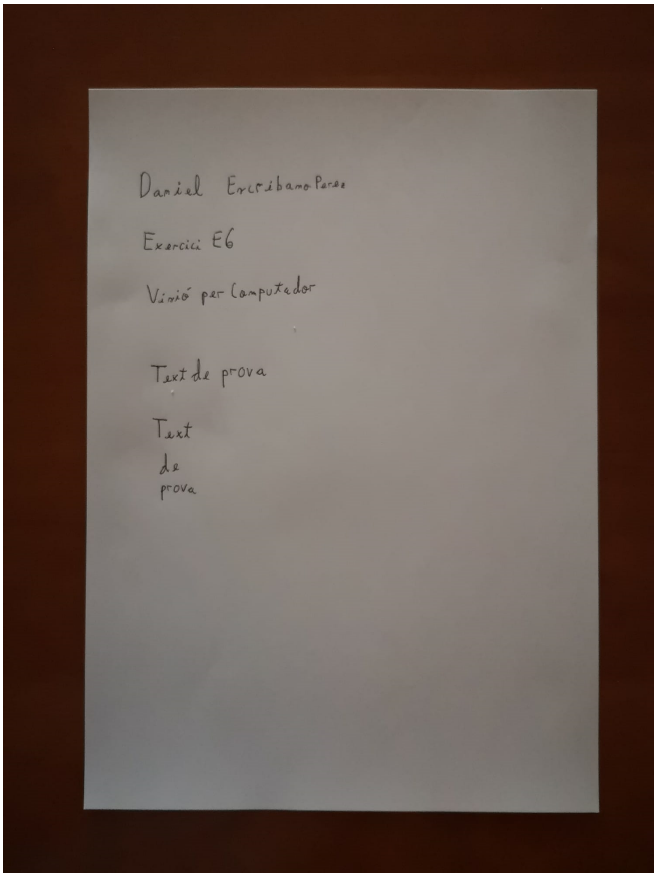
Daniel Escribano Pérez

VISIÓ PER COMPUTADOR

Exercici 6 de Laboratori



Cal notar que el text està escrit en llapis. En escriure amb bolígraf blau o negre els resultats no eren bons, en realitzar la binarització les lletres quedaven incompletes.

```
I = rgb2gray(imread('DN2.jpg'));
perc = 0.649;
llindar = globalBinE6(I,perc);
BW = I > llindar;
```

CODI DE LA FUNCIÓ globalBinE6

```
function [llindar] = globalBinE6(I,perc)

h = imhist(I);
chist = cumsum(h);
totalPixels = chist(end);
chistUnder = chist > (totalPixels - totalPixels*perc);
llindar = find(chistUnder,1);

end
```

