

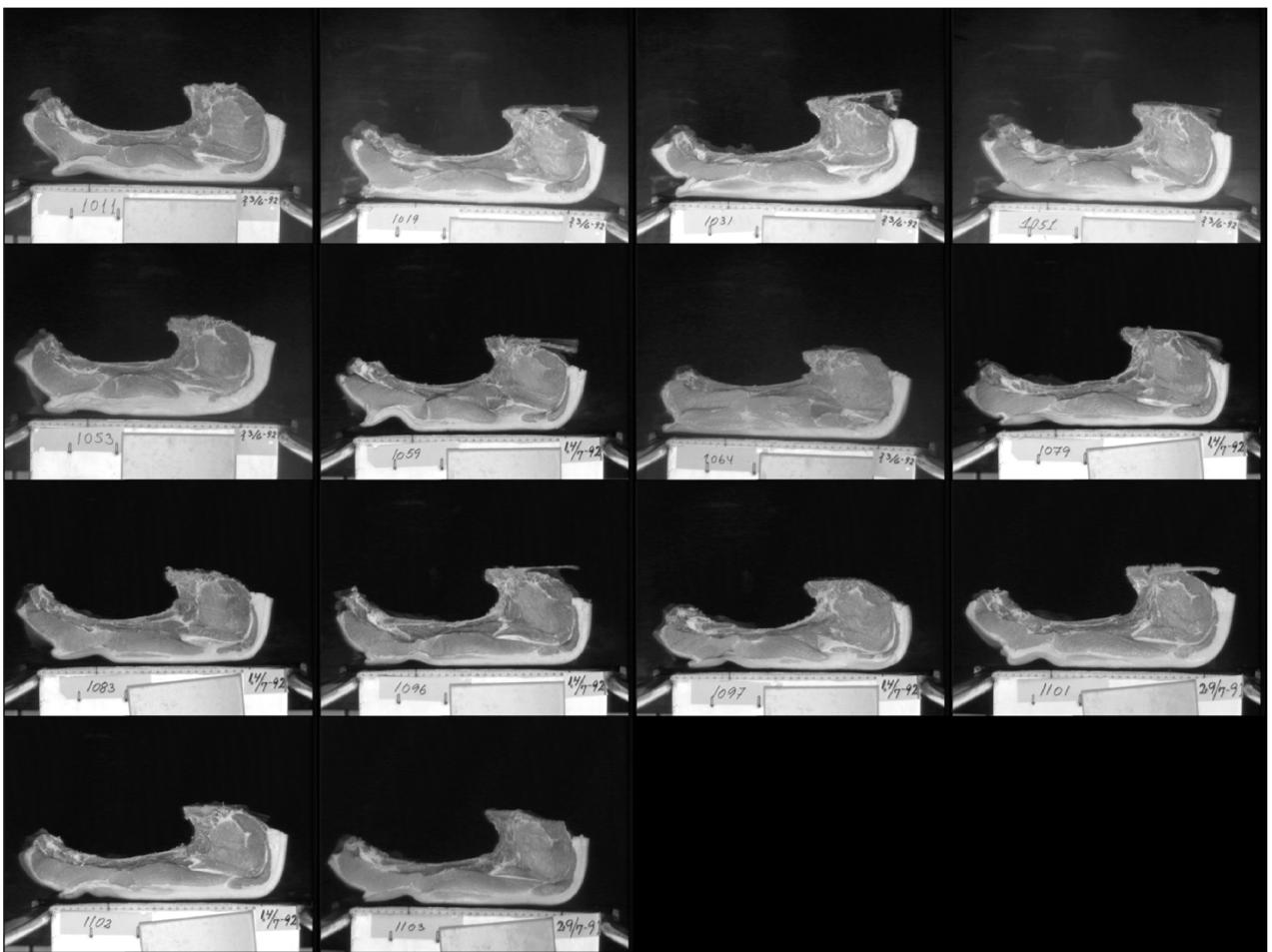
Binarització d'imatge "I" a partir del threshold "llindar"

```
function [IW,percen] = funcio(I, llindar)
IW = zeros(size(I));
IW(I < 80) = 0.5; % esborrat del background
IW(I > llindar) = 1; % binarització
blanc = (IW==1);
negre = (IW == 0);
% càlcul percentatge respecte els píxels totals de la carn
percen = sum(sum(blanc))/ (sum(sum(blanc))+sum(sum(negre)));
end
```

