

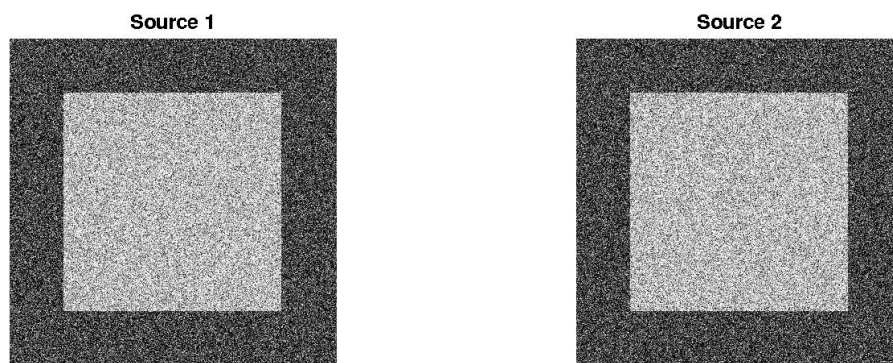
# TP1 : Segmentation par fusion de données à base de fonctions de croyance

## Objectif du TP :

Mettre en évidence l'intérêt de modéliser l'imprécision des sources en plus de leur incertitude pour améliorer l'estimation ou la décision. Application à la segmentation d'images.

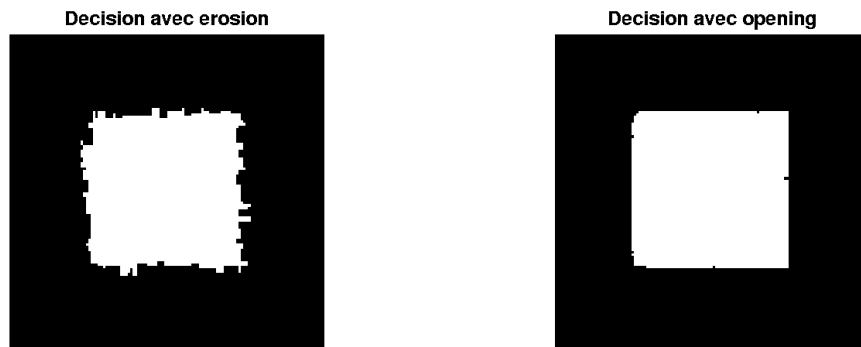
## I- Segmentation 2-classes à partir d'images issues de sources bruitées

Deux capteurs fournissent deux images d'observation bruitées correspondant à une même image des labels ayant 2 classes :  $A$  et  $B = \bar{A}$ . Le bruit (centré) présent sur ces images d'observation (qui vous sont fournies à partir de simulations) correspond à différentes distributions gaussiennes de paramètres respectifs  $s_1$  et  $s_2$ . Notez que même si  $s_1 = s_2$ , les réalisations du bruit sont différentes et décorrélées. La Figure 1 présente un exemple d'images d'observation. Les deux images sont en niveaux de gris dans l'intervalle  $[0,1]$ .



