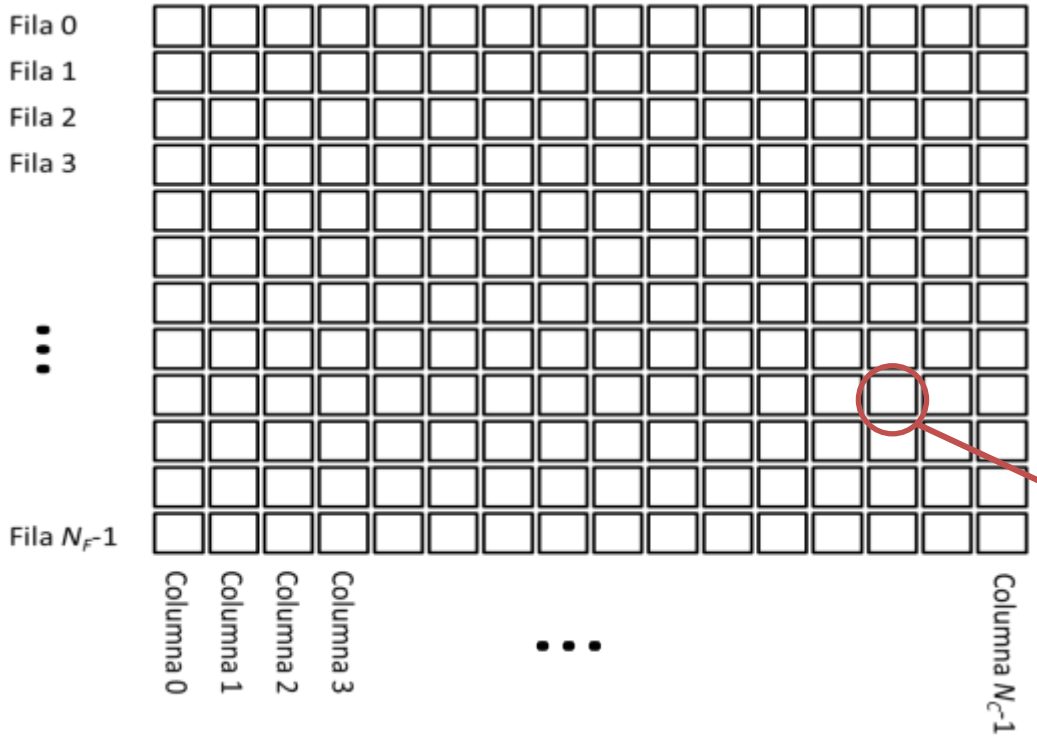
# Auto-Organización y Segmentación de Imágenes

UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS

Marco A. Alzate

# Imagen Digital

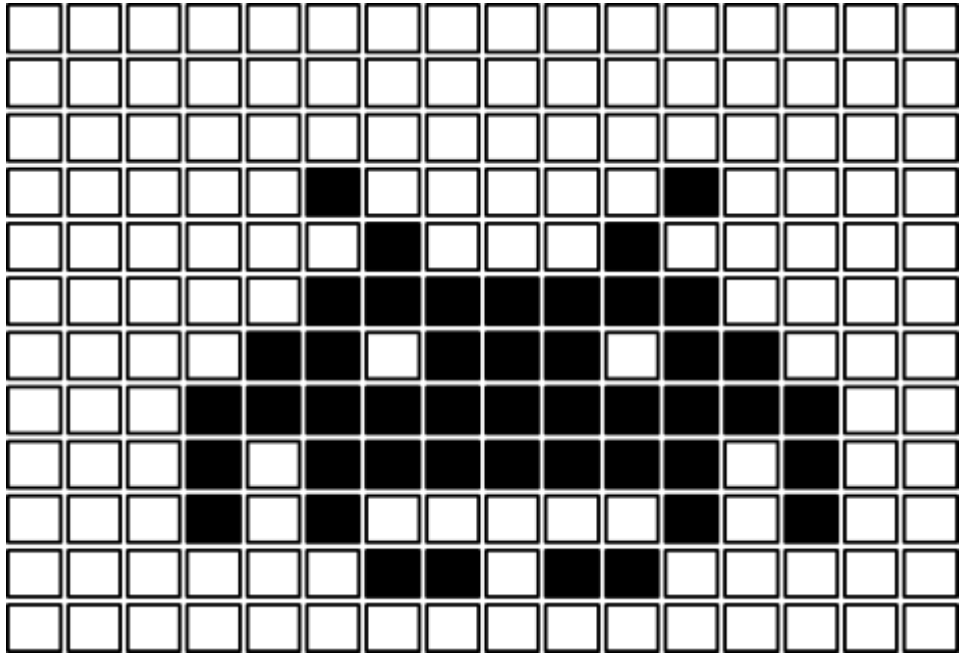
$$I : \mathbb{Z}_{N_F} \times \mathbb{Z}_{N_C} \rightarrow \wp$$




Una imagen es un arreglo de $N_F$ filas por $N_C$ columnas, donde a cada elemento del arreglo le corresponde un elemento del codominio, $\wp$.

# Imagen binaria

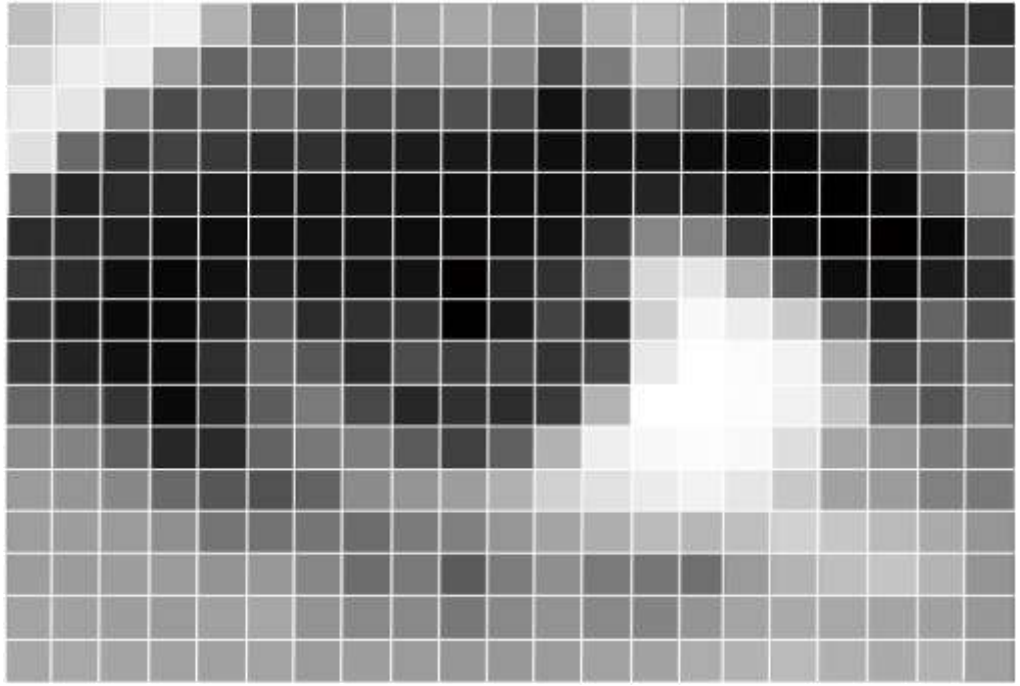
$$I_B : \mathbb{Z}_{N_F} \times \mathbb{Z}_{N_C} \rightarrow \{0,1\}$$



Una imagen binaria es un arreglo de $N_F$ filas por $N_C$ columnas, donde a cada elemento del arreglo le corresponde un elemento del codominio $\mathbb{Z}_2 = \{0,1\}$

# Imagen en grises

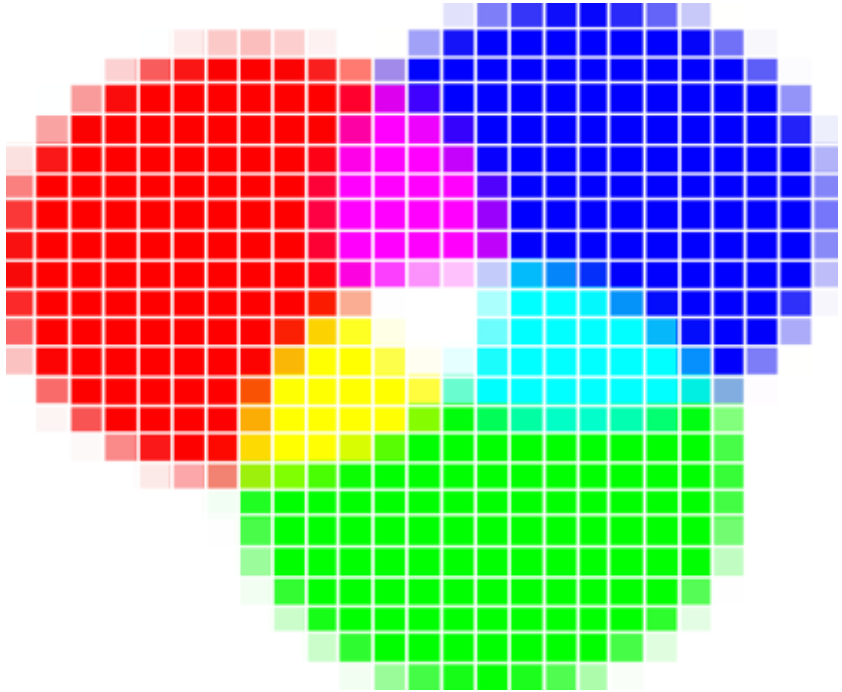
$$I_G : \mathbb{Z}_{N_F} \times \mathbb{Z}_{N_C} \rightarrow \mathbb{Z}_L$$



Una imagen en grises es un arreglo de $N_F$ filas por $N_C$ columnas, donde a cada elemento del arreglo le corresponde un elemento del codominio $\mathbb{Z}_L = \{0,1,\ldots,L-1\}$ . El valor cero se representa como negro, el valor $L$-1 como blanco y los valores intermedios se representan como tonos intermedios de gris.

# Imagen en color

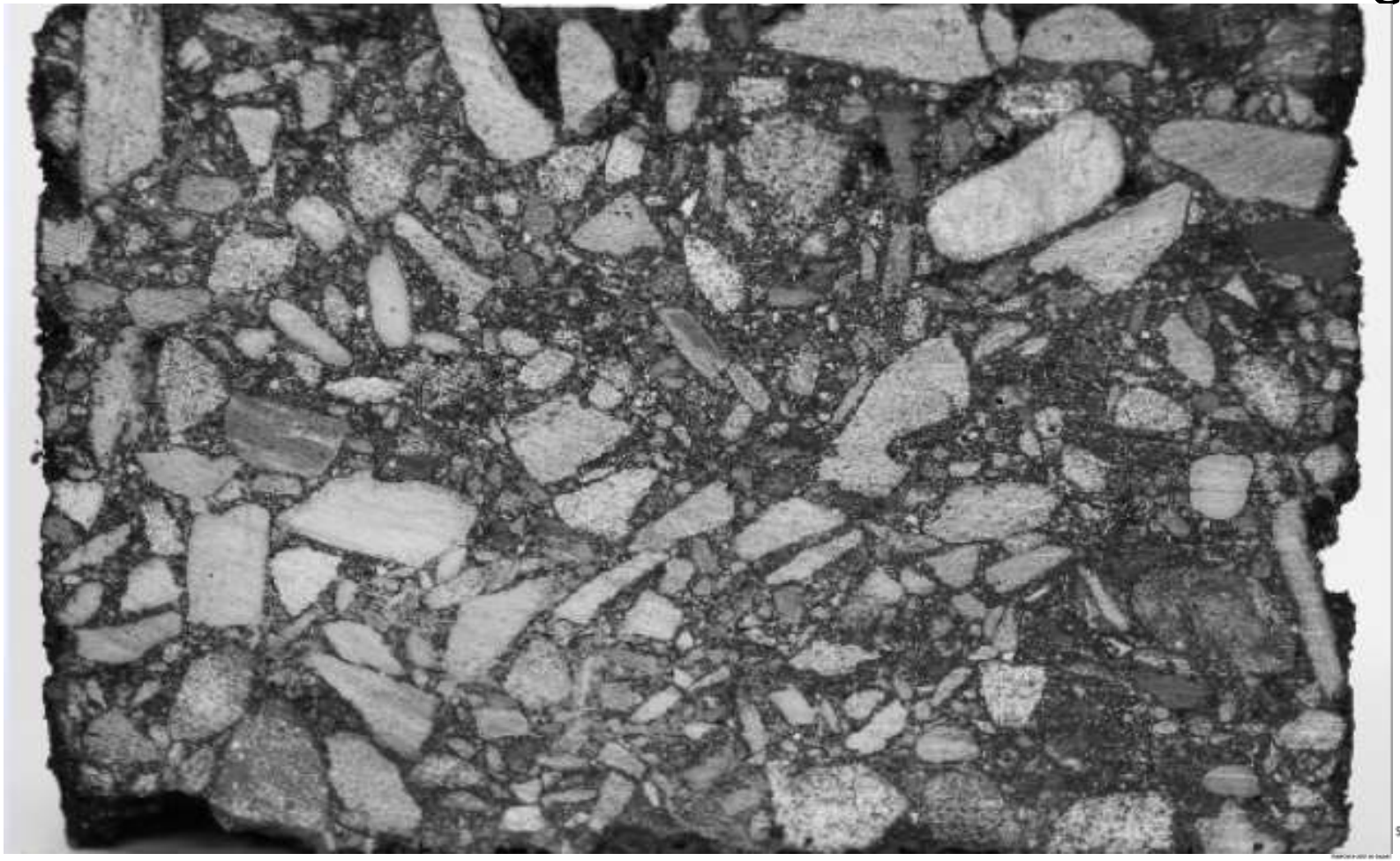
$$I_C : \mathbb{Z}_{N_F} \times \mathbb{Z}_{N_C} \rightarrow \mathbb{Z}_L \times \mathbb{Z}_L \times \mathbb{Z}_L$$



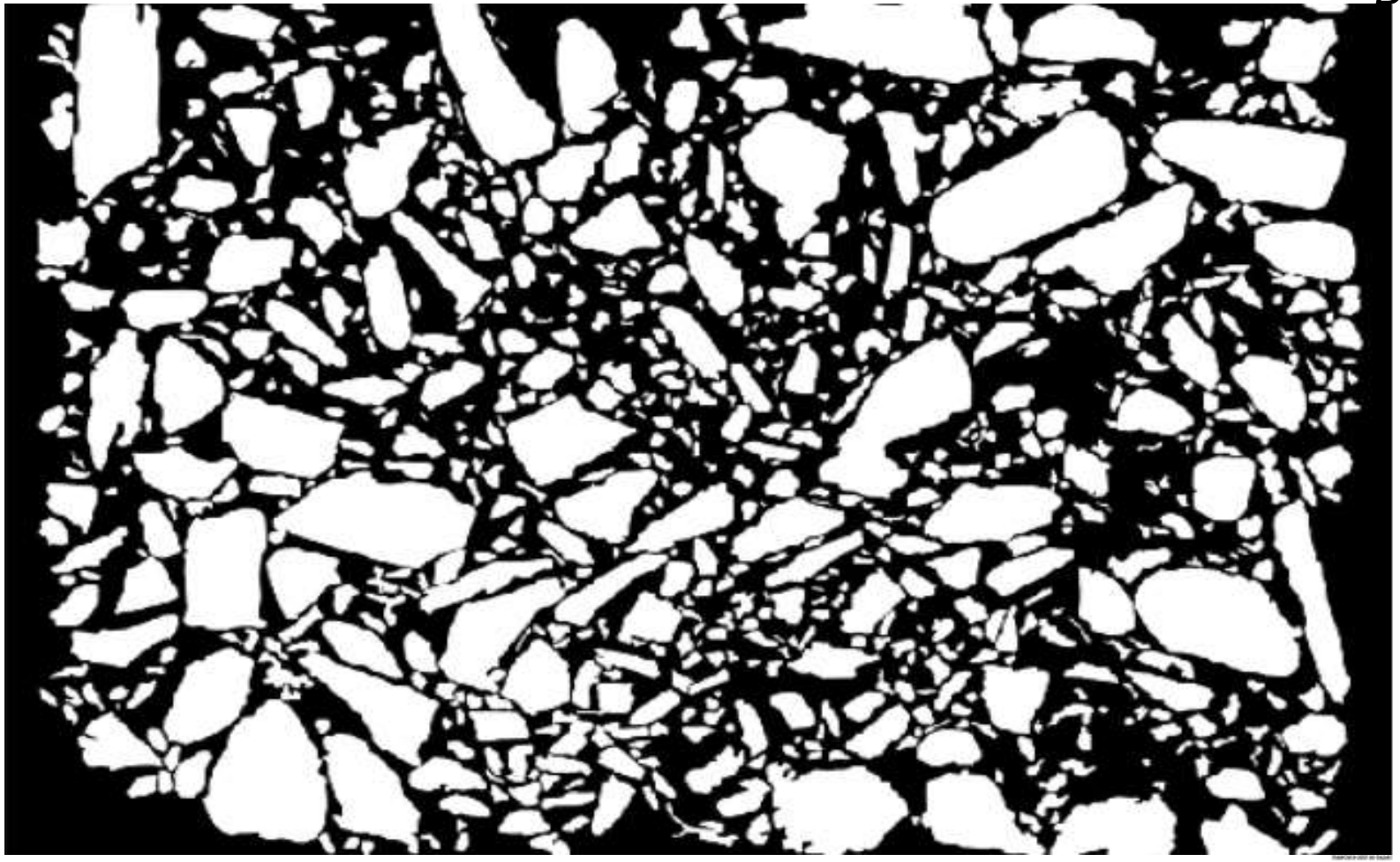
Una imagen en color es un arreglo de $N_F$ filas por $N_C$ columnas, donde a cada elemento del arreglo le corresponde un elemento del codominio $\mathbb{Z}_L{}^3$, correspondiendo a intensidades de tres capas espectrales: rojo, verde y azul.