# Control de qualitat de peces de carn

## 0. Lectura d'imatges

```
% codi extret de myscanfiles.m
close all
a = dir('.\Chuletons\*.bmp');
nf = size(a);
figure
for i = 1:nf
filename = horzcat(a(i).folder,'/',a(i).name);
I(:,:,:,i) = imread(filename); % matriu de les 14 imatges
drawnow
end
montage({I(:,:,:,1),I(:,:,:,2),I(:,:,:,3),I(:,:,:,4),I(:,:,:,5),I(:,:,:,6),I(:,:,:,7),I(:,:,:,8),I(:,:,:,9), ...
I(:,:,:,10),I(:,:,:,11),I(:,:,:,12),I(:,:,:,13),I(:,:,:,14)})
```



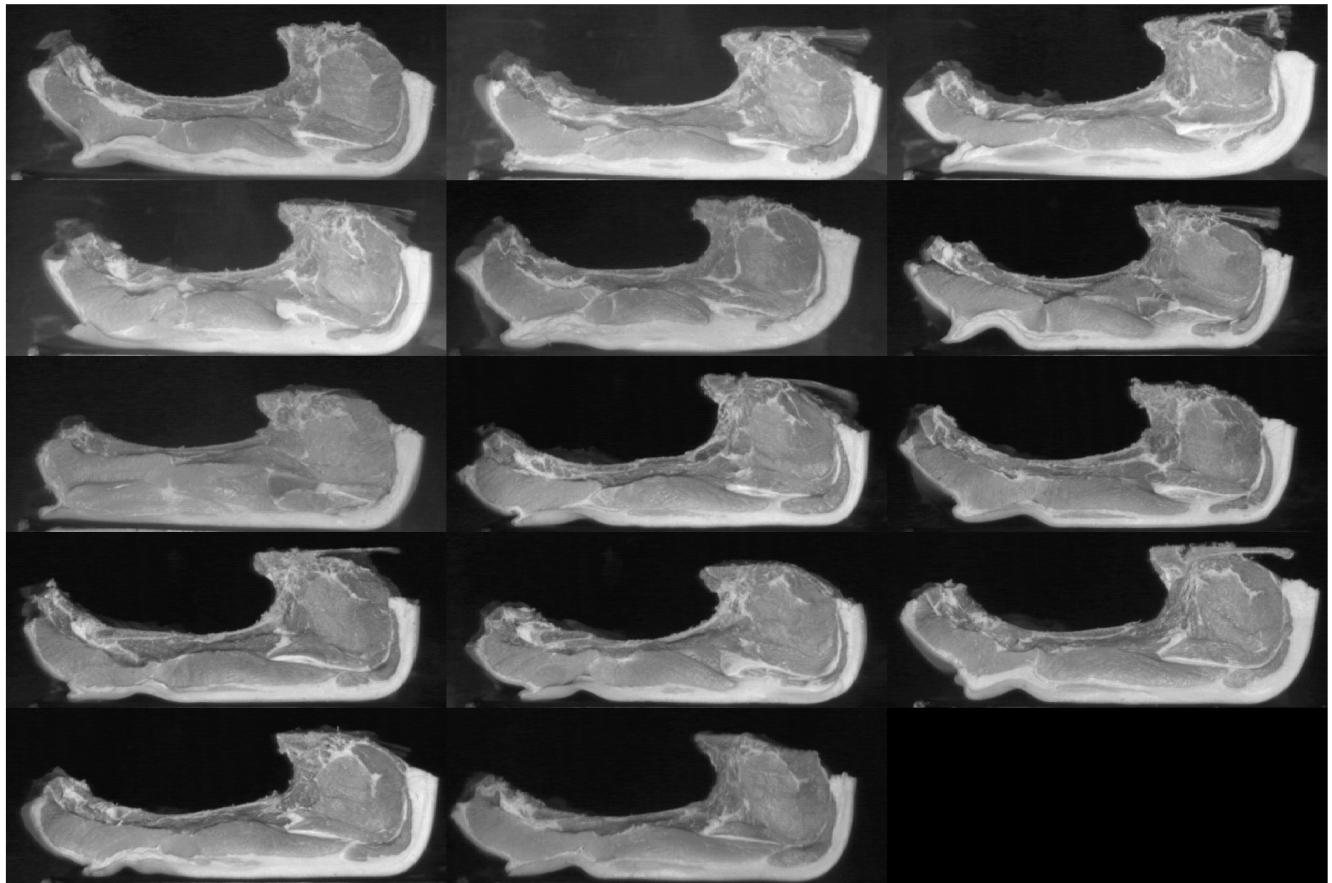
## 1. Retallat d'imatges

```
% offsets de l'eix Y per cada imatge, obtinguts a ull.
```

```

offsetY = [100,147,143,145,95,135,148,120,122,130,138,133,125,133];
for i = 1:14
    B(:,:,:,i) = I(offsetY(i)+55:offsetY(i)+330,15:705,i); % retallat
end
montage({B(:,:,:1),B(:,:,:2),B(:,:,:3),B(:,:,:4),B(:,:,:5),B(:,:,:6),B(:,:,:7),B(:,:,:8),B(:,:,:9),
B(:,:,:10),B(:,:,:11),B(:,:,:12),B(:,:,:13),B(:,:,:14)}) ...