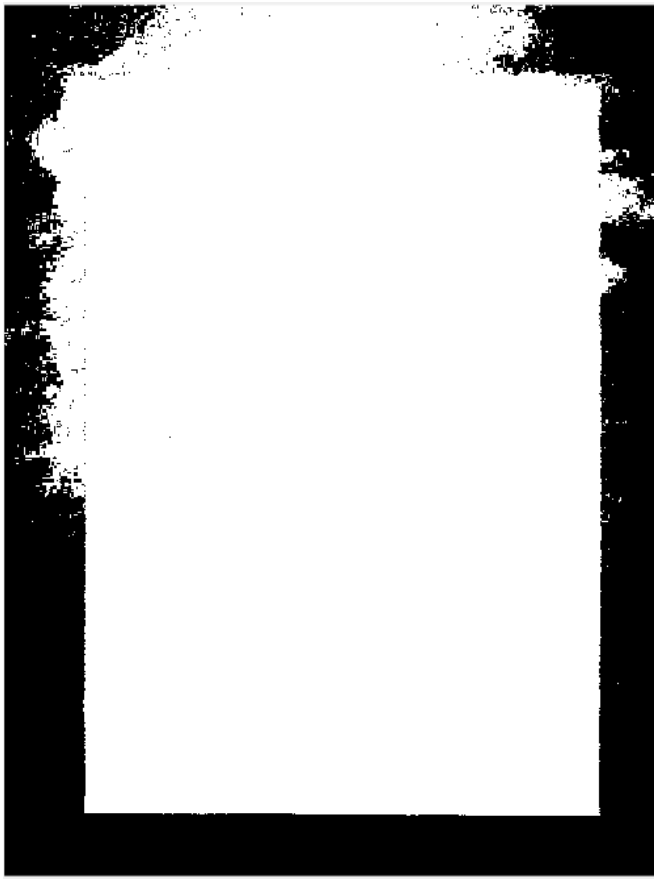


Figura 1: Resultat de binaritzar el 80% de la imatge a blanc

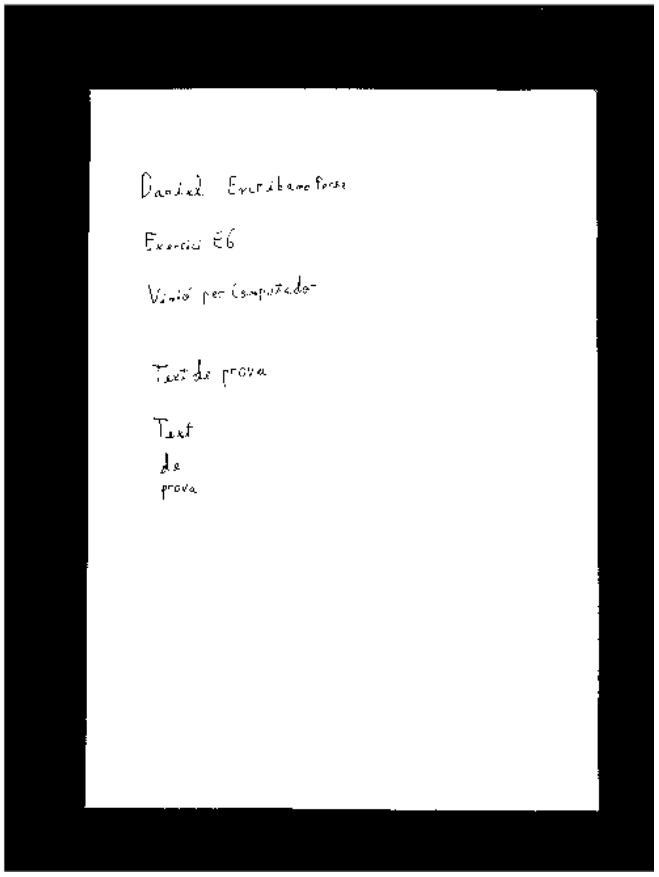


Figura 2: Resultat de binaritzar el 64.9% de la imatge a blanc

Per tal de realitzar la binarització global he fet servir la funció *cumsum* de MATLAB, que realitza la suma acumulativa de un histograma, es a dir, va sumant el nombre de pixels que hi ha a cada nivell de gris(el nivell 2 te la suma del nivell , del nivell 0 i la del propi nivell 2, i així progressivament). Amb la variable chistUnder posem a 1 els nivells de grisos que assoleixen el % indicat i amb la variable llindar trobem el primer valor que está a 1, el nivell de gris que marca el llindar.
En realitzar diverses proves he vist que amb un percentatge del 64.9% obtenim un bon resultat.

CC = bwconncomp(BW);
R = regionprops(CC, 'BoundingBox', 'Area');
J = imcrop(BW, [184 191 1178 1655]);

Per tal de retallar la imatge he fet servir la funció *regionprops* per generar una llista amb el bounding box i l'àrea de tots els components 8 connectats de la imatge. Per obtenir aquesta llista necessitem invocar abans la funció bwconncomp que ens proporciona aquests components.
De la variable R deduïm que l'objecte més gran i amb més area correspon al full, per tant agafem les dimensions que ens proporcionar la columna BoundingBox i retallem en funció a aquestes.