# Ejemplo : Muestra de mezcla asfáltica, $I_C$

# Ejemplo : Muestra de mezcla asfáltica, $I_G$



# Ejemplo : Muestra de mezcla asfáltica, $I_B$



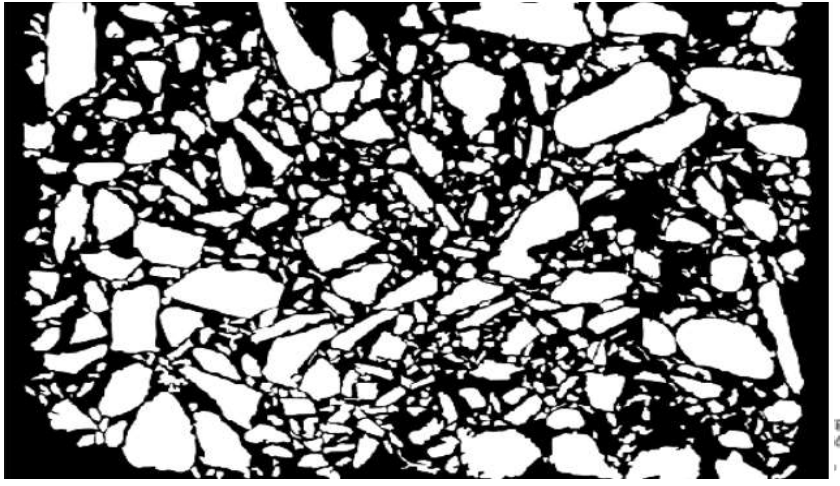
# Ejemplo : Segmentación de una imagen

Hemos dividido la imagen en regiones, de manera que a cada pixel lo hemos clasificado como gravilla o (asfalto o fondo):

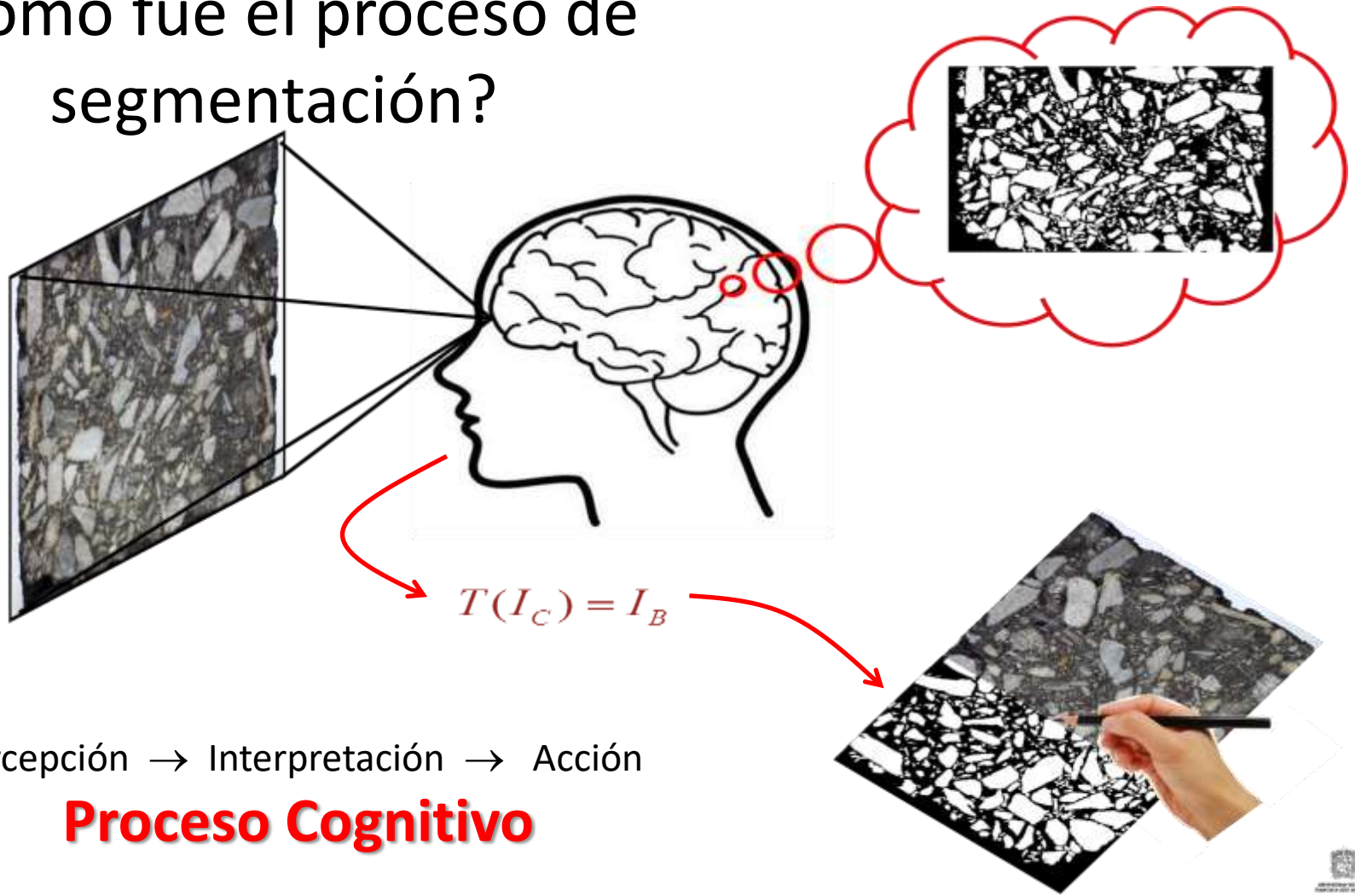
$$T(I_C) = I_B$$

Observemos bien las imágenes:
¿usamos el color?
¿la textura?
¿los bordes?

**¡La segmentación es una tarea MUY difícil y está lejos de estar resuelta!**



# ¿Cómo fue el proceso de segmentación?




Percepción $\rightarrow$ Interpretación $\rightarrow$ Acción

**Proceso Cognitivo**

# Sistema Dinámico Cognitivo

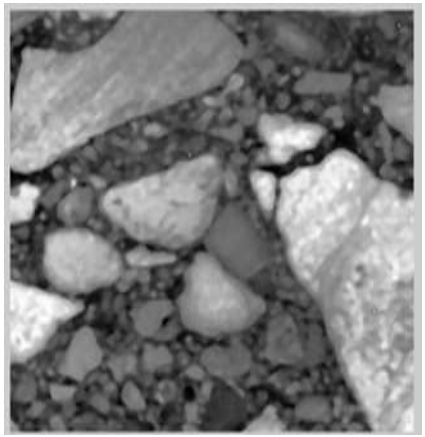
*"Veo la emergencia de un* (…) *campo integrador de múltiples disciplinas* (…) *llamado "*Sistemas Dinámicos Cognitivos*" que se construye sobre las ideas ya bien establecidas del procesamiento estadístico de señales, el control estocástico y la teoría de la información, y las entreteje con nuevas ideas tomadas de la neurociencia, el aprendizaje estadístico y la teoría de juegos* (…). *El marco de referencia serán las capacidades cognitivas del ser humano (percepción, memoria, atención, inteligencia, lenguaje, acción), bajo la perspectiva de que ellas constituyen un sistema de computación".*

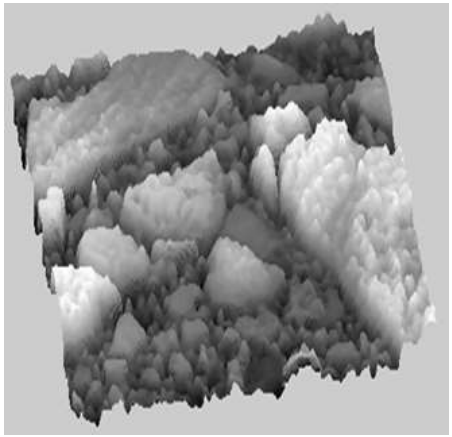
Simon Haykin, 2006-2011

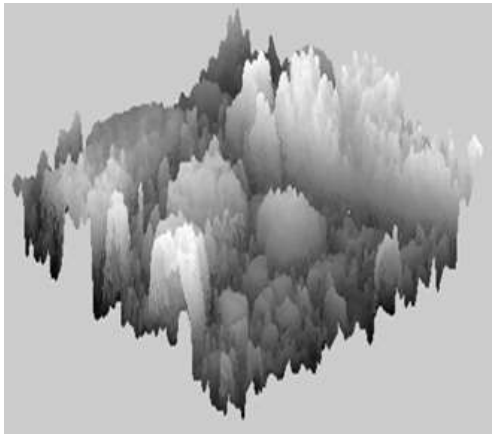
# Primera característica: Nivel de gris (La gravilla suele ser clara y el asfalto oscuro)

$$I_S(x,y)=\begin{cases}0 & si \quad I_G(x,y)<l_0\\ 1 & si \quad I_G(x,y)\geq l_0\end{cases} \qquad \forall(x,y)\in\mathbb{Z}_{N_V}\times\mathbb{Z}_{N_H}$$

Segmentación mediante un nivel de comparación $l_0$



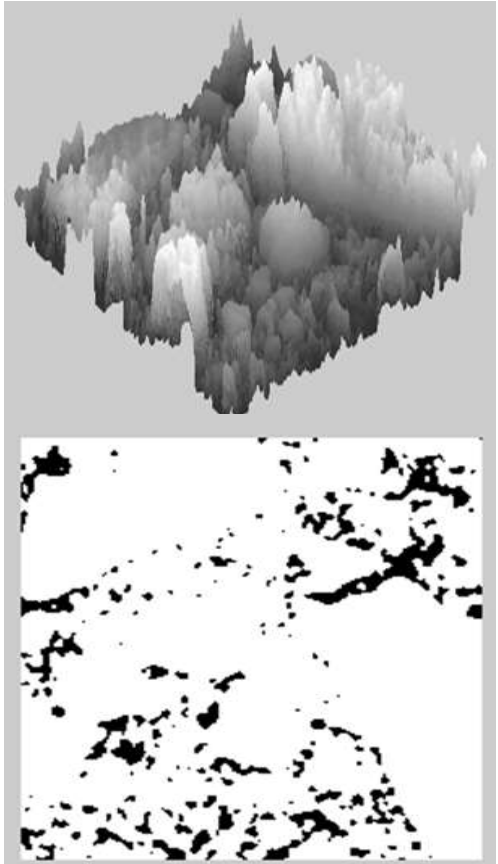


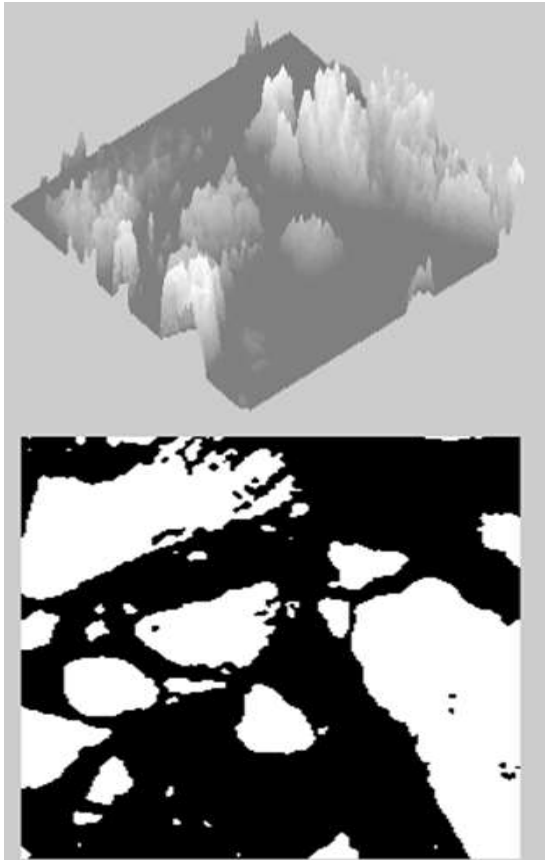


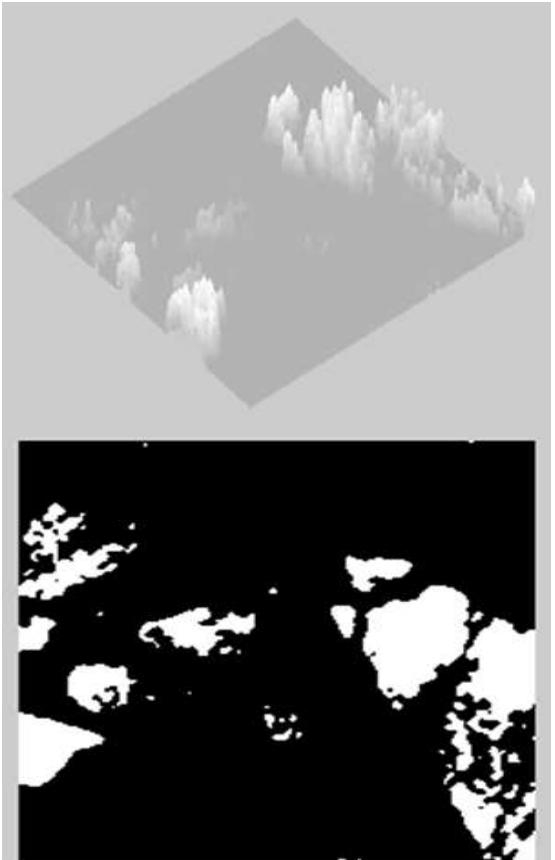
# Primera característica: Nivel de gris (La gravilla suele ser clara y el asfalto oscuro)

$$I_S(x,y)=\begin{cases}0 & si \quad I_G(x,y)<l_0\\ 1 & si \quad I_G(x,y)\geq l_0\end{cases} \qquad \forall(x,y)\in\mathbb{Z}_{N_V}\times\mathbb{Z}_{N_H}$$

Segmentación mediante un nivel de comparación $l_0$







# Primera característica: Nivel de gris (La gravilla suele ser clara y el asfalto oscuro)

$N_P = N_F \cdot N_C = \sum_{l=0}^{L-1} n_l$ donde $n_l =$ Número de pixeles en el nivel $l, l \in \mathbb{Z}_L$

$p_l = \Pr[\text{escoger un pixel con nivel de gris } l] = \frac{n_l}{N_P}, \qquad l \in \mathbb{Z}_L$

$\mu = \sum_{l=0}^{L-1} l \cdot p_l, \qquad \sigma^2 = \sum_{l=0}^{L-1} (l-\mu)^2 \cdot p_l$ Valor promedio y varianza del nivel de gris de un pixel tomado al azar

$$I_B(x,y) = \begin{cases} 0 & si \quad I_G(x,y) < l_0 \\ 1 & si \quad I_G(x,y) \geq l_0 \end{cases} \qquad \forall (x,y) \in \mathbb{Z}_{N_V} \times \mathbb{Z}_{N_H}$$

Segmentación mediante un nivel de comparación $l_0$

$p_A = \sum_{l=0}^{l_0-1} p_l, \quad p_G = \sum_{l=l_0}^{L-1} p_l$ Probabilidad de que un pixel escogido al azar sea ***A***(sfalto) o ***G***(ravilla)

$\mu_A = \frac{1}{p_A} \sum_{l=0}^{l_0-1} l \cdot p_l, \qquad \mu_G = \frac{1}{p_G} \sum_{l=l_0}^{L-1} l p_l,$

$\sigma_A^2 = \frac{1}{p_A} \sum_{l=0}^{l_0-1} (l-\mu_A)^2 \cdot p_l, \qquad \sigma_G^2 = \frac{1}{p_G} \sum_{l=l_0}^{L-1} (l-\mu_G)^2 \cdot p_l$

Medias y varianzas de cada clase, ***G***(ravilla) y ***A***(sfalto)

$$l_0^* = \arg\max_{l_0=0,1,\ldots,L-1} p_A (\mu - \mu_A)^2 + p_G (\mu - \mu_G)^2 = \arg\min_{l_0=0,1,\ldots,L-1} p_A \sigma_A^2 + p_G \sigma_G^2$$

Nivel óptimo que logra la mínima variabilidad dentro de cada clase y la máxima variabilidad entre clases.