```



## 2.Binarització programable

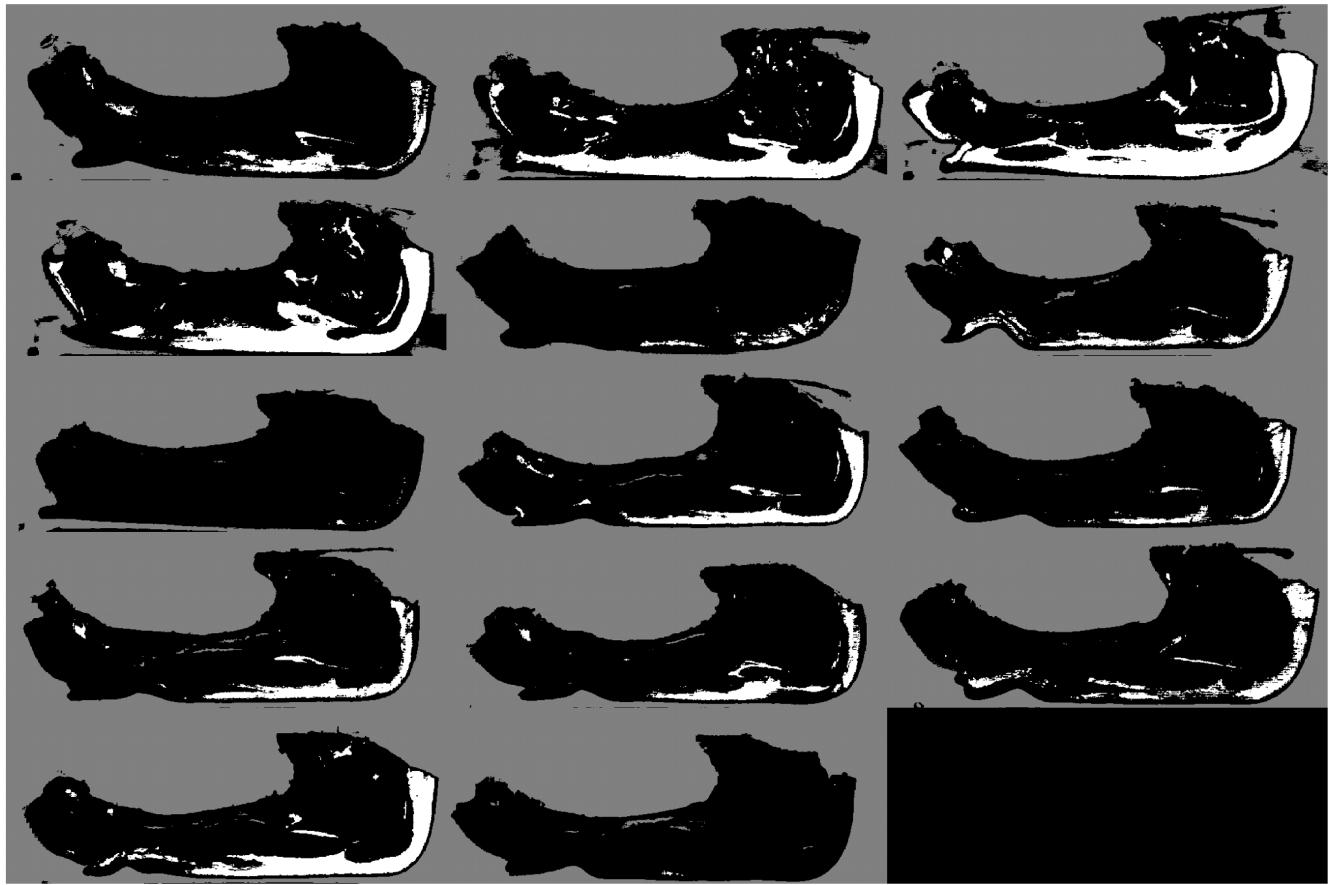
```

llindar = 200; % exemple per comprovar funcionament de "funcio"
% aquest és el paràmetre programable

for i = 1:14
    % obtenim l'imatge binaritzada i el seu percentatge de greix