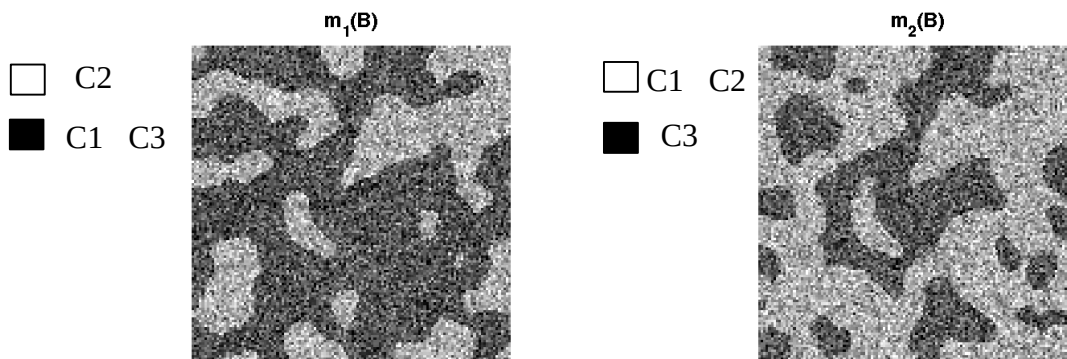
1. Soit, pour chacune des deux sources  $i$ , à partir de l'image d'observation  $I_i$ , en chaque pixel  $s$ , une bba telle que  $m_i(A) = I_i(s)$  et  $m_i(B) = 1 - m_i(A)$  avec  $B = \bar{A}$ . Utilisez une 'image de bba' telle que la troisième dimension de l'image (les deux premières étant les lignes et les colonnes) représente les différentes hypothèses de  $2^\Omega$  (dans l'ordre  $A, B, A \cup B, \emptyset$  puisque les indices des tableaux en Matlab commencent à 1). Calculez et affichez l'image des valeurs de  $m_i(A)$  et  $m_i(B)$  pour chacune des sources.
2. Pour chacune des deux sources  $i$ ,  $i \in \{1, 2\}$ , calculez les érosions, notées  $m'_i(A)$  et  $m'_i(B)$ , de  $m_i(A)$  et  $m_i(B)$  pour un rayon  $d$  de l'élément structurant (faire une fonction Matlab spécifique pour l'érosion). Variez  $d$  pour valider votre code.
3. En chaque pixel, déduisez  $m'_i(\Omega) = 1 - (m'_i(A) + m'_i(B))$ , avec  $\Omega = A \cup B$ . Commentez l'image des valeurs de  $m_i(A)$ ,  $m_i(B)$  et  $m_i(\Omega)$  pour chacune des sources.
4. Effectuez la combinaison conjonctive des bbas  $m'_1$  et  $m'_2$ .
5. En chaque pixel, calculez les valeurs de BetP, décidez et affichez le résultat de classification.
6. Pour chacune des deux sources  $i$ ,  $i \in \{1, 2\}$ , calculez les ouvertures, notées  $m''_i(A)$  et  $m''_i(B)$ , de  $m_i(A)$  et  $m_i(B)$  pour un rayon  $d$  de l'élément structurant (faire une fonction Matlab spécifique pour l'ouverture ; on rappelle que, pour un élément structurant symétrique, l'ouverture est obtenue par dilatation de l'érodé). Variez  $d$  pour valider votre code.

7. Refaites les étapes 3., 4., et 5. à partir de  $m''_i(A)$  et  $m''_i(B)$ .
8. Faites varier  $d$  et commentez les résultats obtenus respectivement par les deux allocations de masse 2. et 6. A titre d'illustration, la Figure 2 donne des exemples de résultats.

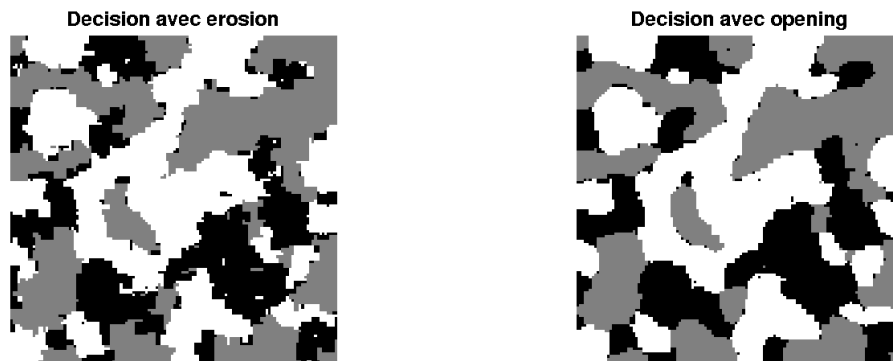


## II- Segmentation 3-classes à partir d'images issues de sources bruitées

On étend le cas précédent au cas d'une classification 3 classes, C1, C2 et C3 à partir de capteurs distinguant respectivement 1 classe parmi les 3 et confondant les deux autres entre elles comme dans l'exemple du cours. Comme précédemment, les capteurs présentent un bruit (centré) correspondant à différentes distributions gaussiennes de paramètres respectifs  $s_1$  et  $s_2$ . La Figure 3 présente un exemple d'images d'observation (niveaux de gris dans  $[0,1]$ ).