# Primera característica: Nivel de gris (La gravilla suele ser clara y el asfalto oscuro)

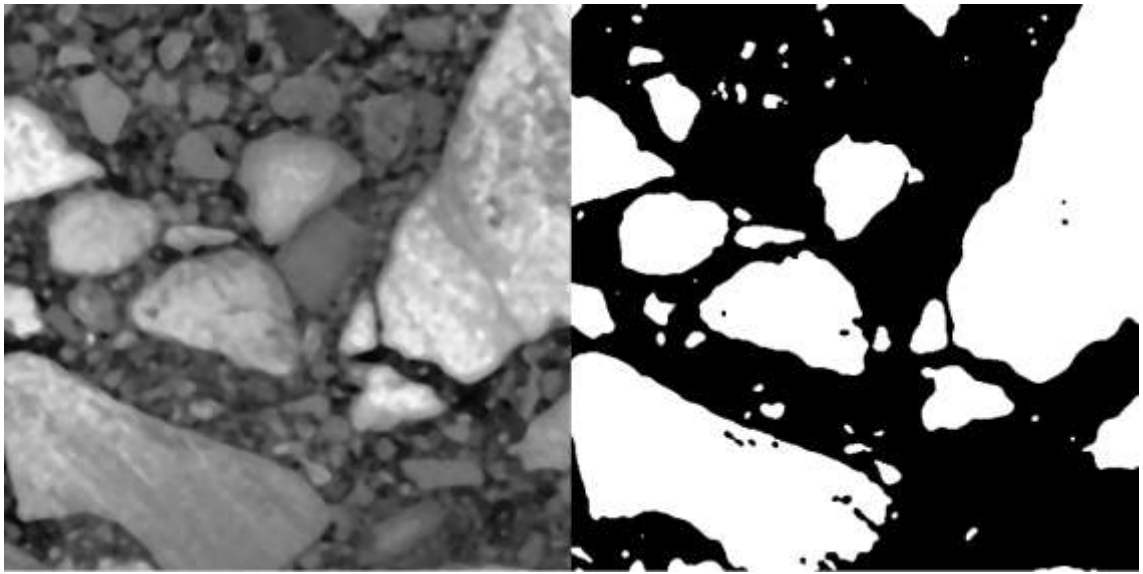
$$I_S(x,y) = \begin{cases} 0 & si \quad I_G(x,y) < l_0 \\ 1 & si \quad I_G(x,y) \geq l_0 \end{cases} \qquad \forall (x,y) \in \mathbb{Z}_{N_V} \times \mathbb{Z}_{N_H}$$

Segmentación mediante un nivel de comparación $l_0$

$$l_0^* = \arg\max_{l_0=0,1,\ldots,L-1} p_A\left(\mu - \mu_A\right)^2 + p_G\left(\mu - \mu_G\right)^2 = \arg\min_{l_0=0,1,\ldots,L-1} p_A\sigma_A^2 + p_G\sigma_G^2$$

Nivel óptimo que logra la mínima variabilidad dentro de cada clase y la máxima variabilidad entre clases.

**Método de Otsu**



# Segunda característica: Semejanzas y diferencias entre pixeles y grupos de pixeles

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}a+b\\ a-b\end{bmatrix}$$

→ Semejanzas

→ Diferencias

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\left(\frac{1}{\sqrt{2}}\begin{bmatrix}a+b\\ a-b\end{bmatrix}\right)=\begin{bmatrix}a\\ b\end{bmatrix}$$

La transpuesta es la inversa (matriz ortonormal)

Transformada wavelet de Haar

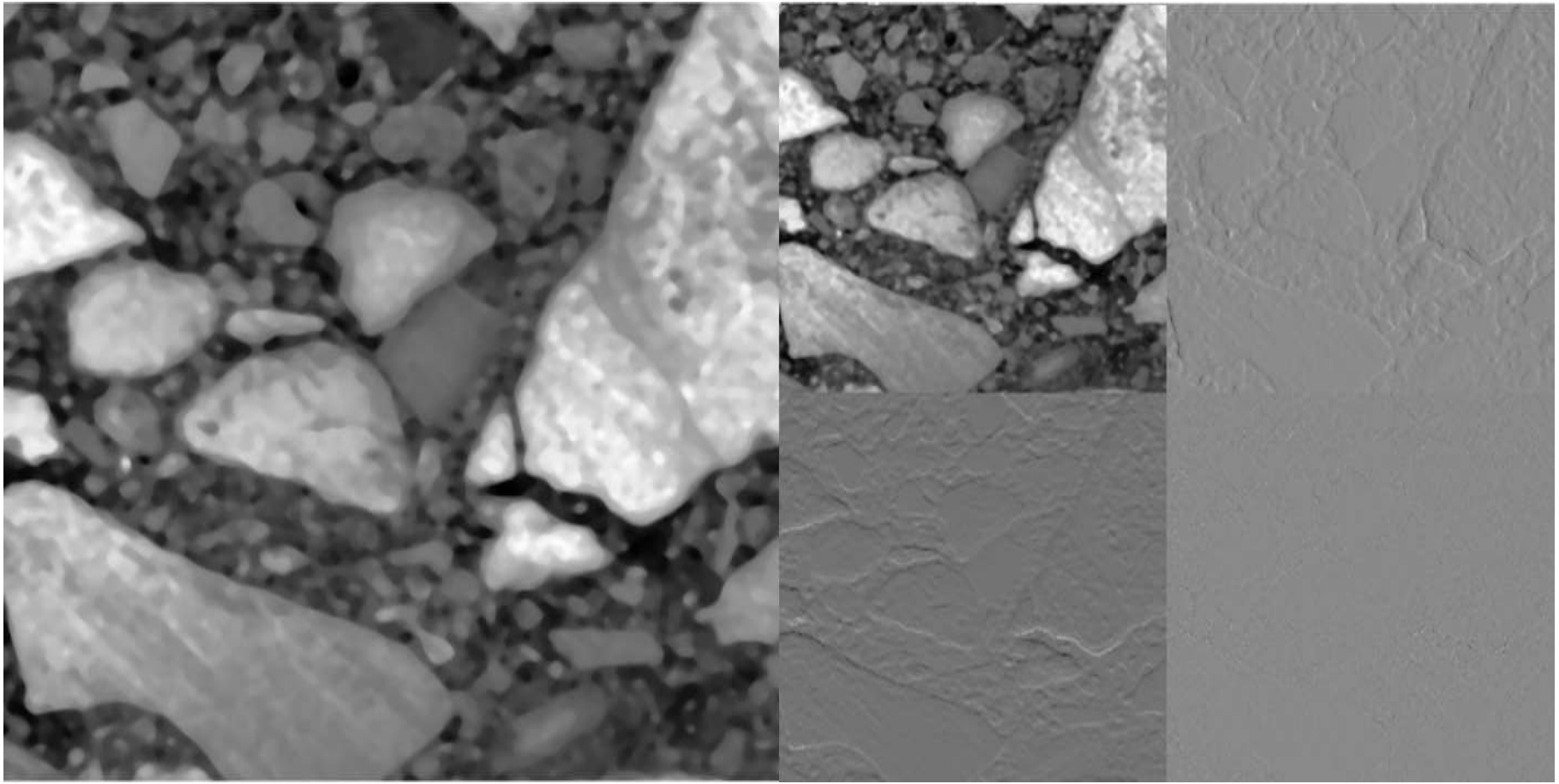
$$\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\right)\begin{bmatrix}a & b\\ c & d\end{bmatrix}\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\right)=\frac{1}{2}\begin{bmatrix}a+b+c+d & a-b+c-d\\ a+b-c-d & a-b-c+d\end{bmatrix}$$

Primero transforma las columnas

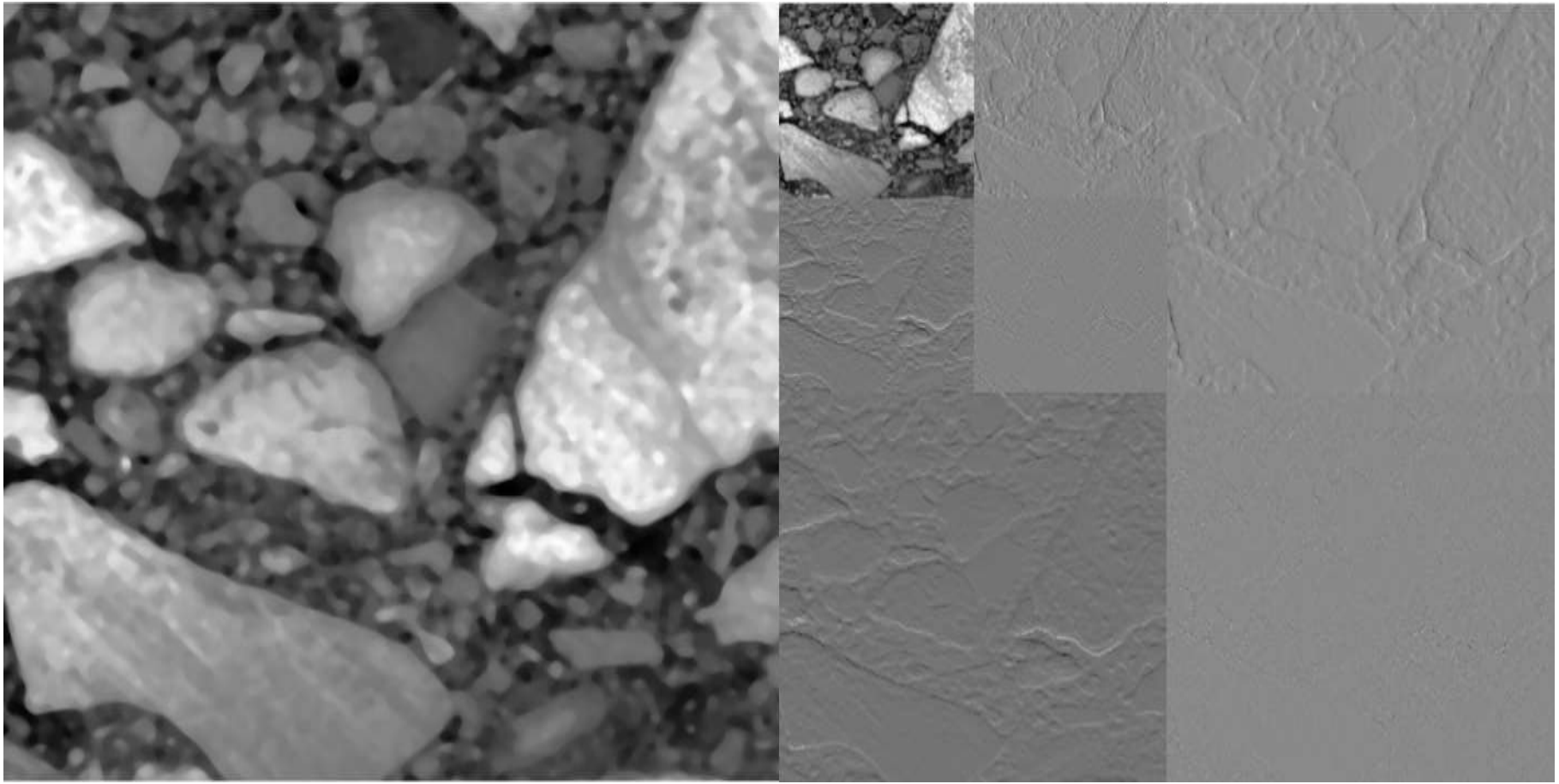
Después transforma las filas

(1,1) : Filtros pasabajos en filas y columnas
(2,1): pasabajos en filas, pasaaltos en columnas
(1,2) : pasabajos en columnas, pasaaltos en filas
(2,2): Filtros pasaaltos en filas y columnas

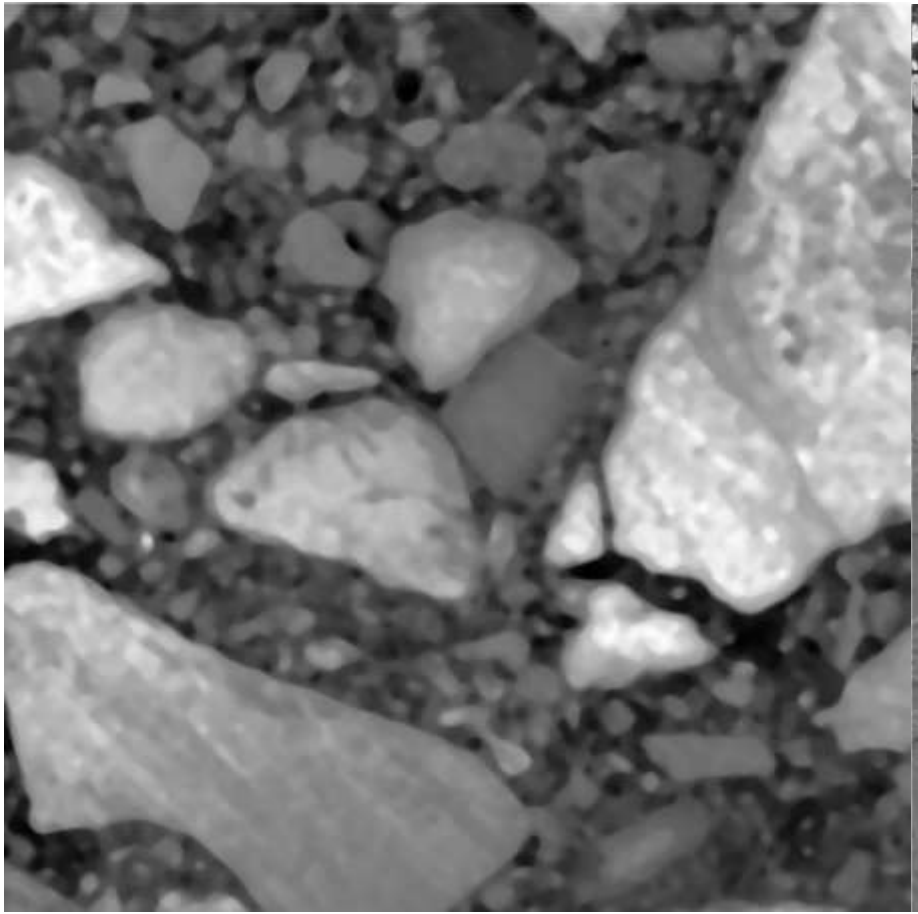
# Segunda característica: Semejanzas y diferencias entre pixeles y grupos de pixeles

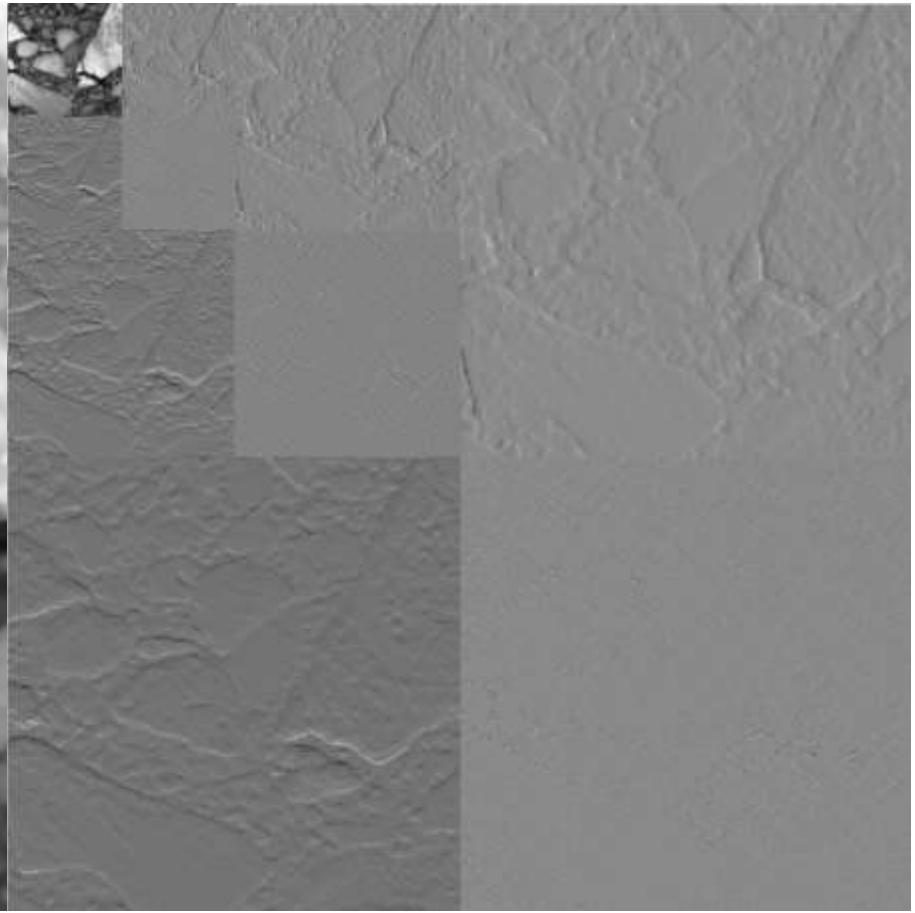


# Segunda característica: Semejanzas y diferencias entre pixeles y grupos de pixeles



# Segunda característica: Semejanzas y diferencias entre pixeles y grupos de pixeles





# Segunda característica: Semejanzas y diferencias entre pixeles y grupos de pixeles

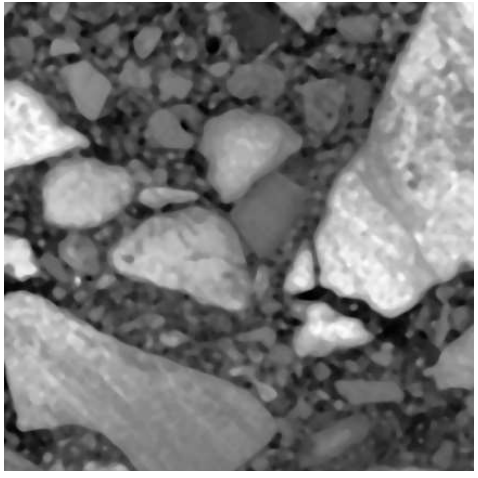
Ahora tenemos un conjunto de *n* coeficientes wavelet para cada pixel $\left\{ d(x,y) \in \mathbb{R}^n, (x,y) \in \mathbb{Z}_{N_F} \times \mathbb{Z}_{N_C} \right\}$