    [C(:,:,:,i), percen2(i)] = funcio(B(:,:,:,i),llindar);

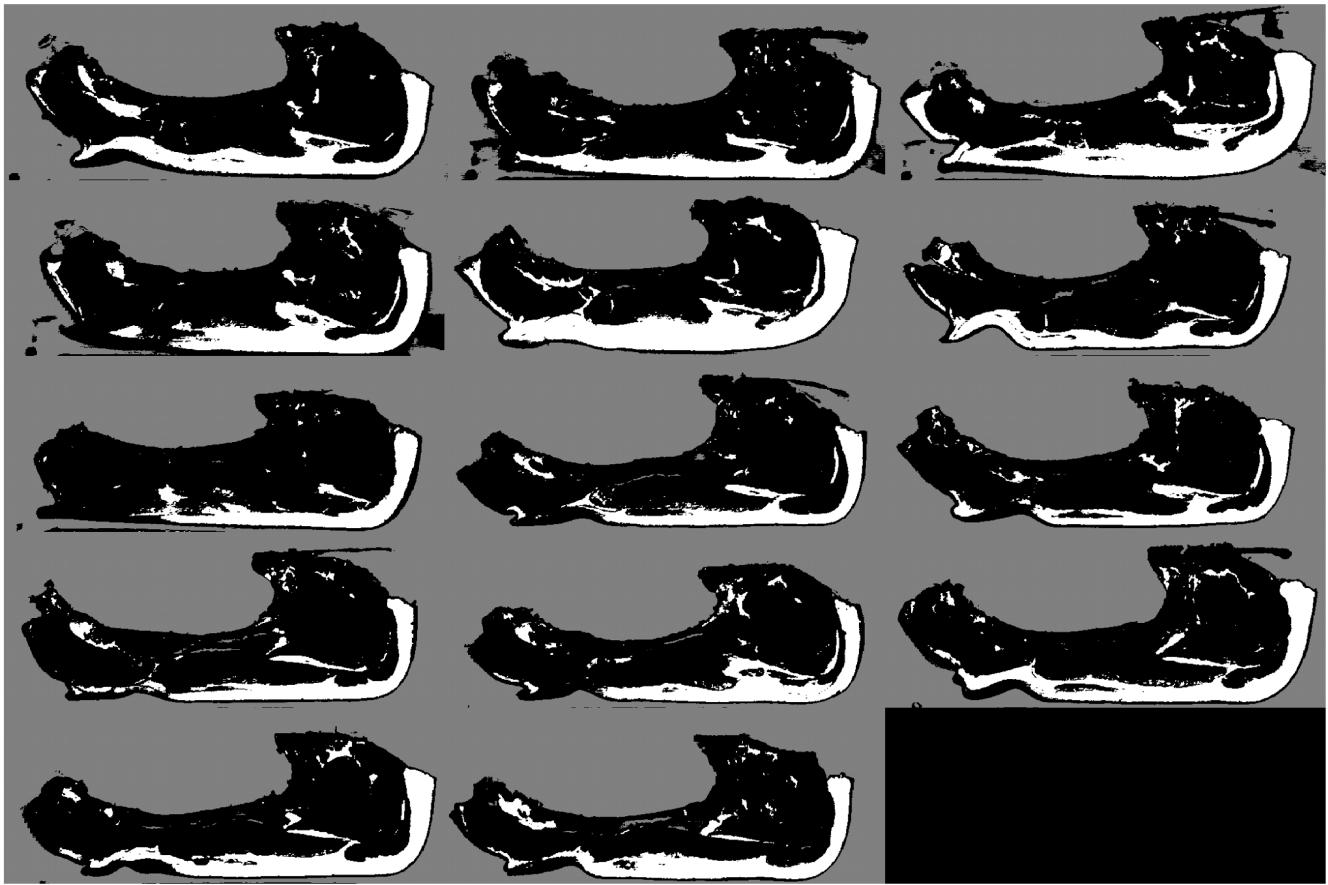
end
montage({C(:,:,:1),C(:,:,:2),C(:,:,:3),C(:,:,:4),C(:,:,:5),C(:,:,:6),C(:,:,:7),C(:,:,:8),C(:,:,:9),
C(:,:,:10),C(:,:,:11),C(:,:,:12),C(:,:,:13),C(:,:,:14)}) ...