1. Pour chacune des deux sources  $i$ , à partir de l'image d'observation  $I_i$ , on définit en chaque pixel, une bba initiale avec deux éléments focaux que vous définirez. Pour représenter les différentes hypothèses, vous utiliserez la notation binaire: par exemple, l'hypothèse C1 est codée 001 et l'hypothèse  $\{C1, C3\}$  ou  $C1 \cup C3$  est codée 101. Calculez et affichez l'image des valeurs de quatre éléments focaux (deux par source).
2. Selon la façon dont vous avez codé la règle conjonctive à la question I.4., implémentez la pour  $|\Omega| \geq 2$  (cas général).
3. Reprenez les questions I.2 à I.6 pour faire la classification 3 classes. Pour afficher la classification associez les classes  $\{C1, C2, C3\}$  aux niveaux de gris  $\{1, 0.5, 0\}$ .
4. Faites varier  $d$  et commentez les résultats obtenus respectivement par les deux allocations de masse. A titre d'illustration, la Figure 4 donne des exemples de résultats.



### III – Régularisation par fusion avec une bba de voisinage

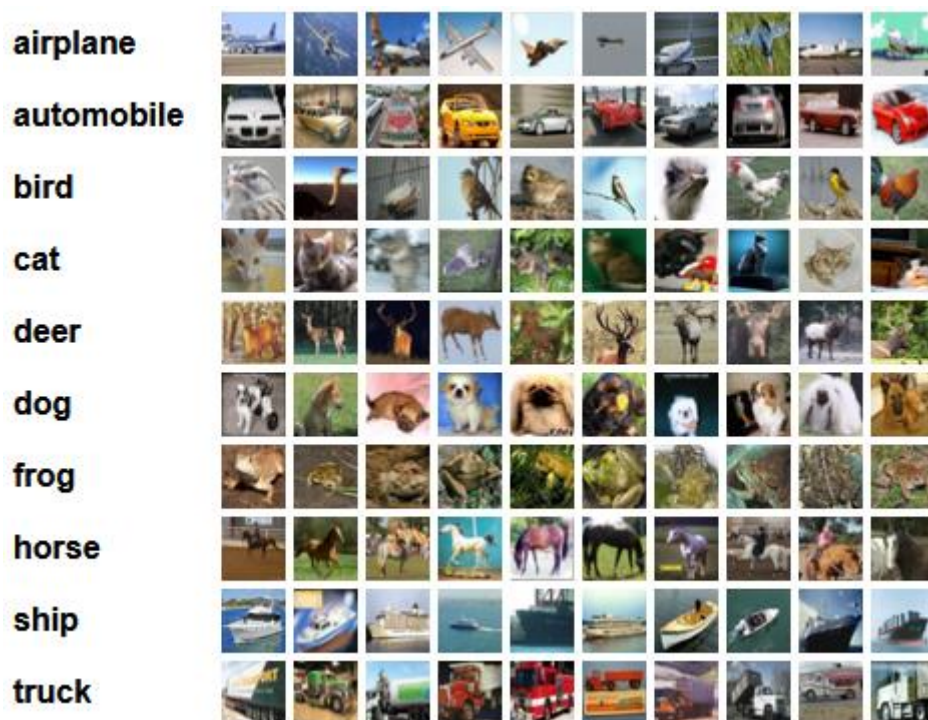
On souhaite à présent intégrer l'information de voisinage à la décision via la fusion avec une bba représentant l'information de voisinage comme vu en cours (slides 78 et 80).

1. Étudiez la fonction 'Reg\_BBA.m' et expliquez en le fonctionnement. Vous pourrez commenter les lignes une à une.
2. Testez la fonction et comparez les résultats à ceux obtenus précédemment.

## TP2 : Classification par fusion de classifieurs à base de fonctions de croyance

### Objectif du TP :

Mettre en évidence l'intérêt de modéliser l'imprécision des sources en plus de leur incertitude pour améliorer l'estimation ou la décision. Application à la classification d'images CIFAR10 (<https://www.cs.toronto.edu/~kriz/cifar.html>).