Versión Otsu en *n* dimensiones: Algoritmo de las *k* medias (*k*=2)

$$I_B(x,y) = \begin{cases} 0 & si \quad \left\| d(x,y) - \mu_a \right\| < \left\| d(x,y) - \mu_g \right\| \\ 1 & si \quad \left\| d(x,y) - \mu_a \right\| \geq \left\| d(x,y) - \mu_g \right\| \end{cases} \qquad \forall (x,y) \in \mathbb{Z}_{N_F} \times \mathbb{Z}_{N_C}$$

$$\mu_g = \frac{\sum_{(x,y) \in \mathbb{Z}_{NF} \times \mathbb{Z}_{NC}} I_B(x,y) d(x,y)}{\sum_{(x,y) \in \mathbb{Z}_{NF} \times \mathbb{Z}_{NC}} I_B(x,y)} \quad , \quad \mu_a = \frac{\sum_{(x,y) \in \mathbb{Z}_{NF} \times \mathbb{Z}_{NC}} \left(1 - I_B(x,y)\right) d(x,y)}{\sum_{(x,y) \in \mathbb{Z}_{NF} \times \mathbb{Z}_{NC}} \left(1 - I_B(x,y)\right)}$$

Repetir hasta convergencia

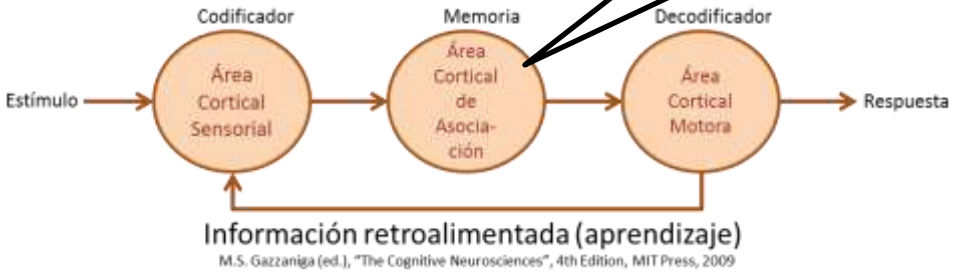


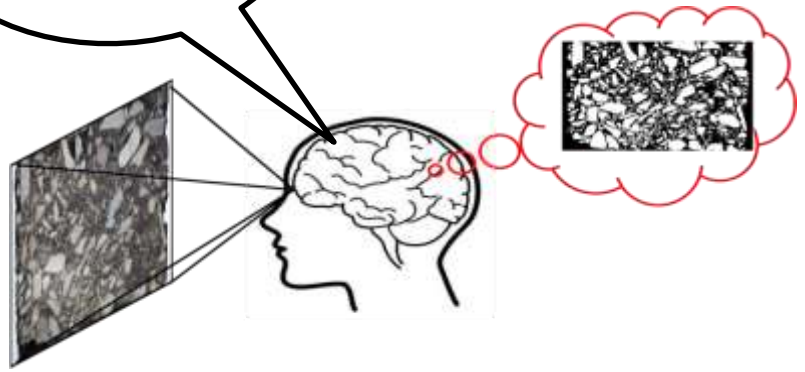
# Segunda característica: Semejanzas y diferencias entre pixeles y grupos de pixeles

Ahora tenemos un conjunto de *n* coeficientes wavelet para cada pixel $\left\{d(x,y) \in \mathbb{R}^n, (x,y) \in \mathbb{Z}_{N_F} \times \mathbb{Z}_{N_C}\right\}$

¿Posiblemente en el área cortical de asociación haya un modelo matemático al respecto?…

$$f\left(d(x,y)\middle|\mu_a,\Sigma_a,\mu_g,\Sigma_g,a_g\right) =$$
$$a_g f_g\left(d(x,y)\middle|\mu_g,\Sigma_g\right) + (1-a_g) f_a\left(d(x,y)\middle|\mu_a,\Sigma_a\right)$$



$$f\left(d(x,y)\middle|\mu_a,\Sigma_a,\mu_g,\Sigma_g,a_g\right)=a_g\,f_g\left(d(x,y)\middle|\mu_g,\Sigma_g\right)+(1-a_g)\,f_a\left(d(x,y)\middle|\mu_a,\Sigma_a\right)$$

$$0\le a_g\le 1$$

$$f_g\left(d(x,y)\middle|\mu_g,\Sigma_g\right)=\frac{1}{\sqrt{(2\pi)^N\left|\Sigma_g\right|}}\exp\left(-\frac{1}{2}\left(d(x,y)-\mu_g\right)^T\Sigma_g^{-1}\left(d(x,y)-\mu_g\right)\right)$$

$$f_a\left(d(x,y)\middle|\mu_a,\Sigma_a\right)=\frac{1}{\sqrt{(2\pi)^N\left|\Sigma_a\right|}}\exp\left(-\frac{1}{2}\left(d(x,y)-\mu_a\right)^T\Sigma_a^{-1}\left(d(x,y)-\mu_a\right)\right)$$

Mezcla de Gausianas

$$a_g=\frac{\sum_{(x,y)}I_B(x,y)}{N_P}$$

$$\mu_g=\frac{1}{N_P\cdot a_g}\sum_{(x,y)}I_B(x,y)d(x,y)$$

$$\mu_a=\frac{1}{NP\cdot(1-a_g)}\sum_{(x,y)}\left(1-I_B(x,y)\right)d(x,y)$$

$$\Sigma_g=\frac{1}{N_P\cdot a_g}\sum_{(x,y)}\left(d(x,y)-\mu_g\right)\cdot\left(d(x,y)-\mu_g\right)^T\cdot I_B(x,y)$$

$$\Sigma_a=\frac{1}{N_P\cdot(1-a_g)}\sum_{(x,y)}\left(d(x,y)-\mu_a\right)\cdot\left(d(x,y)-\mu_a\right)^T\left(1-I_B(x,y)\right)$$

Inicializa el algoritmo Expectation/Maximization

Expectation:

$$P\left(G\middle|d(x,y)\right)=\frac{a_g f_g\left(d(x,y)\middle|\mu_g,\Sigma_g\right)}{a_g f_g\left(d(x,y)\middle|\mu_g,\Sigma_g\right)+(1-a_g) f_a\left(d(x,y)\middle|\mu_a,\Sigma_a\right)}$$

Maximization:

$$a_g=\frac{1}{NP}\sum_{(x,y)}P\left(G\middle|d(x,y)\right)$$

$$\mu_g=\frac{1}{NP\cdot a_g}\sum_{(x,y)}d(x,y)P\left(G\middle|d(x,y)\right)$$

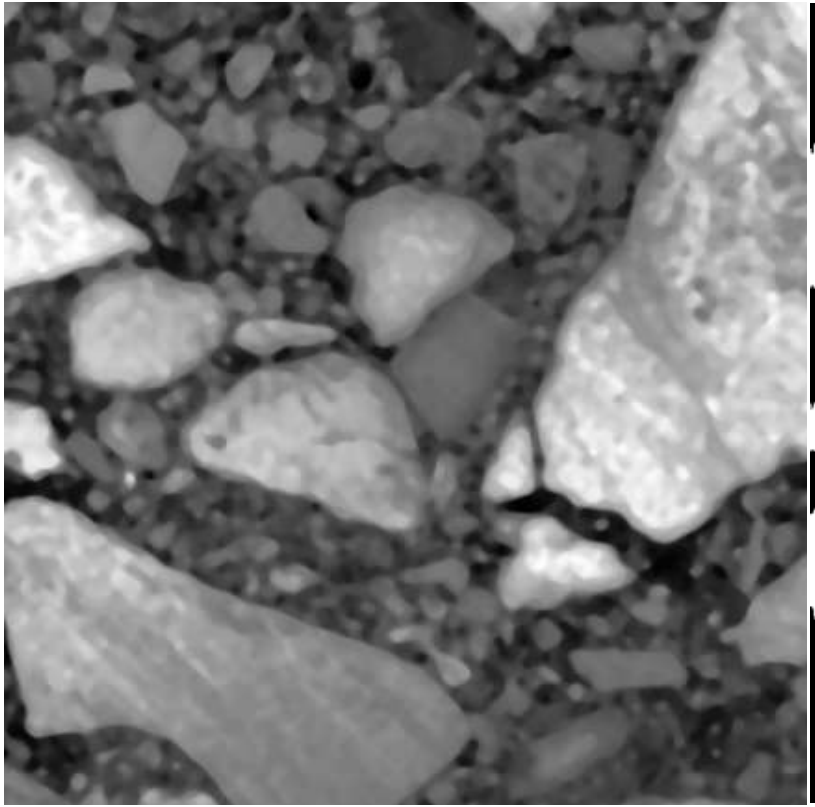
$$\mu_a=\frac{1}{NP\cdot(1-a_g)}\sum_{(x,y)}d(x,y)\left(1-P\left(G\middle|d(x,y)\right)\right)$$

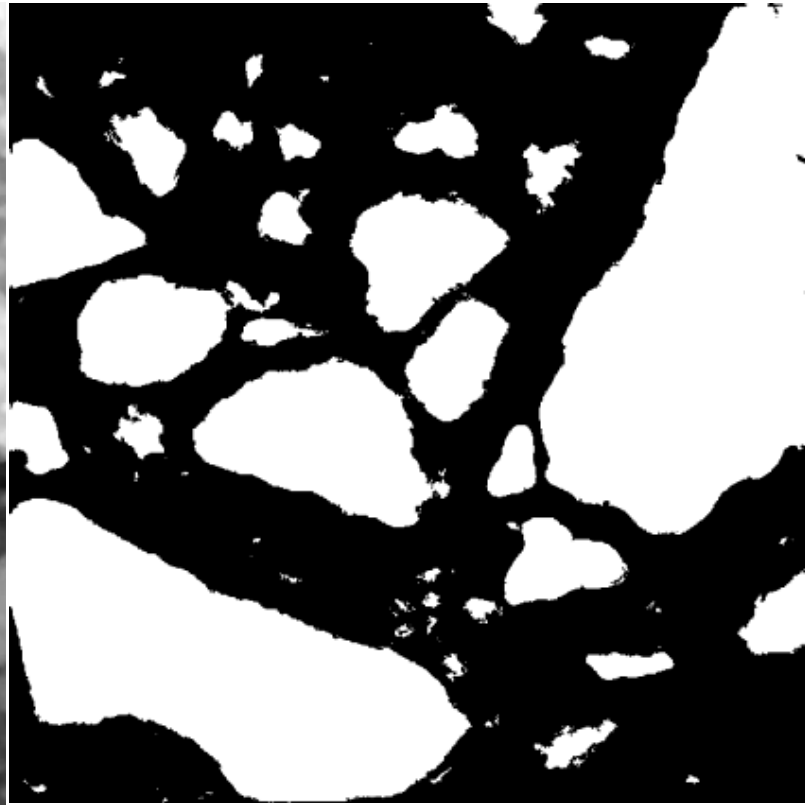
$$\Sigma_g=\frac{1}{NP\cdot a_g}\sum_{(x,y)}\left(d(x,y)-\mu_g\right)\cdot\left(d(x,y)-\mu_g\right)^T P\left(G\middle|d(x,y)\right)$$

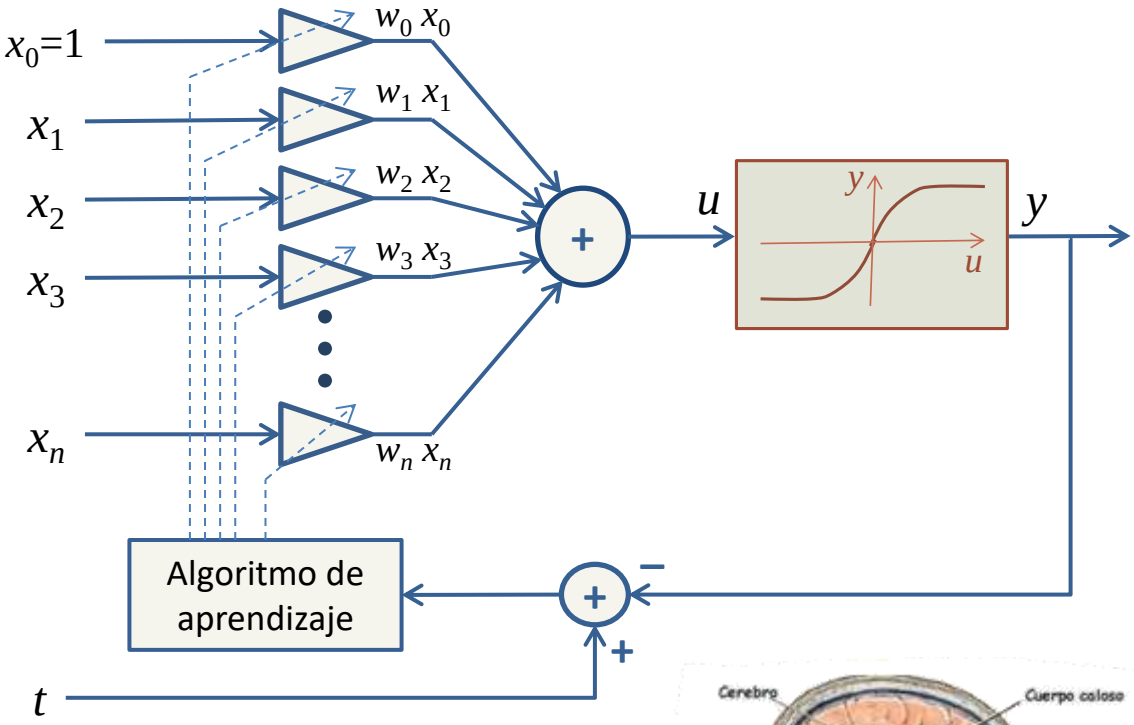
$$\Sigma_a=\frac{1}{NP\cdot(1-a_g)}\sum_{(x,y)}\left(d(x,y)-\mu_a\right)\cdot\left(d(x,y)-\mu_a\right)^T\left(1-P\left(G\middle|d(x,y)\right)\right)$$

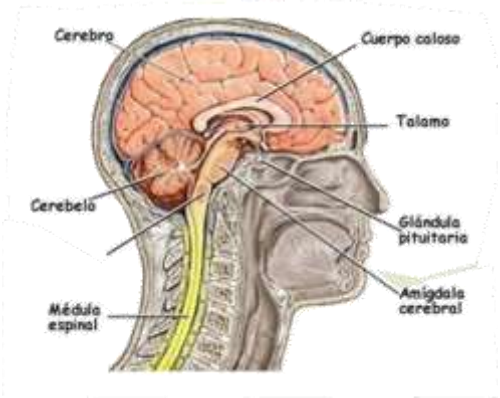
Repetir hasta convergencia

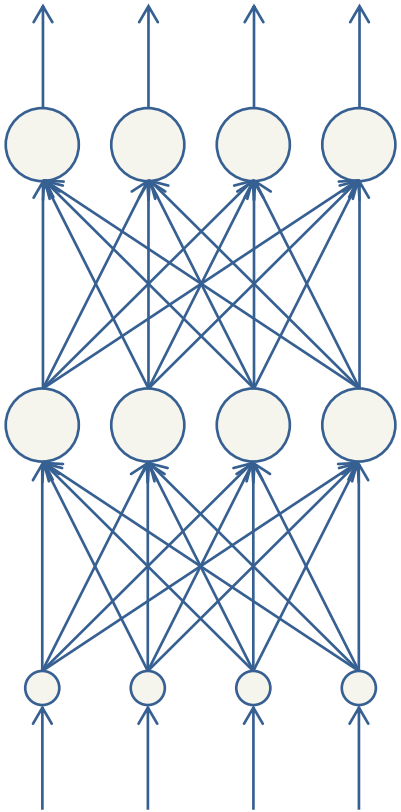
$$I_B(x,y)=\begin{cases}0 & si \quad P\left(G\middle|d(x,y)\right)<1/2\\ 1 & si \quad P\left(G\middle|d(x,y)\right)\geq 1/2\end{cases}$$

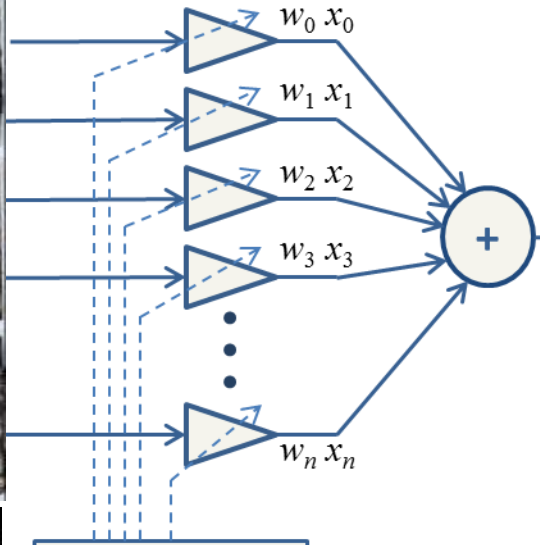


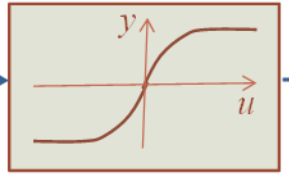


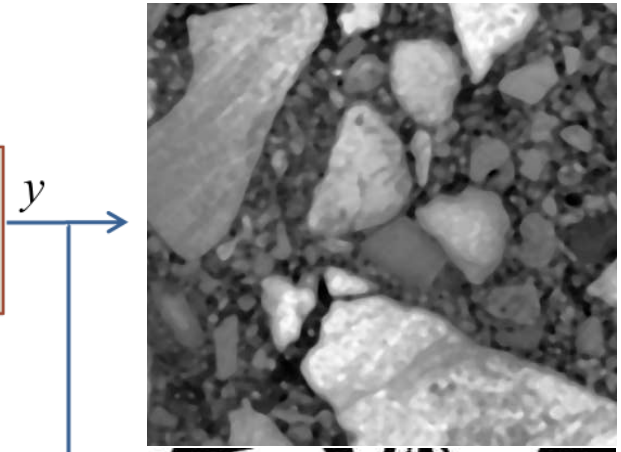

$x_0$=1
$x_1$
$x_2$
$x_3$
$x_n$
$w_0\ x_0$
$w_1\ x_1$
$w_2\ x_2$
$w_3\ x_3$
$w_n\ x_n$
+
$u$
y
u
$y$
Algoritmo de aprendizaje
+
−
+
$t$