```



### 3. Llindar per histogrames

```
% obtinguts a ull pels histogrames (agafant un valor aproximant a la seogna "vall")
llindars_ull = [172,203,194,200,155,174,166,180,166,173,172,176,181,151];
for i = 1:14
    % obtenim l'imatge binaritzada i el seu percentatge de greix
    [D(:,:,i), percen(i,1)] = funcio(B(:,:,i),llindars_ull(i));
end
montage({D(:,:,1),D(:,:,2),D(:,:,3),D(:,:,4),D(:,:,5),D(:,:,6),D(:,:,7),D(:,:,8),D(:,:,9),
D(:,:,10),D(:,:,11),D(:,:,12),D(:,:,13),D(:,:,14)})
```

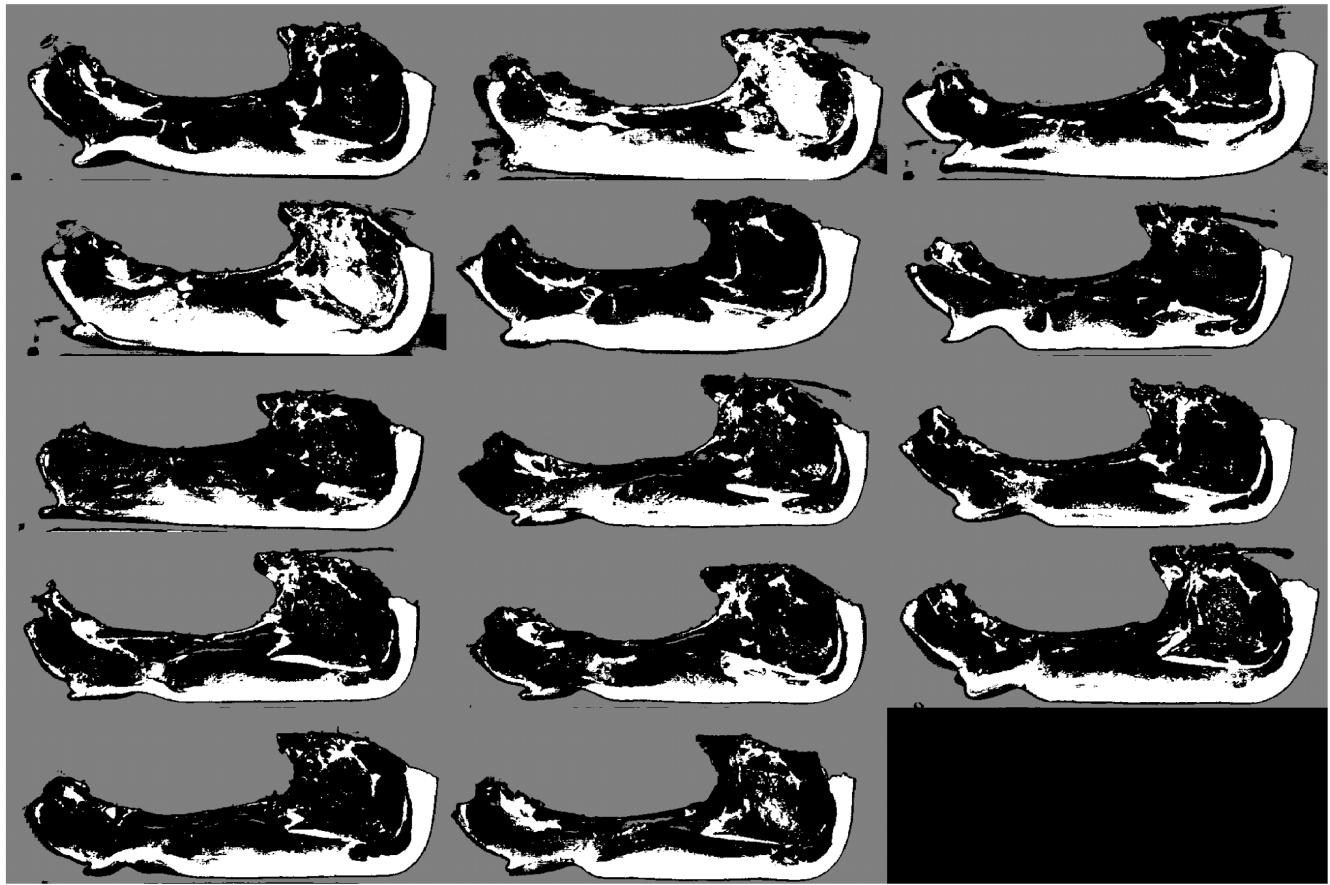


## 4.Llindar per graythresh

Hem utilitzat otsuthresh, equivalent a graythresh, que ens permet passar com a paràmetre un histograma, en comptes d'una imatge. Això ens permet eliminar els valors corresponents al background.

```

for i = 1:14
    hist = imhist(B(:,:,i));
    % esborrem els valors del fons per tal que no interfereixin amb el càlcul del threshold
    hist(1:80) = 0;
    llindar(i) = otsuthresh(hist)*255;
    [E(:,:,:,i), percen(i,2)] = funcio(B(:,:,:,i),llindar(i));