Le code fourni est un code à trous que vous devrez compléter après avoir recopié chez vous le projet donné au lien :

<https://colab.research.google.com/drive/1ArVt-qA25nQphg259fTeVg8DqsqoKt6g?usp=sharing>

Par ailleurs, nous vous conseillons d'utiliser les GPUs disponibles sous google colab pour accélérer les temps de calcul et (au moins dans un premier temps) d'utiliser les poids pré-entraînés pour les classifieurs/dichotomizers à utiliser dans le TP.

### I- Classification à partir de deux classifieurs 10-classes

On considère deux classifieurs légers (un CNN et un Fully-connected NN) entraînés sur le dataset CIFAR10. Chacun fournit en sortie un vecteur de scores normalisés par softmax sur les 10 classes du dataset.

1. Complétez la fonction `calc_betP(bbas)` qui, pour chaque élément ligne d'un tenseur de vecteurs de `bbas`, calcule le tenseur associé de vecteurs de probabilité pignistique. Pour calculer le cardinal d'un élément focal `i`, vous pourrez utiliser `i.bit_count()`, et les opérateurs logiques bit à bit tels que `np.bitwise_and()`

2. Complétez la fonction `alloc_consonant_bba(predictions)` qui, pour chaque élément ligne d'un tenseur de vecteurs de scores, crée une BBA consonante telle que la probabilité pignistique associée à cette BBA soit égale au vecteur de scores initial. Pour cela, vous coderez l'algorithme~1.
3. Complétez la fonction `conjunctive_rule(bbas1, bbas2)` qui prend deux tenseurs de BBAs en entrée et réalise la combinaison conjonctive entre les BBAs représentées sur chaque élément ligne des tenseurs respectifs.
4. Observez les résultats de la combinaison des deux classifieurs CNN et Fully-connected NN fournis dans le code. Qu'en pensez-vous ?

$i_1 = \underset{i \in \Omega}{\operatorname{argmin}}(s_i)$ $m^\Omega(\Omega) = s_{i_1} \times  \Omega $ <p>Pour <math>j \in [2,  \Omega ]</math> :</p> $i_j = \underset{i \in \Omega, i \notin \{i_1, \dots, i_{j-1}\}}{\operatorname{argmin}}(s_i)$ $H_j = \Omega \setminus \{i_1, \dots, i_{j-1}\}$ $m^\Omega(H_j) = (s_{i_j} - s_{i_{j-1}}) \times ( \Omega  + 1 - j)$
--

Algorithme~1 : Allocation de BBA consonnante

## II- Classification à partir de deux classifieurs 10-classes et de dichotomizers

Pour séparer les classes difficiles à séparer, on se propose d'ajouter des dichotomizers dédiés. Les paires de classes identifiées comme difficiles à séparer sont les paires {chat, chien}, {oiseau, chevreuil} et {automobile, camion}. On va donc ajouter trois dichotomizers qui ont la même architecture mais ont été entraînés sur des datasets différents : les restrictions respectives du dataset CIFAR10 aux classes dans la paire concernée.

1. Complétez la fonction `alloc_dichoto_bba(predictions, n_classes, label_val)` qui, pour chaque élément ligne d'un tenseur de vecteurs de scores  $(s_i, s_j)$ , crée une BBA à deux éléments focaux qui sont les complémentaires dans  $\Omega$  des hypothèses  $(\omega_i, \omega_j)$  considérées par le dichotomizer :  $m^\Omega(\Omega \setminus \omega_i) = s_j$ ,  $m^\Omega(\Omega \setminus \omega_j) = s_i$ . Expliquez pourquoi on a choisi cette allocation.
2. Complétez la fonction `merge_2cl_bbas(bbas, nb_lab, i, j)` qui transfère la masse potentielle sur les deux hypothèses indicées  $i$  et  $j$  vers la disjonction de ces deux hypothèses  $\{\omega_i, \omega_j\}$ . Expliquez pourquoi il est souhaitable (voire nécessaire) de réaliser cet affaiblissement sur les BBAs des classifieurs 10 classes.
3. Observez les résultats de la combinaison des deux classifieurs CNN et Fully-connected NN et des trois dichotomizers. Qu'en concluez-vous ?
4. Proposez vos propres améliorations. Selon le temps restant, testez-les.