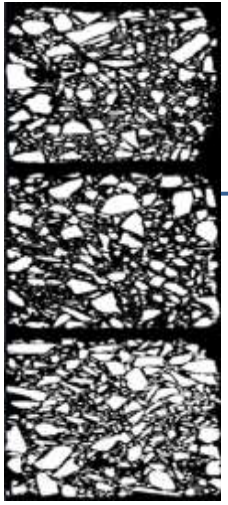

Cerebro
Cuerpo calloso
Tálamo
Cerebelo
Glándula pituitaria
Amígdala cerebral
Médula espinal

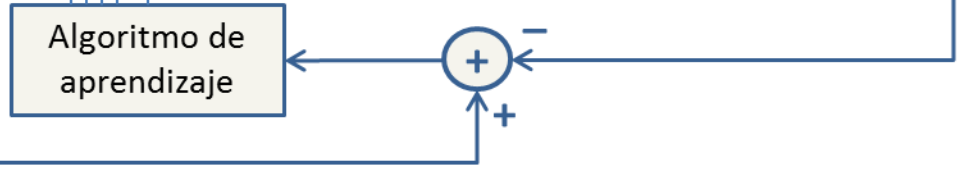


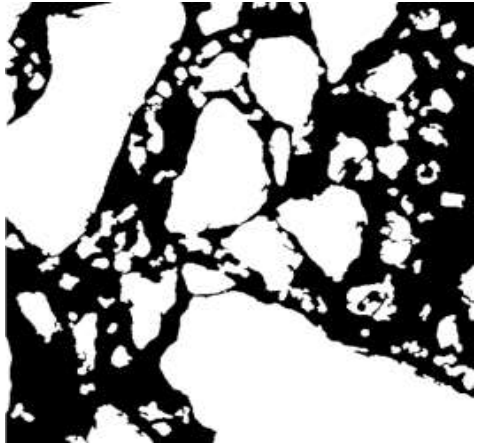

$w_0\ x_0$
$w_1\ x_1$
$w_2\ x_2$
$w_3\ x_3$
$w_n\ x_n$
+


u
y
u

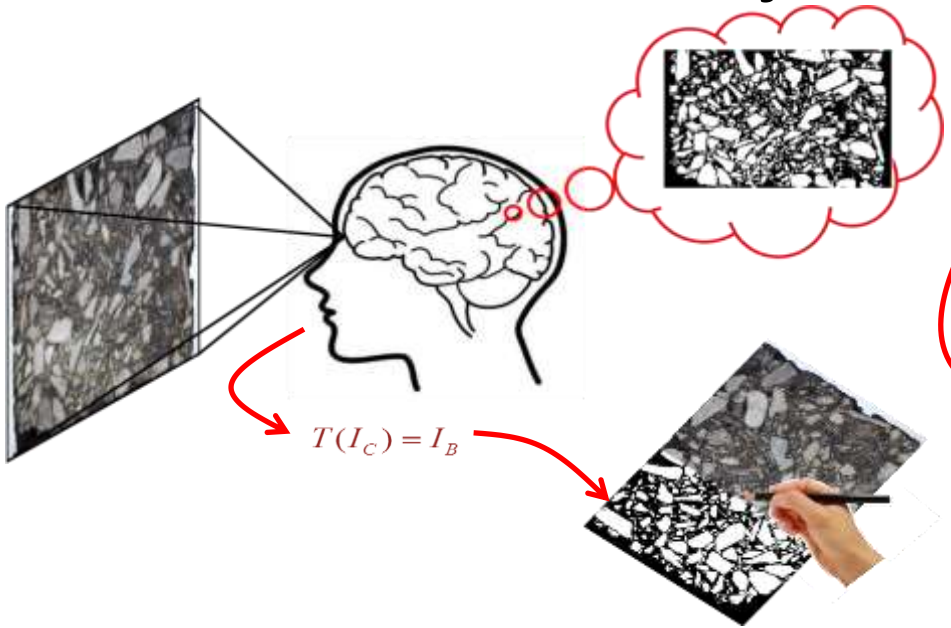

y

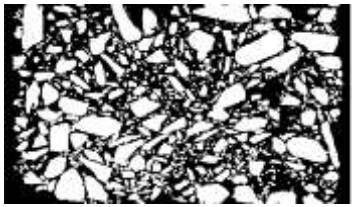



Algoritmo de aprendizaje
+
−
+

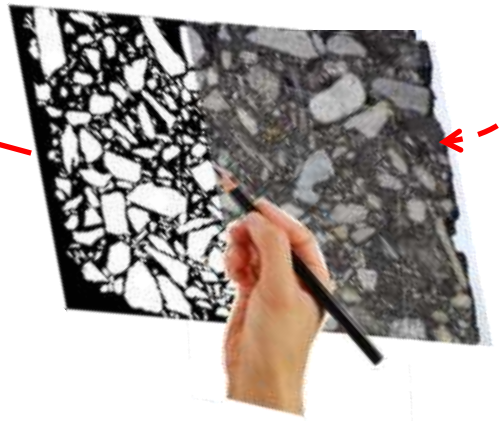


# ¿Una inteligencia centralizada haciendo el trabajo?

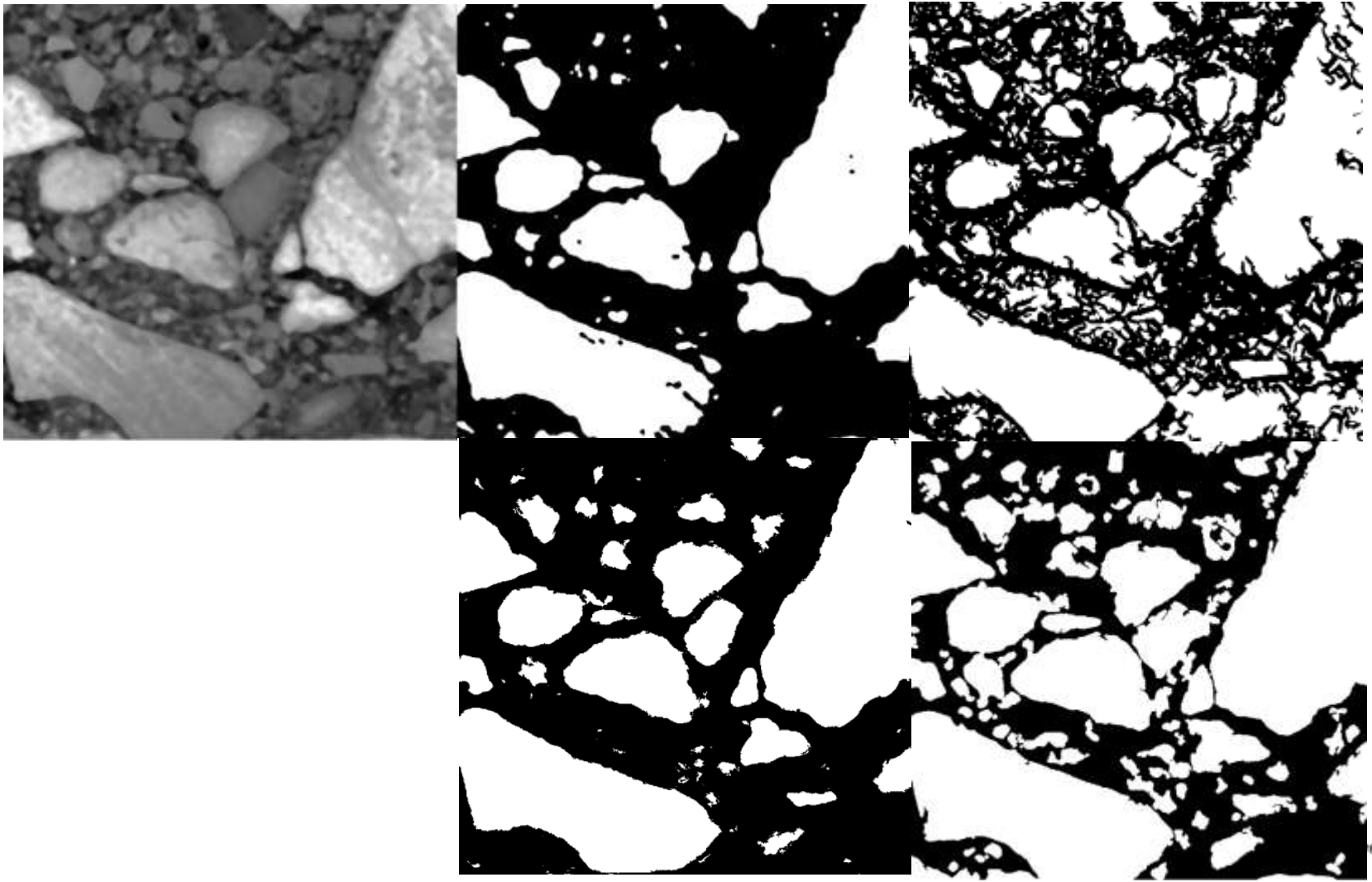


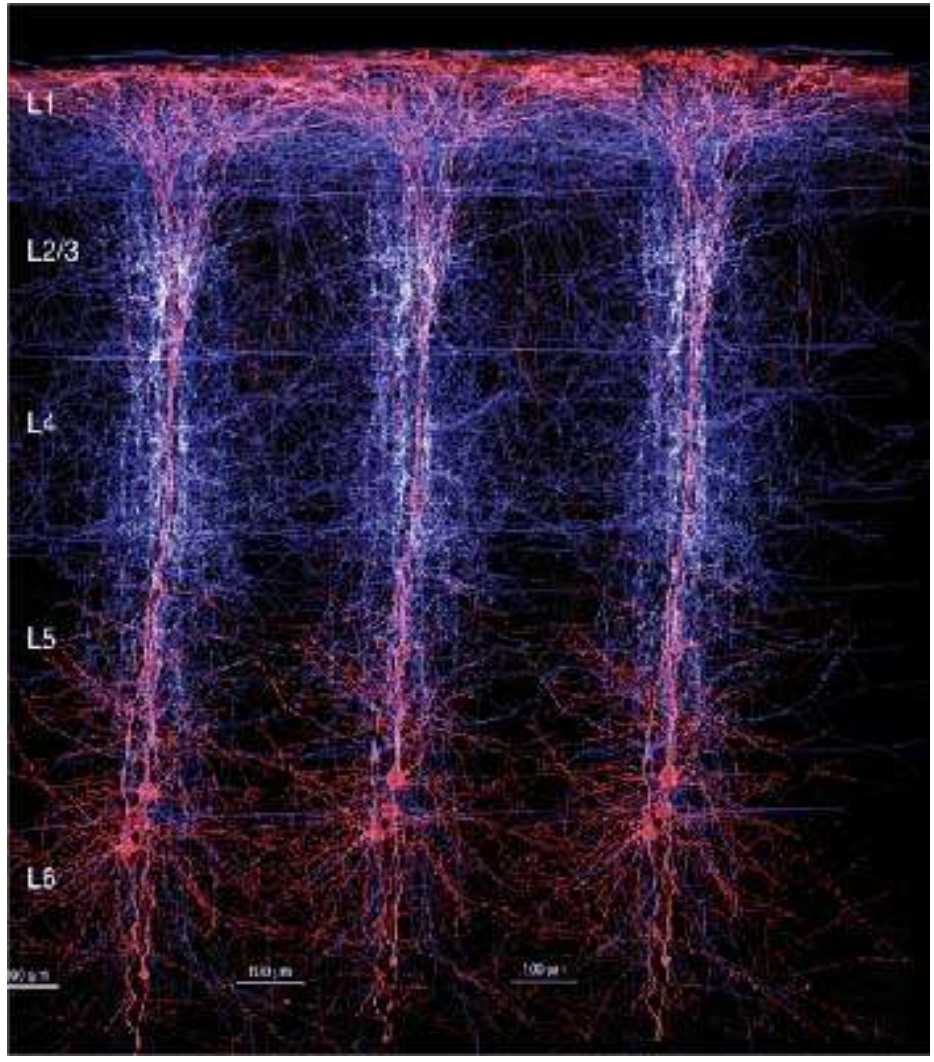



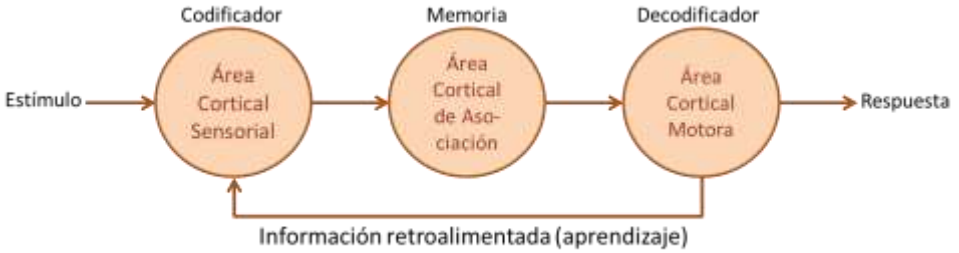

S. Haykin, "Cognitive Dynamic Systems", Cambridge University Press, 2012



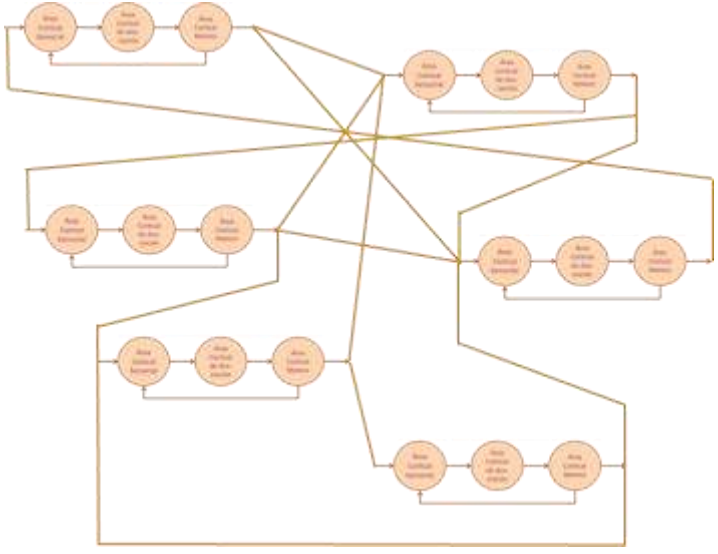
¡Aprendizaje!