end
montage({E(:,:,:,1),E(:,:,:,2),E(:,:,:,3),E(:,:,:,4),E(:,:,:,5),E(:,:,:,6),E(:,:,:,7),E(:,:,:,8),E(:,:,:,9),
E(:,:,:,10),E(:,:,:,11),E(:,:,:,12),E(:,:,:,13),E(:,:,:,14)})
```



## 5 a) Llindar per mitjana

Primer mètode propi per calcular el threshold:

Per cada imatge agafem la mitjana ponderada dels valors dels píxels que NO pertanyen al fons

```

for i = 1:14
    hist = imhist(B(:,:,i));
    % esborrem els valors del fons per tal que no interfereixin amb el càlcul del threshold
    hist(1:80) = 0;

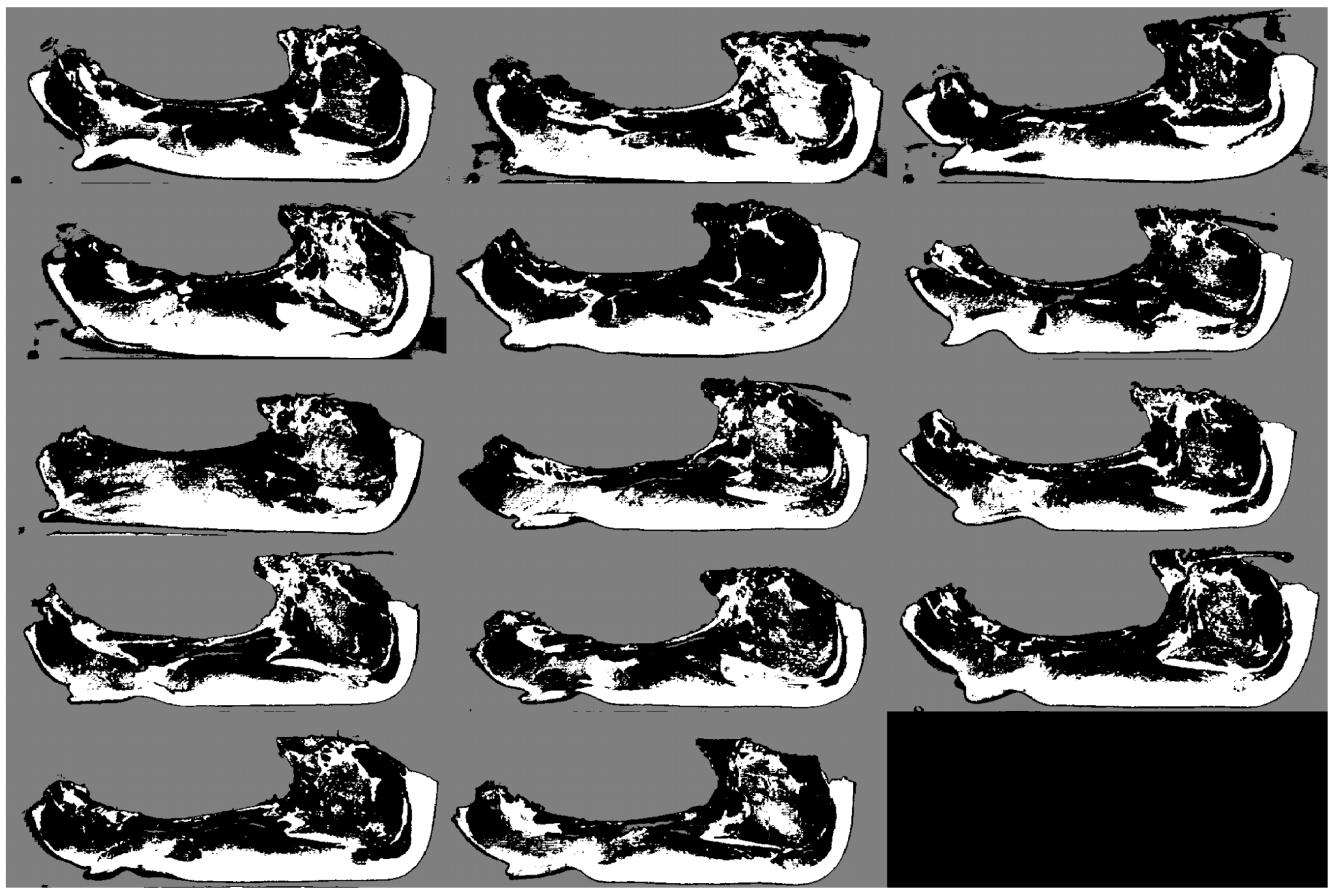
    haccum = cumsum(hist);
    [f, c] = size(B(:,:,i));
    hprob = hist / (f*c);
    %plot(hprob);

    w0 = sum(hprob(1:255));
    llindar(i) = sum((1:255)*hprob(1:255))/w0; % càlcul de la mitjana ponderada de l'imatge

    % resultat
    [G(:,:,:,i), percen(i,3)] = funcio(B(:,:,:,i),llindar(i));
end
montage({G(:,:,:,1),G(:,:,:,2),G(:,:,:,3),G(:,:,:,4),G(:,:,:,5),G(:,:,:,6),G(:,:,:,7),G(:,:,:,8),G(:,:,:,9), ...

