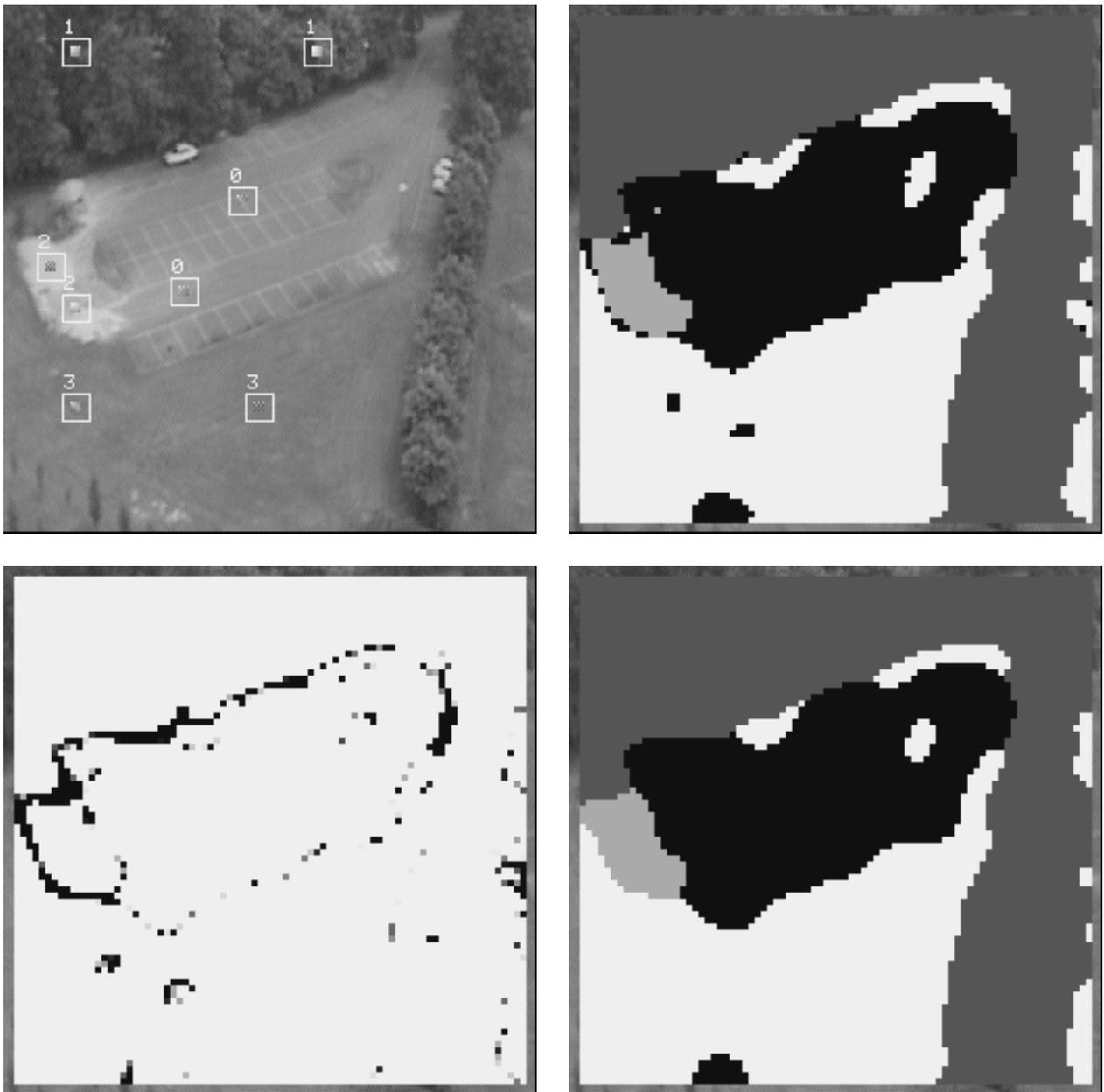


FIG. 8: Classification sur image aérienne