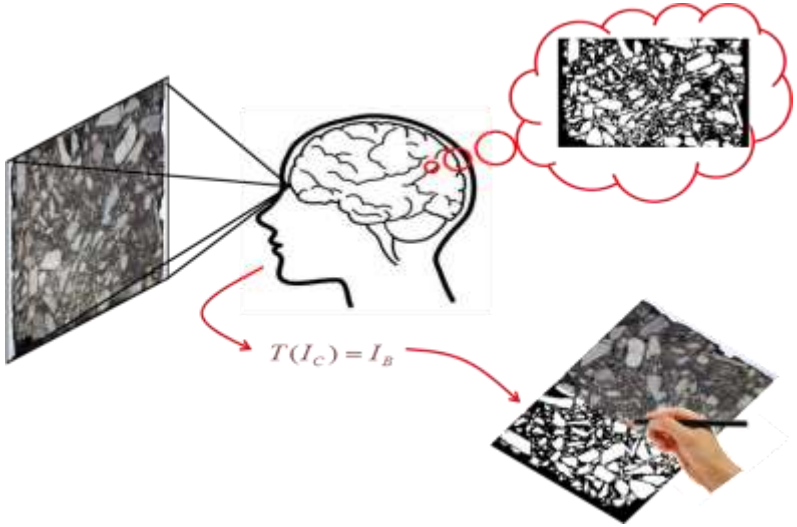


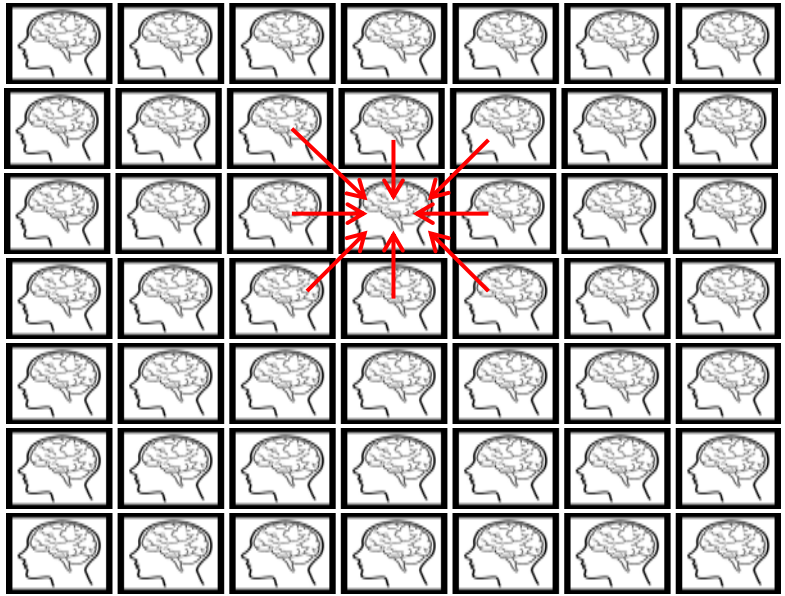


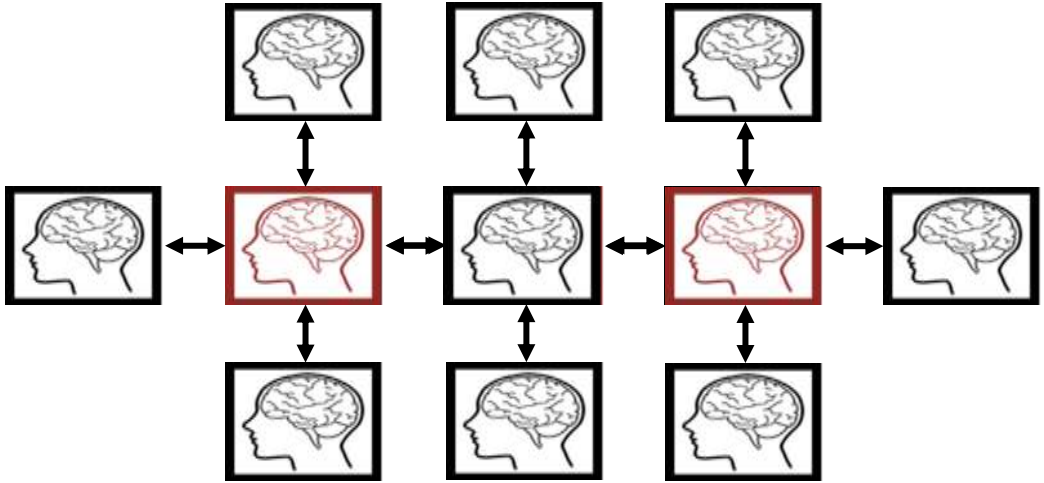

M.S. Gazzaniga (ed.), "The Cognitive Neurosciences", 4th Edition, MIT Press, 2009

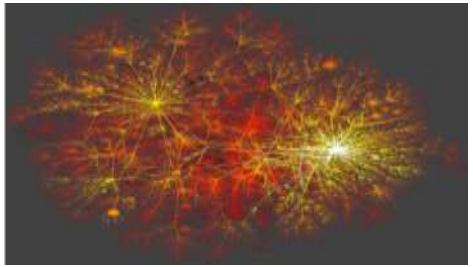


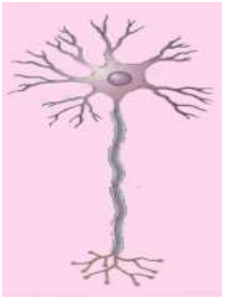


**¡Cada pixel es un sistema cognitivo!**

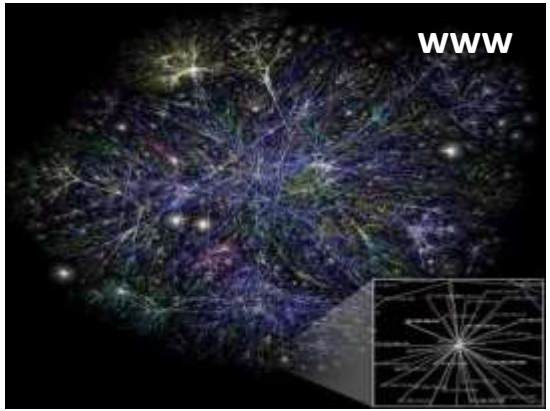


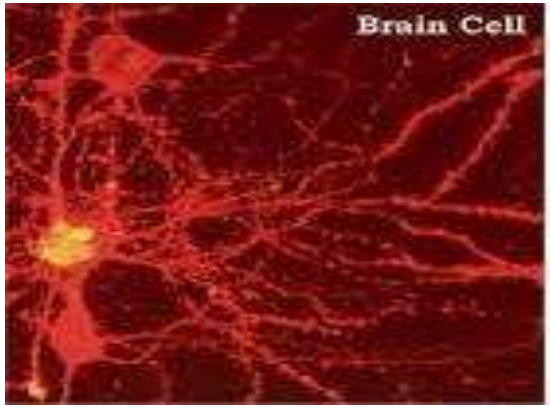




# Sistemas complejos

# Aspectos comunes de los sistemas complejos

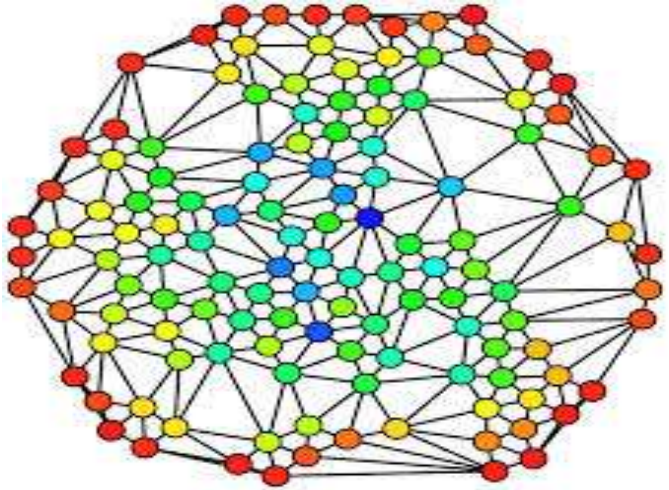




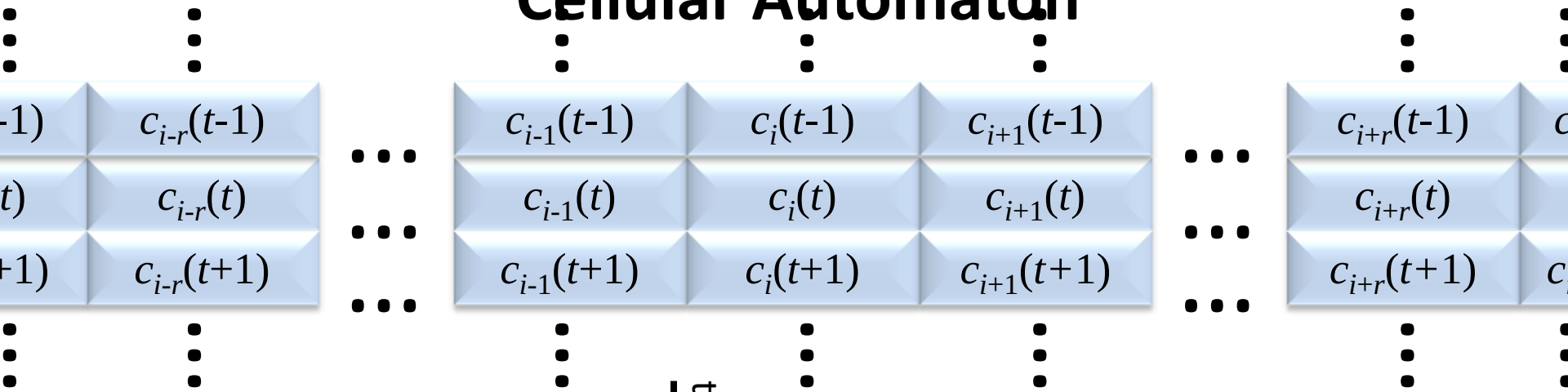
Large networks of individual components
Simple rules with no central control
Collective actions of many components  give rise to complex patterns
Production and utilization of information
Adaptation through learning and evolution

# Natural math model



- Local transmission of electrochemical impulses among neighbor neurons
- Local plasmid migration among neighbor bacteria
- Local social behavior learning among neighbor individuals (gossip, epidemics,…)
- Local chemical reactions among neighbor molecules
- Local packet exchanges among neighbor routers
- Etc.

# Cellular Automaton




$c_i(t)$ is the state of cell $i$ at instant $t$

The next state of cell $i$ depends on the current states of the neighbor cells in a radius $r$

For $k$ states we need $R = k^{2r+1}$ rules and there are $k^R$ possible sets of rules

For example : Number of states = 2

States = {0,1}

radius = 1

One of 256 possible sets of rules

$2^3$=8 rules : if $1 \leq \Sigma_{k=-1,0,1} c_{i+k}(t) \leq 2$, $c_i(t+1) \leftarrow 1$, else $c_i(t+1) \leftarrow 0$

# Finite State Automaton

**If**

$1 \leq \Sigma_{k=-1,0,1} c_{i+k}(t) \leq 2$

**then**

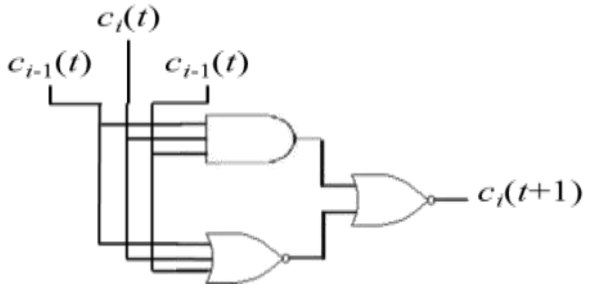
$c_i(t+1) \leftarrow 1$

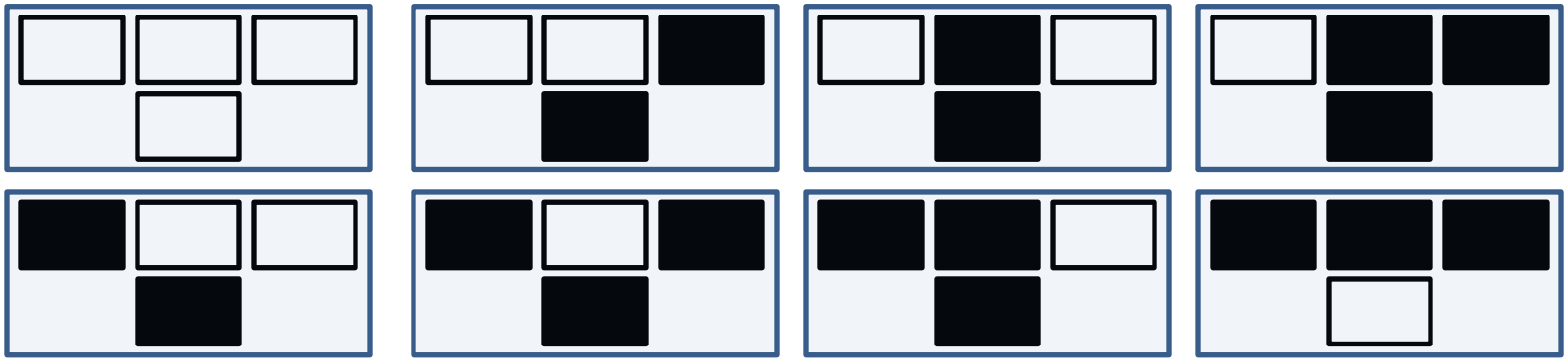
**else**

$c_i(t+1) \leftarrow 0$

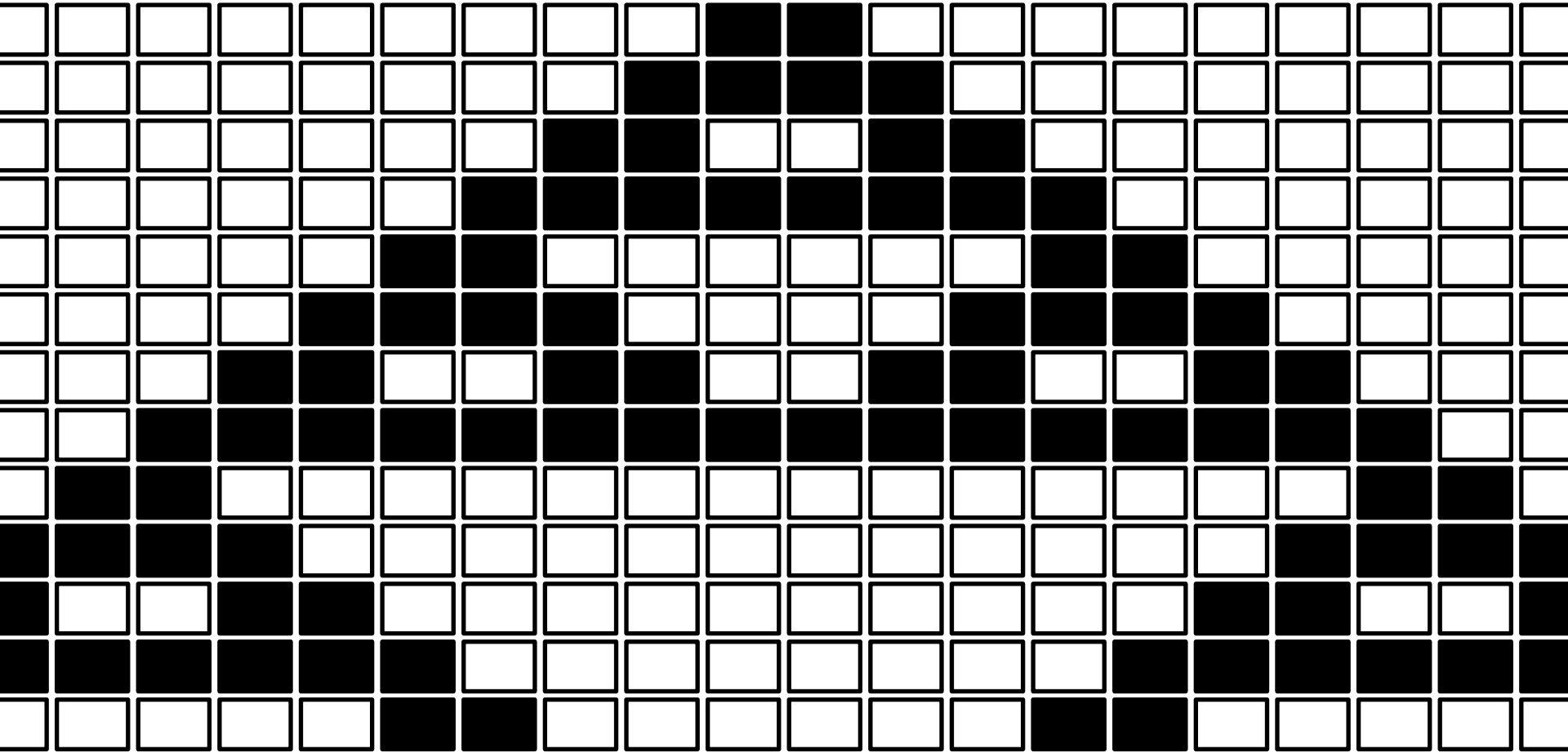
**end**

| $c_{i-1}(t)$ | $c_i(t)$ | $c_{i+1}(t)$ | $c_i(t+1)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