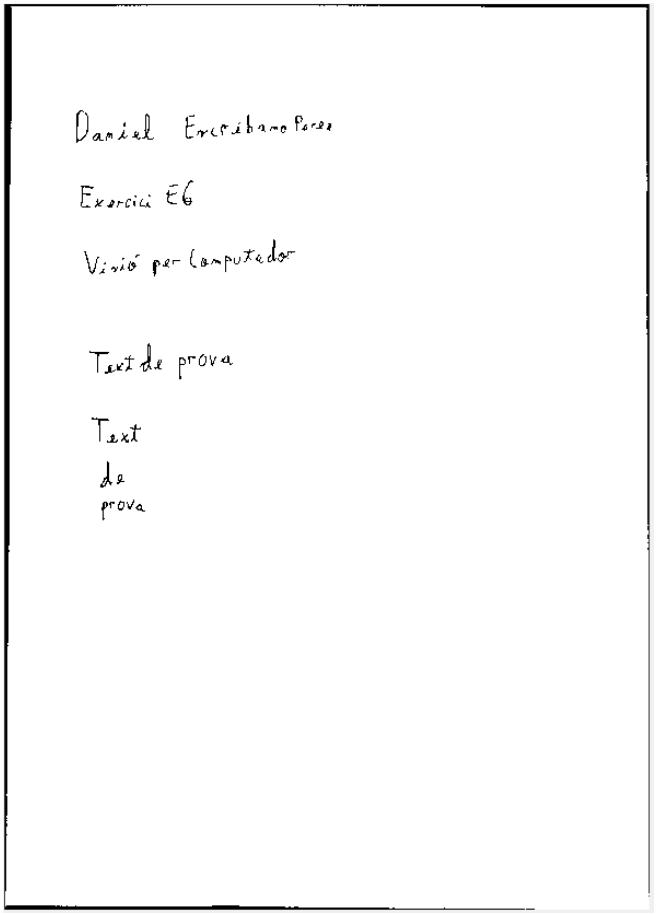


Figura 3: Resultat de retallar la imatge amb les dimensions de la Bounding Box

```
AfterColFilt = colfilt(I, [3 1], 'sliding', @localBinE6, 20);
 CODI DE LA FUNCIÓ localBinE6
function [RES] = localBinE6(I,K)

    [f c] = size(I);
    RES = I(ceil(f/2),1); %linia central
    suma = sum(I);
    mean = suma /f;
    RES = ((RES - uint8(mean)) > K)*255;

end
```

Fent ús de les eines que proporciona MATLAB, he observat la imatge original i he trobat que, en general, les línies ocupen uns 40 píxels i els caràcters uns 30. Això varia en funció del caràcter i la línia, ja que al haver escrit a mà el text no tots els caràcters tenen la mateixa mida. Si realitzem el binaritzat local amb aquests valors obtenim el següent:
AfterColFilt = colfilt(I, [40 30], 'sliding', @localBinE6, 10);

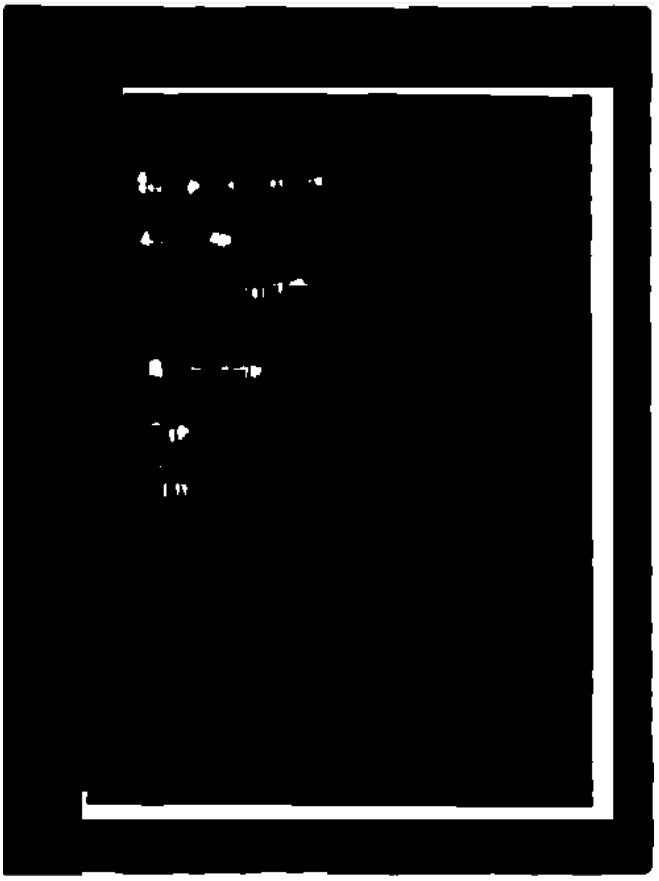


Figura 4: Resultat en aplicar una finestra lliscant de [40 30]

Com podem veure, el resultat no és satisfactori. Per tal de trobar uns valors correctes per a la finestra lliscant vaig anar provant diversos valors més petits. He trobat que amb una finestra lliscant de [5 5] obtenim un resultat acceptable.

Pel que fa al nombre de nivells de grisos al que ha de ser superior el píxel actual, he decidit deixar un valor de 10.

