Computer vision 1

Homework 2

姓名 ： 蘇宛琳

系所 ： 電信所碩一

學號 ： R05942060

指導教授 ： 傅楸善老師

Computer Vision Report – Homework 2

R05942060 蘇宛琳

Write a program to generate:

a binary image (threshold at 128)

將影像取二值化，就是將影像變成 1.0 這兩種二元值。也就是說，當亮度大於 128 的值就顯示成白色（255）;反之小於 128 的值就顯示成黑色(0)。 在簡化來說就是將影像變成黑白兩種色階。由臨界值 128 來決定此圖片的 0 或是 1 位元。

lena.bmp binarizelena.bmp

Source Code (binary)

LENA = imread('lena.bmp');

INFO = imfinfo('lena.bmp');

for x = 1:INFO.Height,

for y = 1:INFO.Width,

T = 128; %設定binary門檻值

if LENA(x,y) > T,

LENA(x,y) = 255; %超過門檻值，顯示白色（1）

else

LENA(x,y) = 0; %低於門檻值，顯示黑色（0）

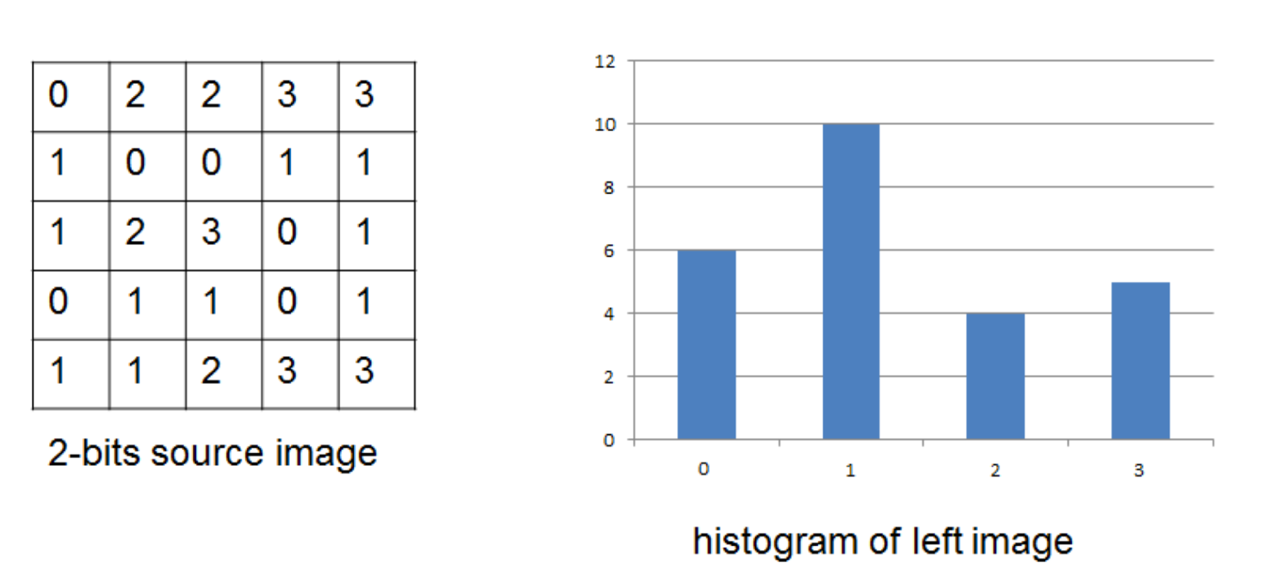
end; end;