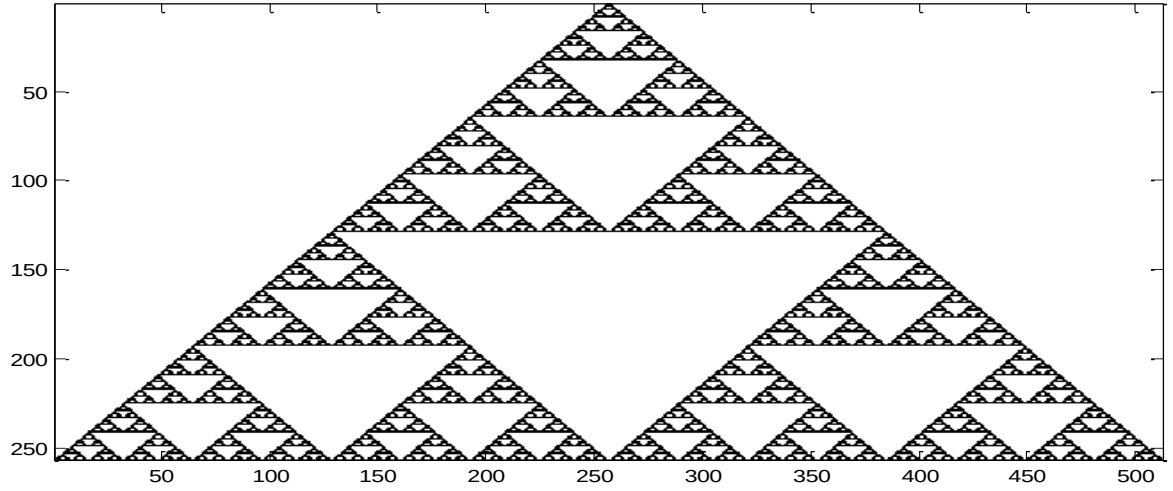


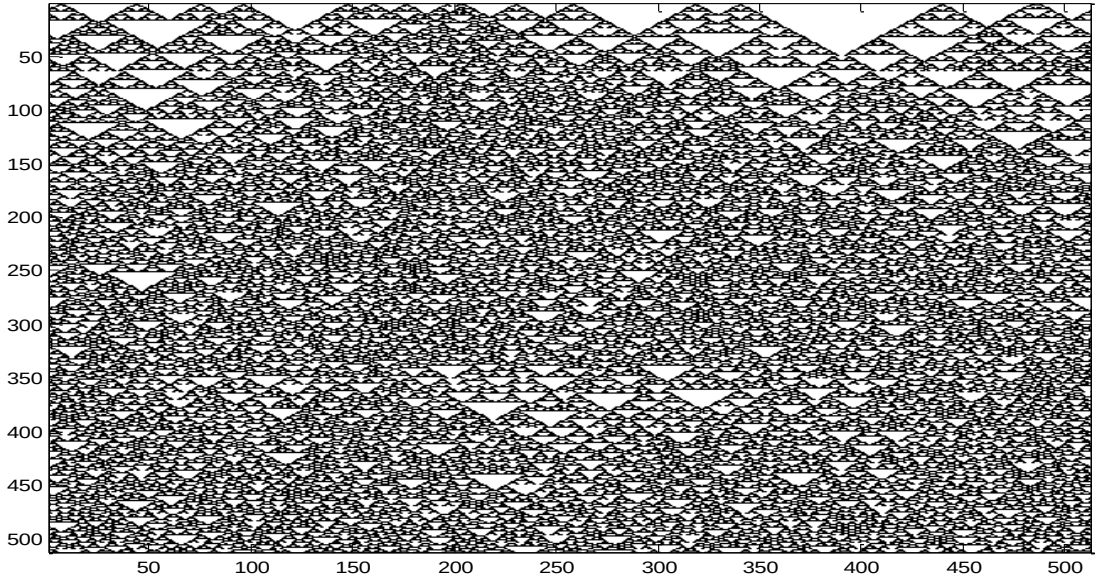
# Finite State Automaton



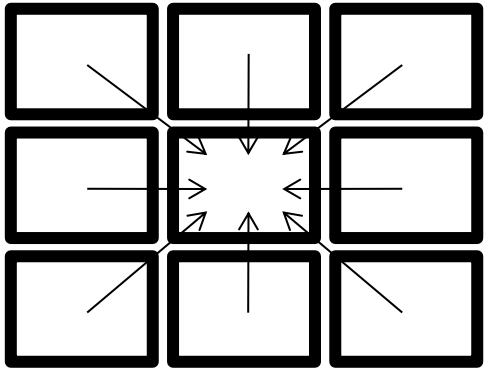
```
colormap([1 1 1; 0 0 0])    % 0 blanco, 1 negro
n = 512; i=2:n-1;           % Tamaño de la línea
c = zeros(n/2,n);           % Historia del autómata
c(1,n/2 + [0 1]) = 1;       % Estado inicial
for t=2:n/2                 % Indice de tiempo
    sum(i) = c(t-1,i-1) + c(t-1,i) + c(t-1,i+1);
    sum(1) = c(t-1,n) + c(t-1,1) + c(t-1,2);
    sum(n) = c(t-1,1) + c(t-1,n) + c(t-1,n-1);
    c(t,:) = (sum<3).*(sum>0); % Evalúa la regla
end
imagesc(c)
```



```
colormap([1 1 1; 0 0 0])    % 0 blanco, 1 negro
n = 512; i=2:n-1;           % Tamaño de la linea
c = zeros(n,n);             % Historia del autómata
c(1,:) = (rand(1,n)>0.95); % Estado inicial
for t=2:n                   % Indice de tiempo
    sum(i) = c(t-1,i-1) + c(t-1,i) + c(t-1,i+1);
    sum(1) = c(t-1,n) + c(t-1,1) + c(t-1,2);
    sum(n) = c(t-1,1) + c(t-1,n) + c(t-1,n-1);
    c(t,:) = (sum<3).*(sum>0); % Evalúa la regla
end
imagesc(c)
```



# Conway's Game of life



1. Any live cell with fewer than two live neighbors dies (under-population).
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies (overcrowding).
4. Any dead cell with exactly three live neighbors becomes alive (reproduction).

The initial pattern constitutes the *seed* of the system.

```
n=64 ;                                  % Tamaño del mundo
mundo = (rand(n,n)<0.5);                % Inicializa el mundo
imagesc(mundo);                         % Visualiza el mundo
colormap([1 1 1; 0 0 0]);               % 0 - blanco, 1 - negro
axis equal
axis tight
while(1)
    mundo = [mundo(n,n) mundo(n,1:n) mundo(n,1); ... % Forma el toroide
             mundo(1:n,n) mundo mundo(1:n,1); ...
             mundo(1,n) mundo(1,1:n) mundo(1,1)];
    % Evalúa la regla de interacción
    suma  =  mundo(1:n,1:n)   + mundo(1:n,2:n+1)   + mundo(1:n,3:n+2) + ...
             mundo(2:n+1,1:n) +                      mundo(2:n+1,3:n+2) + ...
             mundo(3:n+2,1:n) + mundo(3:n+2,2:n+1) + mundo(3:n+2,3:n+2);
    mundo = ((suma==3) + (suma==2).*mundo(2:n+1,2:n+1));
    imagesc(mundo);                     % Actualiza la imagen
    drawnow
end
```

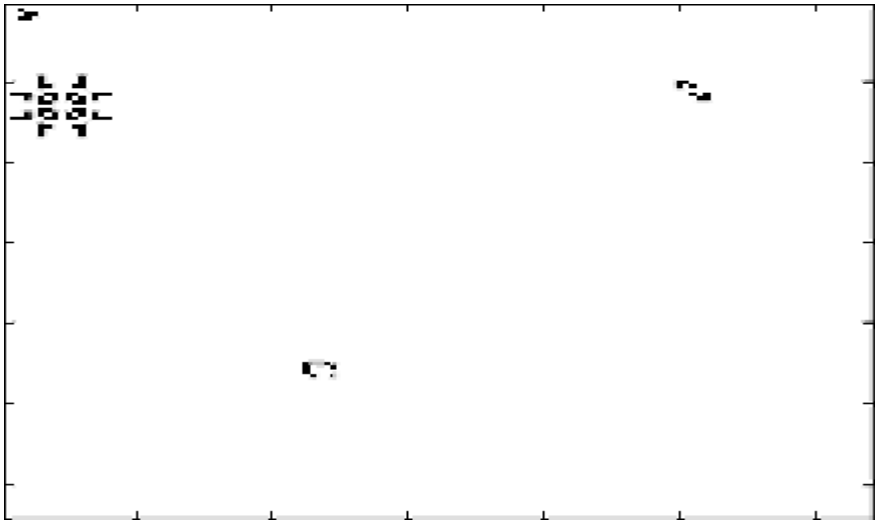
# Game of life

```
n=64 ;                                  % Tamaño del mundo
mundo = (rand(n,n)<0.5);                % Inicializa el mundo
imagesc(mundo);                         % Visualiza el mundo
colormap([1 1 1; 0 0 0]);               % 0 - blanco, 1 - negro
axis equal
axis tight
while(1)
    mundo = [mundo(n,n) mundo(n,1:n) mundo(n,1); ... % Forma el toroide
             mundo(1:n,n) mundo mundo(1:n,1); ...
             mundo(1,n) mundo(1,1:n) mundo(1,1)];
    % Evalúa la regla de interacción
    suma  =  mundo(1:n,1:n)   + mundo(1:n,2:n+1)   + mundo(1:n,3:n+2) + ...
             mundo(2:n+1,1:n) +                      mundo(2:n+1,3:n+2) + ...
             mundo(3:n+2,1:n) + mundo(3:n+2,2:n+1) + mundo(3:n+2,3:n+2);
    mundo = ((suma==3) + (suma==2).*mundo(2:n+1,2:n+1));
    imagesc(mundo);                     % Actualiza la imagen
    drawnow
end
```

# Game of life

# Visualizing periodic and moving objects

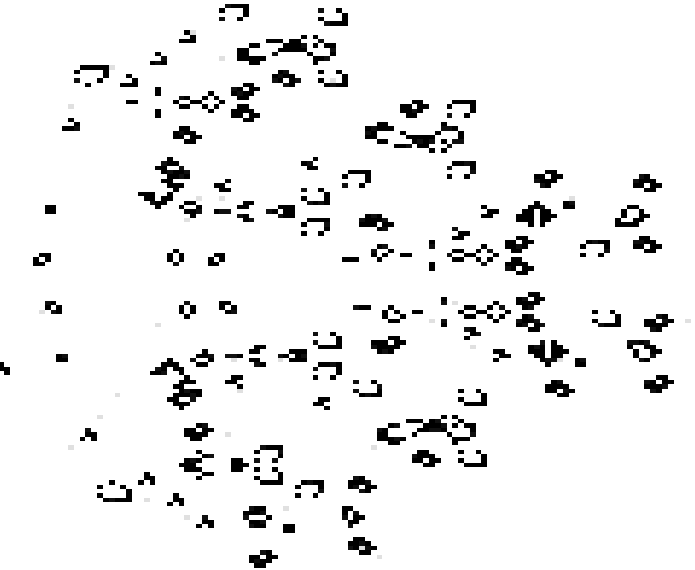
```
n = 128;
mundo = zeros(n,n);
mundo(2:4,2:4) = [0 0 1; 1 0 1; 0 1 1];
mundo(20:32,3:15) = [0 0 1 1 1 0 0 0 1 1 1 0 0;...
                     0 0 0 0 0 0 0 0 0 0 0 0 0;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     0 0 1 1 1 0 0 0 1 1 1 0 0;...
                     0 0 0 0 0 0 0 0 0 0 0 0 0;...
                     0 0 1 1 1 0 0 0 1 1 1 0 0;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     0 0 0 0 0 0 0 0 0 0 0 0 0;...
                     0 0 1 1 1 0 0 0 1 1 1 0 0];

mundo(90:93,45:49) = [0 1 1 0 0;...
                      1 1 0 1 1;...
                      0 1 1 1 1;...
                      0 0 1 1 0];

mundo(20:24,100:104) = [1 1 0 0 0;...
                        1 0 0 0 0;...
                        0 1 0 1 0;...
                        0 0 0 0 1;...
                        0 0 0 1 1];
```

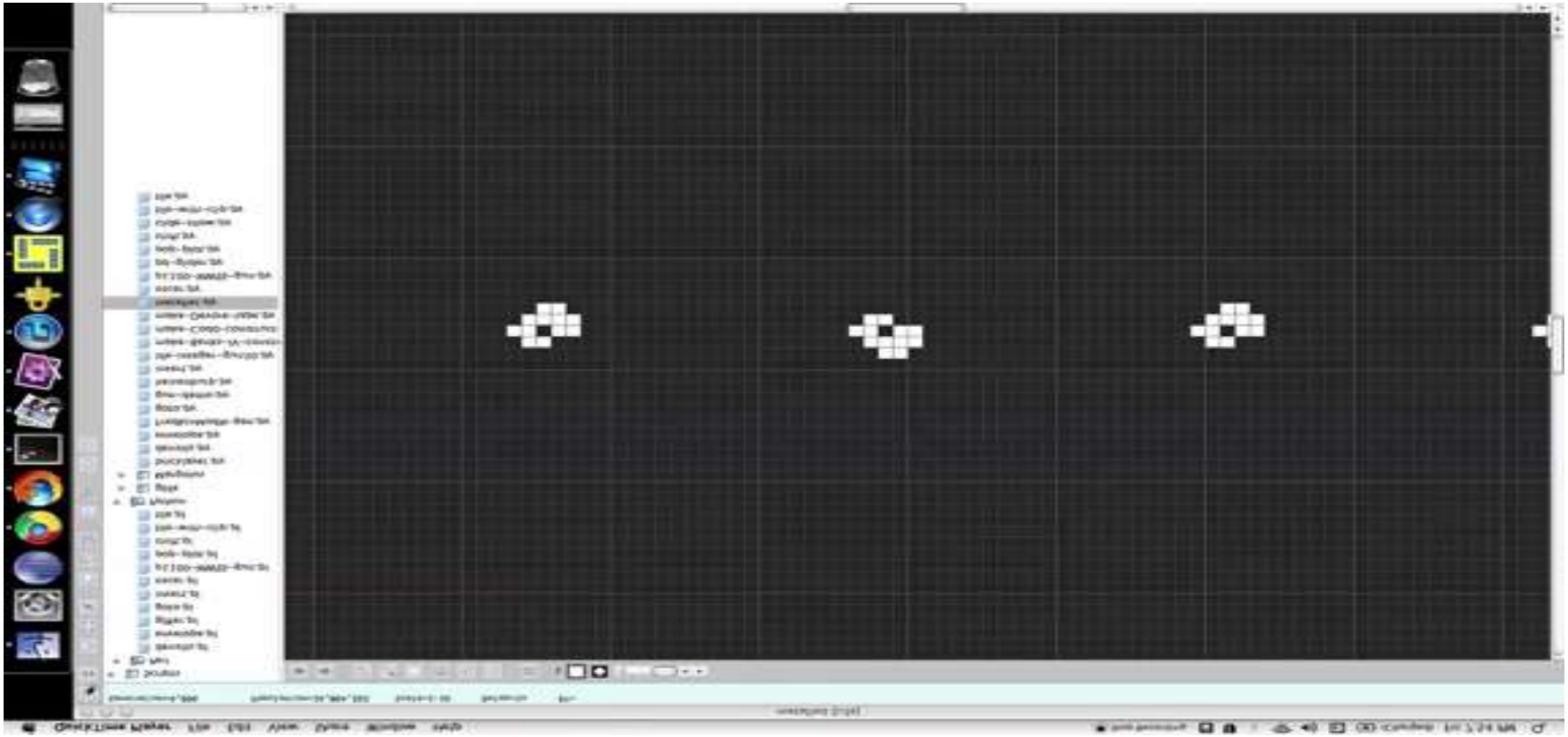


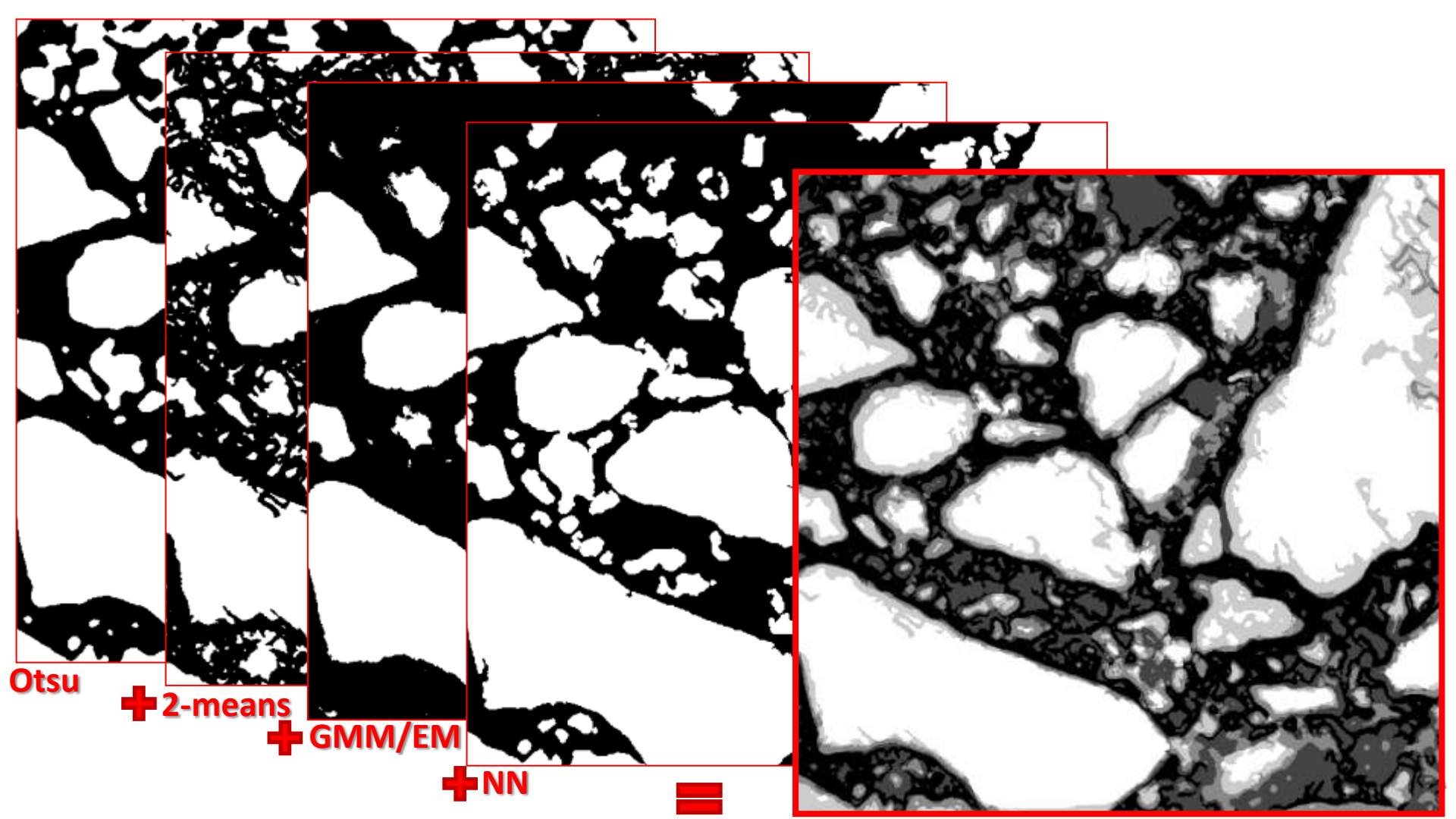
```
mundo = zeros(n,n);                    % Inicializa el mundo
mundo(2:10,2:37) = [0 0 0 0 1 1 0 0 0; ...
                    0 0 0 0 1 1 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 1 1 1 0 0; ...
                    0 0 0 1 0 0 0 1 0; ...
                    0 0 1 0 0 0 0 0 1; ...
                    0 0 1 0 0 0 0 0 1; ...
                    0 0 0 0 0 1 0 0 0; ...
                    0 0 0 1 0 0 0 1 0; ...
                    0 0 0 0 1 1 1 0 0; ...
                    0 0 0 0 0 1 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 1 1 1 0 0 0 0; ...
                    0 0 1 1 1 0 0 0 0; ...
                    0 1 0 0 0 1 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    1 1 0 0 0 1 1 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 1 1 0 0 0 0 0; ...
                    0 0 1 1 0 0 0 0 0];';
```

# Game of life:Glider gun

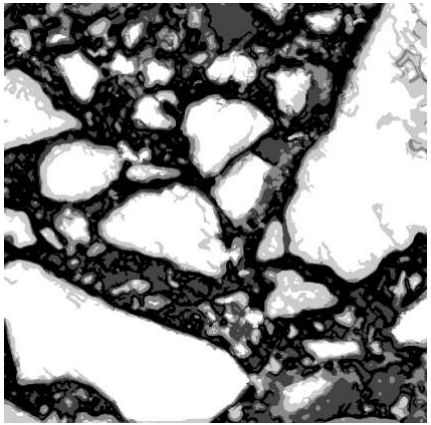
# Why “Game of life”?
# … Self-reproduction!