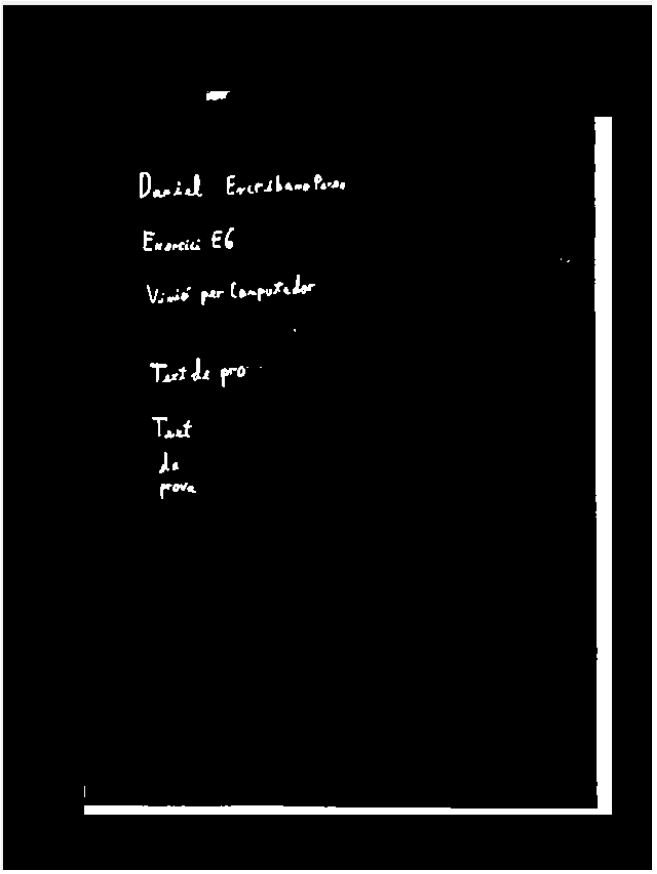


Figura 5: resultat en aplicar una finestra lliscant de [5 5]

Per tal de eliminar les taques petites que observem a la imatge he realitzat un open amb un element estructurant de tipus disk.

SE = strel('disk',3);
RES = imopen(AfterColFilt, SE);

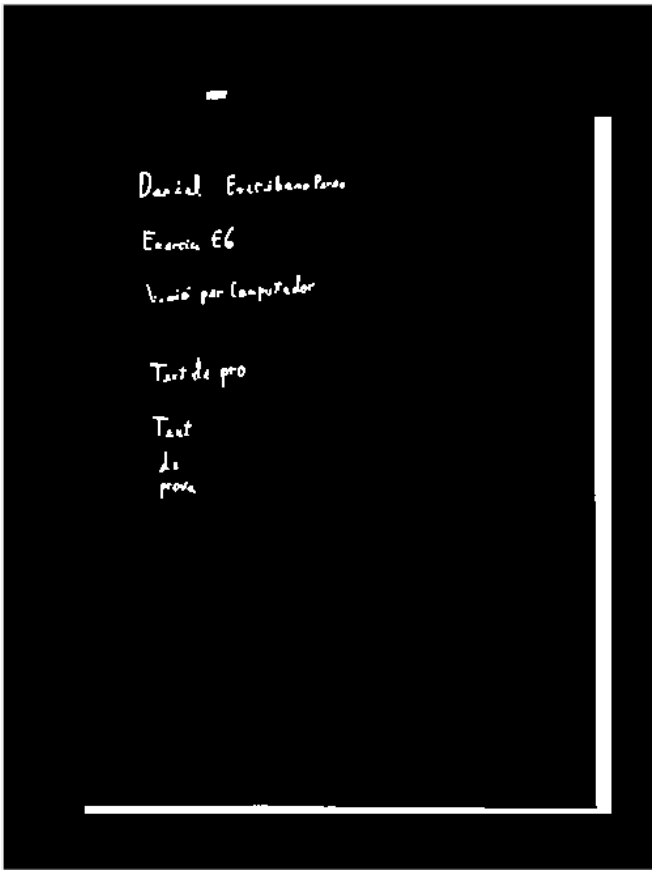


Figura 6:Resultat en aplicar el open

Després de realitzar el open, fent ús de la funció *bwconncomp* he comptat el nombre de lletres que hi ha a la imatge.El nombre de objectes(67) s'aproxima al nombre de lletres reals(70). Com podem veure en el resultat, hem perdut un parell de lletres i ha aparegut una taca a la part superior i un delineament al full.