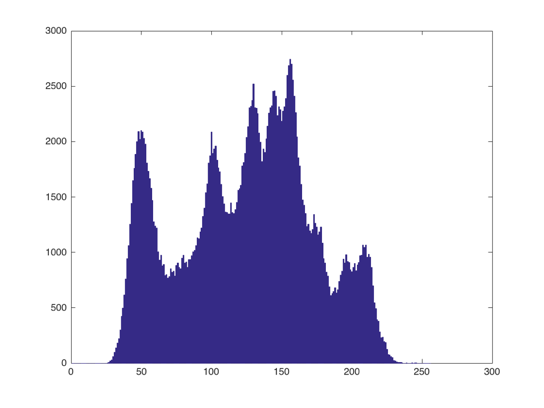
end;

imwrite(LENA,'binarizelena.bmp');

Source Code (histogram)

histogram 直方圖就是計算每一個像素值出現的次數，依此頻率所繪製的直線圖(bar 圖)，首先將影像載入程式中，再針對每個像素逐一計算 256(0-255)種灰度值出現的 頻率，如以下圖作為概念:



lena.bmp LENA Histogram

計算LENA中0-255位元數出現次數 做成統計圖

LENA = imread('lena.bmp');

INFO = imfinfo('lena.bmp');

HISTO = zeros(256,1);

for i = 1:INFO.Height,

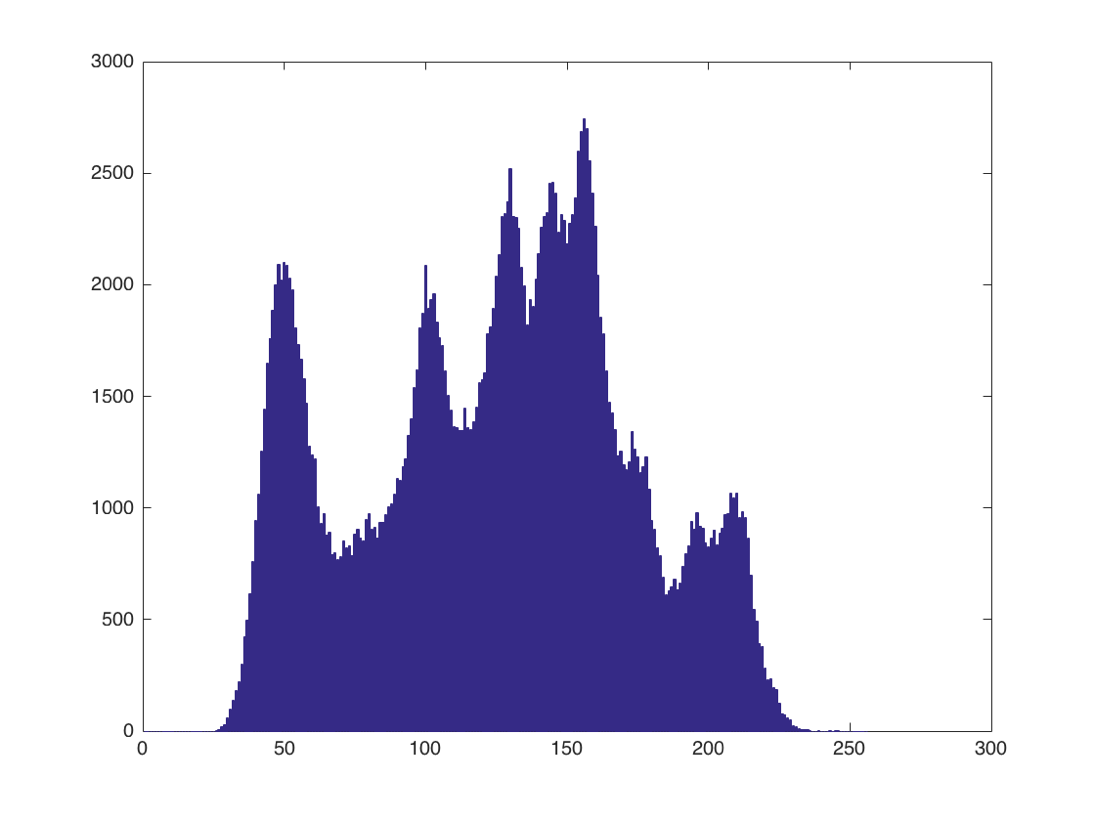
for j = 1:INFO.Width,

HISTO(LENA(i,j)+1) = HISTO(LENA(i,j)+1)+1; %計算每個元素的次數

end;

end;

bar(HISTO)



histogram.bmp

Source Code (connected components :regions with + bounding box)