```

```
G(:,:,10),G(:,:,11),G(:,:,12),G(:,:,13),G(:,:,14)})
```



## 5 b) Otsu simplificat

Aquest és el segon mètode implementat:

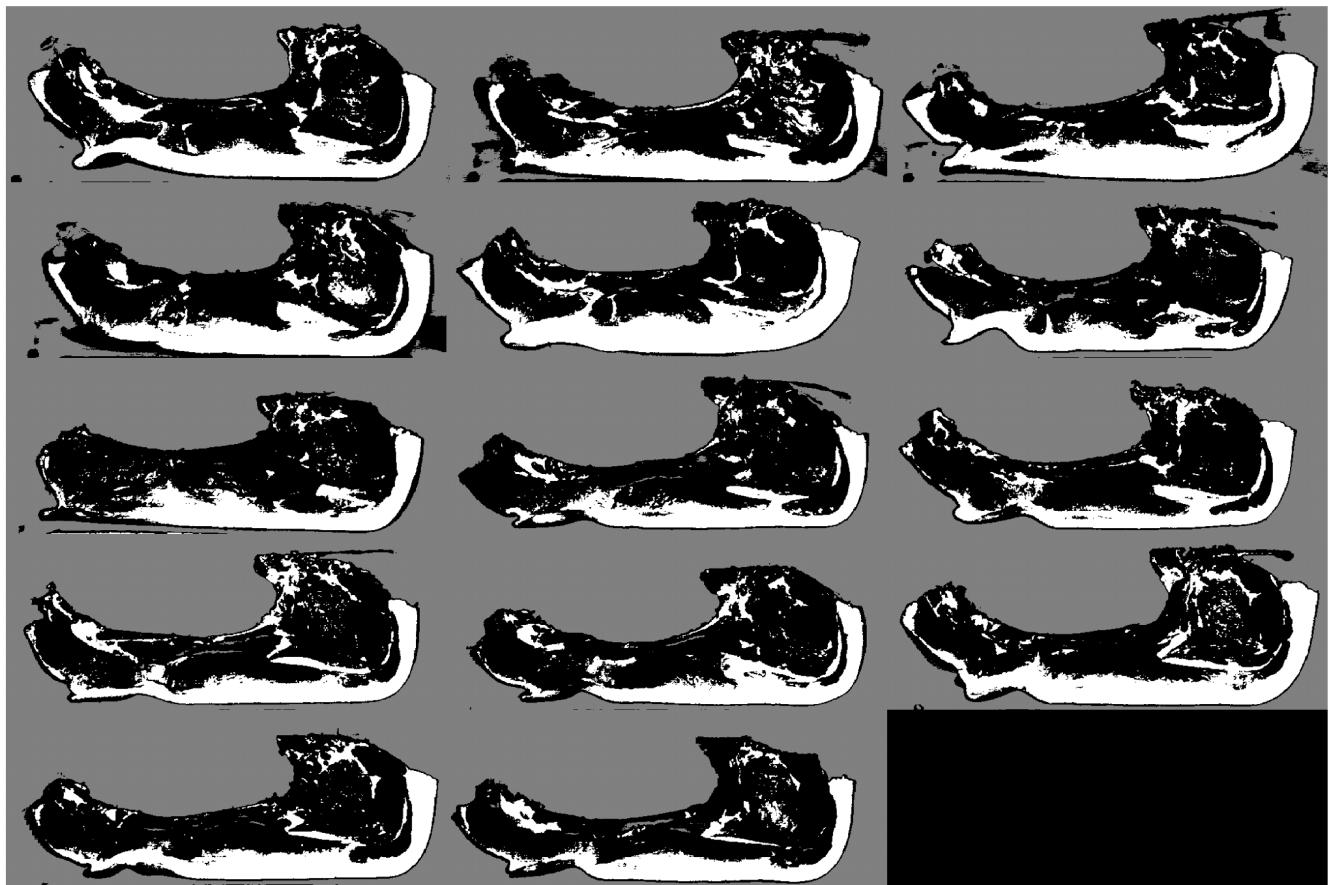
Otsu, amb codi propi, simplificat on només es té en compte la variància i no la mitjana. Això permet estalviar càcul mantenint un resultat acceptable.

```
for k=1:14  
  
hist = imhist(B(:,:,k));  
hist(1:80) = 0; % eliminació de background  
  
hacum = cumsum(hist);  
[f, c] = size(B(:,:,k));  
hprob = hist / (f*c);  
%plot(hprob);  
  
% declarem els vectors  
w0 = zeros(255,1);  
w1 = zeros(255,1);  
%guardarem el maxim
```

```

ob = 0;
lind = 0;
for i=1:255
    % w
    w0(i) = sum(hprob(1:i));
    w1(i) = 1 - w0(i);
    aux = w0(i)*w1(i)/i; % dividim per la i per evitar que s'agafi un llindar massa alt
    % guardem el maxim
    if (aux > ob), ob = aux; lind = i;
    end
end
% resultat
[Res(:,:,:,k),percen(k,4)] = funcio(B(:,:,:,:,k),lind);

end
montage({Res(:,:,:,1),Res(:,:,:,2),Res(:,:,:,3),Res(:,:,:,4),Res(:,:,:,5),Res(:,:,:,6),Res(:,:,:,7), ...
Res(:,:,:,8),Res(:,:,:,9), ...