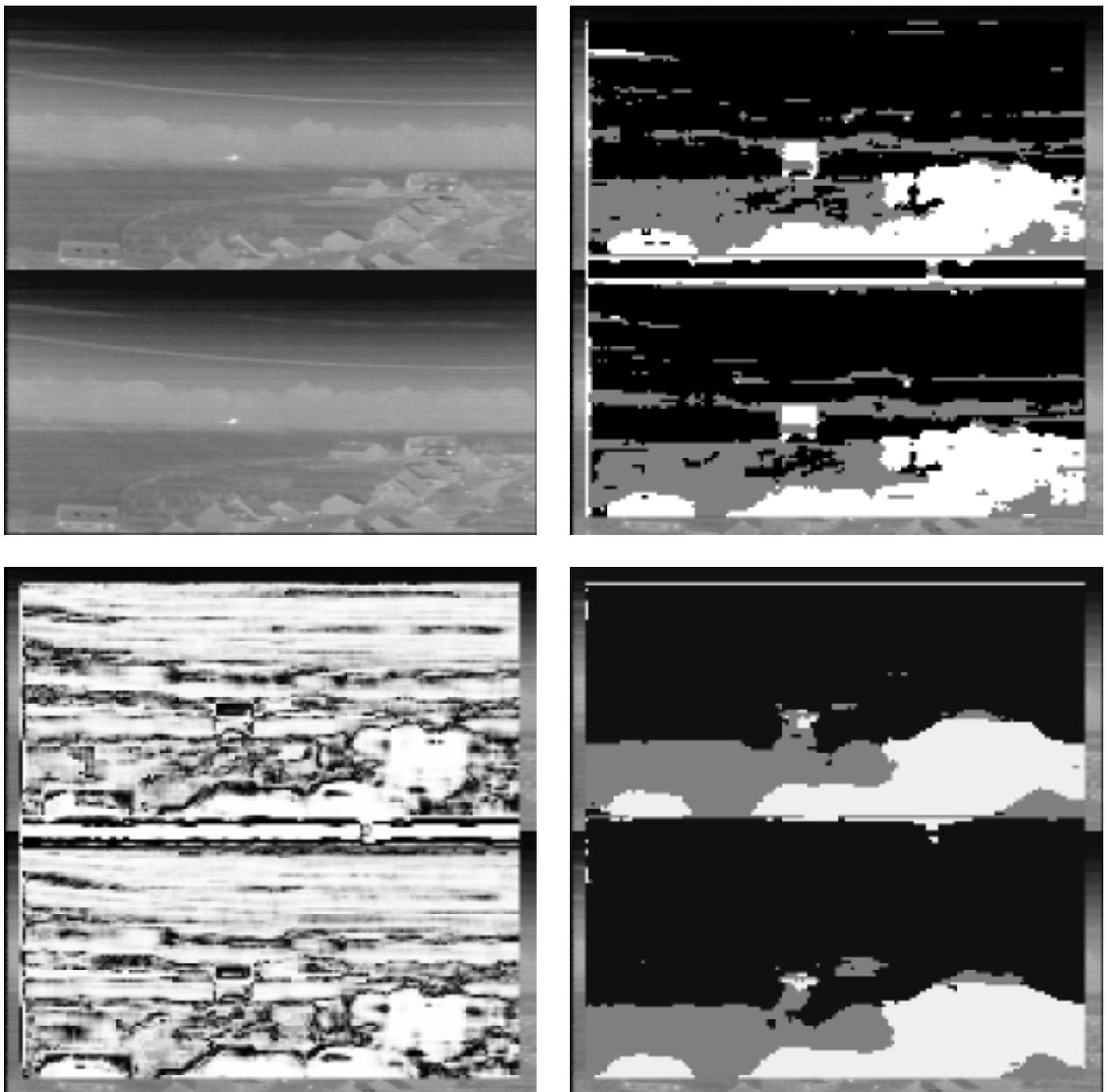


FIG. 9: Classification sur le site du Nord de la France