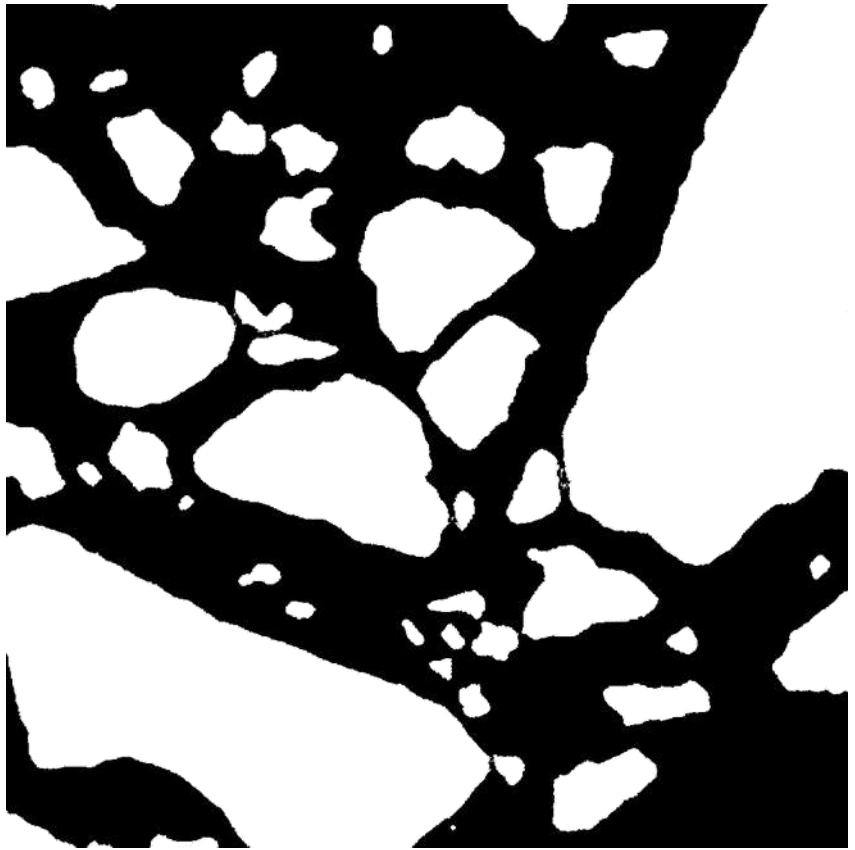


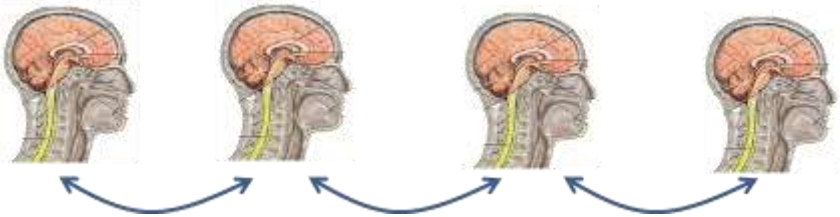

Otsu
2-means
GMM/EM
NN

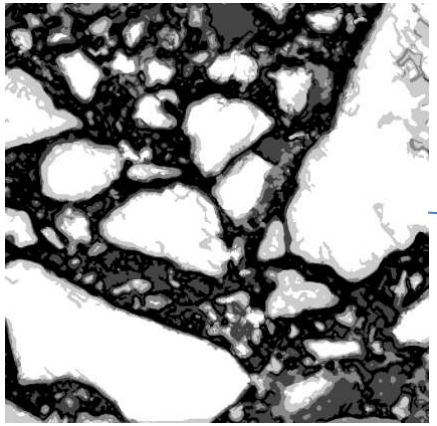
Aumente la clase del pixel si es menor que el promedio de las clases del vecindario y disminúyala si es mayor que el promedio de las clases del vecindario (hay 5 clases y las clases 0 y 4 no se alteran)

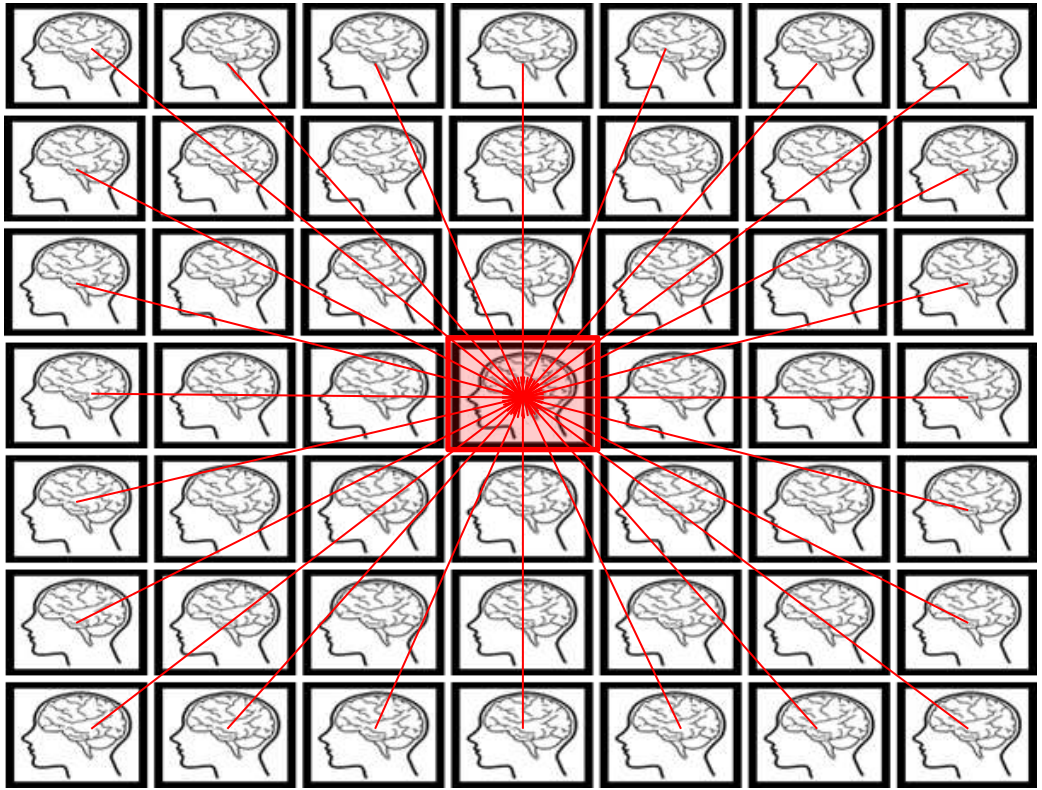


Los pixeles dudosos se ajustan a la mayoría de los vecinos. Si la mayoría son dudosos, pone la clase mayoritaria entre asfalto y gravilla









Si dispongo 49 muestras de las características de mis vecinos, puedo construir un modelo cognitivo tipo GMM/EM para determinar mi probabilidad de ser gravilla.

Expectation:

$$P\left(G\middle|d(x,y)\right)=\frac{a_g f_g\left(d(x,y)\middle|\mu_g,\Sigma_g\right)}{a_g f_g\left(d(x,y)\middle|\mu_g,\Sigma_g\right)+(1-a_g) f_a\left(d(x,y)\middle|\mu_a,\Sigma_a\right)}$$

Maximization:

Repetir hasta convergencia

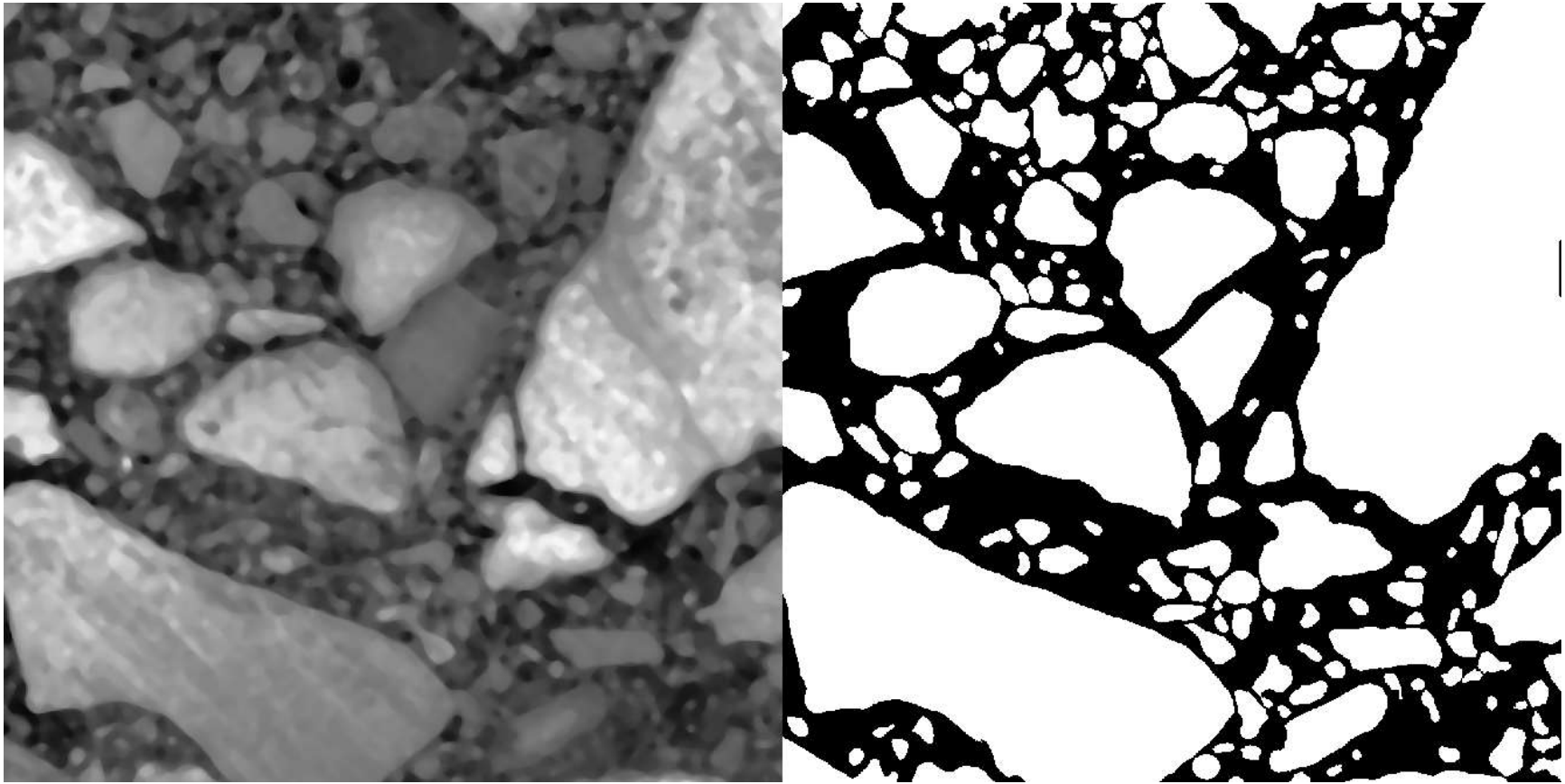
$$a_g=\frac{1}{NP}\sum_{(x,y)}P\left(G\middle|d(x,y)\right)$$

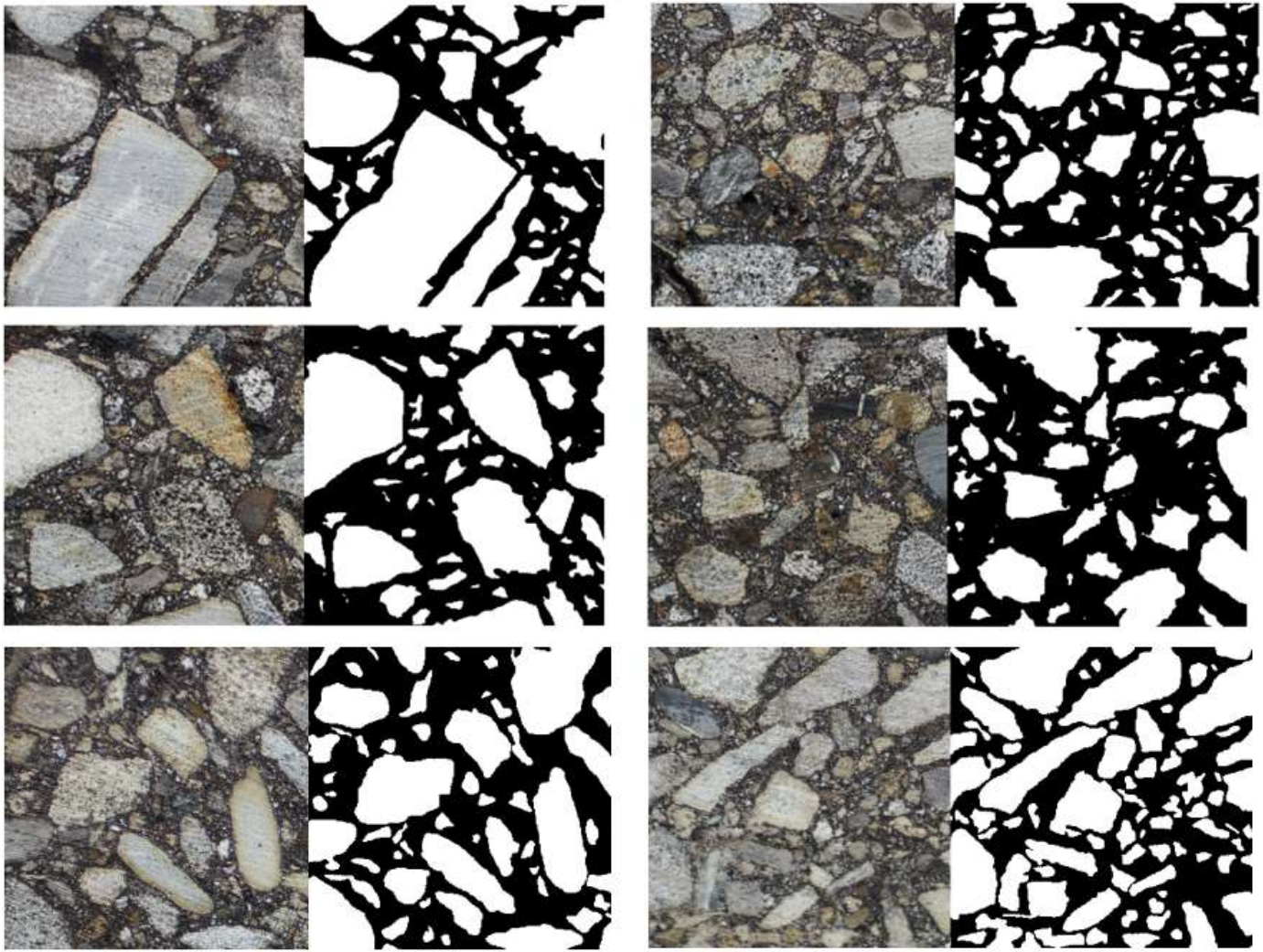
$$\mu_g=\frac{1}{NP\cdot a_g}\sum_{(x,y)}d(x,y)P\left(G\middle|d(x,y)\right)$$

$$\mu_a=\frac{1}{NP\cdot(1-a_g)}\sum_{(x,y)}d(x,y)\left(1-P\left(G\middle|d(x,y)\right)\right)$$

$$\Sigma_g=\frac{1}{NP\cdot a_g}\sum_{(x,y)}\left(d(x,y)-\mu_g\right)\cdot\left(d(x,y)-\mu_g\right)^T P\left(G\middle|d(x,y)\right)$$

$$\Sigma_a=\frac{1}{NP\cdot(1-a_g)}\sum_{(x,y)}\left(d(x,y)-\mu_a\right)\cdot\left(d(x,y)-\mu_a\right)^T\left(1-P\left(G\middle|d(x,y)\right)\right)$$





# Conclusiones I

- La inteligencia centralizada parece tener dificultades para segmentar (faltan procesos estándar por comparar, pero no es de esperar grandes mejoras)
- La auto-organización permite resultados casi tan satisfactorios (o tan insatisfactorios) como la segmentación manual.
- ¡Casi tres horas en una imagen de 3500x5000!
- ¿Cómo buscar reglas de autómatas celulares (cognitivos o no) que logren objetivos específicos?