Res(:,:,:,10),Res(:,:,:,11),Res(:,:,:,12),Res(:,:,:,13),Res(:,:,:,14)})
```



## Taula percentatges greix

Taula de resultats de els percentatges de cada imatge obtinguts a partir dels mètodes:

Columna 1: Apartat 3, binarització per histogrames

Columna 2: Apartat 4, binarització per otsu

Columna 3: Apartat 5 a, binarització per mètode propi 1

Columna 4: Apartat 5 b, binarització per mètode propi 2

La taula s'imprimeix dividida en 3 seccions degut a limitacions del matlab.

```
percen(1:5,:)
```

```
ans = 5x4
0.2134    0.2894    0.3896    0.3263
0.1871    0.5492    0.4917    0.3113
0.3167    0.3889    0.4312    0.3889
0.2021    0.5166    0.4839    0.3174
0.3278    0.3564    0.4029    0.4114
```

```
percen(6:10,:)
```

```
ans = 5x4
0.1946    0.2850    0.4112    0.2976
0.1425    0.2680    0.4369    0.2680
0.1587    0.3104    0.4469    0.2628
0.1776    0.2772    0.4190    0.2772
0.1628    0.2878    0.4229    0.3071
```

```
percen(11:14,:)
```

```
ans = 4x4
0.1997    0.2960    0.4054    0.2907
0.2084    0.3326    0.4126    0.3475
0.2121    0.2819    0.3828    0.3019
0.2115    0.3213    0.4332    0.2798
```