參考課本內容中提到的的方法『An Iterative Algorithm』，以及 『4-connected neighborhood detection』來實現此題的圖像 connected components 分析。iterative的流程三步驟(1)將圖像中每個像素做初始化，至對應的標籤 label 中。(2)分別從 top-down & bottom-up passes ,來與鄰近相鄰的點做比較，取較小的編號值。(3)檢視到兩個方向的 pass 都沒有變化後才算成功。

lena.bmp binarizelena.bmp

4-connected component

LENA = imread('lena.bmp');

INFO = imfinfo('lena.bmp');

for x = 1:INFO.Height,

for y = 1:INFO.Width,

T = 128; % binarize lena

if LENA(x,y) > T,

LENA(x,y) = 255;

else

LENA(x,y) = 0;

end;

end;

end;

imwrite(LENA,'bilena.bmp')

An Iterative Algorithm

LENA1 = imread('bilena.bmp');

INFO1 = imfinfo('bilena.bmp');

mm = 0;

LABEL = zeros(INFO1.Height,INFO1.Width); % label binarizedlena

for x = 1:INFO1.Height,

for y = 1:INFO1.Width,

if LENA1(x,y) > 0,

mm = mm + 1;

LABEL(x,y) = mm;

end;

end;

end;

change = 1;

while change > 0,

change = 0;

for x = 1:INFO1.Height,

for y = 1:INFO1.Width,

if LABEL(x,y) > 0,

min = LABEL(x,y);

if x > 1 && LABEL(x-1,y) ~= 0 && LABEL(x-1,y) < min,

min = LABEL(x-1,y); % top-to-bottom

end;

if y > 1 && LABEL(x,y-1) ~= 0 && LABEL(x,y-1) < min,

min = LABEL(x,y-1); % left-to-right

end;

if min ~= LABEL(x,y),

change = 1;

LABEL(x,y) = min;

end;

end;

end;

end;

for x = INFO1.Height:-1:1,

for y = INFO1.Width:-1:1,

if LABEL(x,y) > 0,

min = LABEL(x,y);

if x<INFO1.Height && LABEL(x+1,y)~=0 && LABEL(x+1,y)< min,

min = LABEL(x+1,y); % bottom-to-top

end;

if y<INFO1.Width && LABEL(x,y+1)~=0 && LABEL(x,y+1)< min,

min = LABEL(x,y+1); % right-to-left

end;

if min ~= LABEL(x,y),

change = 1;

LABEL(x,y) = min;

end;

end;

end;

end;

end;

Regions

REGION = zeros(mm,1);

for x = 1:INFO1.Height,

for y = 1:INFO1.Width,

if LABEL(x,y) > 0,

REGION(LABEL(x,y)) = REGION(LABEL(x,y))+1;

end;

end;

end;

bounding box

for r = 1:mm;

if REGION(r) >= 500,

top = INFO1.Height;

bottom = -1;

left = INFO1.Width;

right = -1;

for x = 1:INFO1.Height,

for y = 1:INFO1.Width,

if LABEL(x,y) == r,

if x < top,

top = x;

end;

if x > bottom,

bottom = x;

end;

if y < left,

left = y;

end;

if y > right,

right = y;

end;

end;

end;

end;

for i = top:bottom,

LENA1(i,left) = 128;

LENA1(i,right) = 128;

end;

for j = left:right,

LENA1(top,j) = 128;

LENA1(bottom,j) =128;

end;

imwrite(LENA1,'four\_connected\_iterative1.bmp')

end;

end;

lena.bmp binarizelena.bmp

結果圖



four\_connected\_iterative1.bmp