HW3_M11015069_劉宇哲

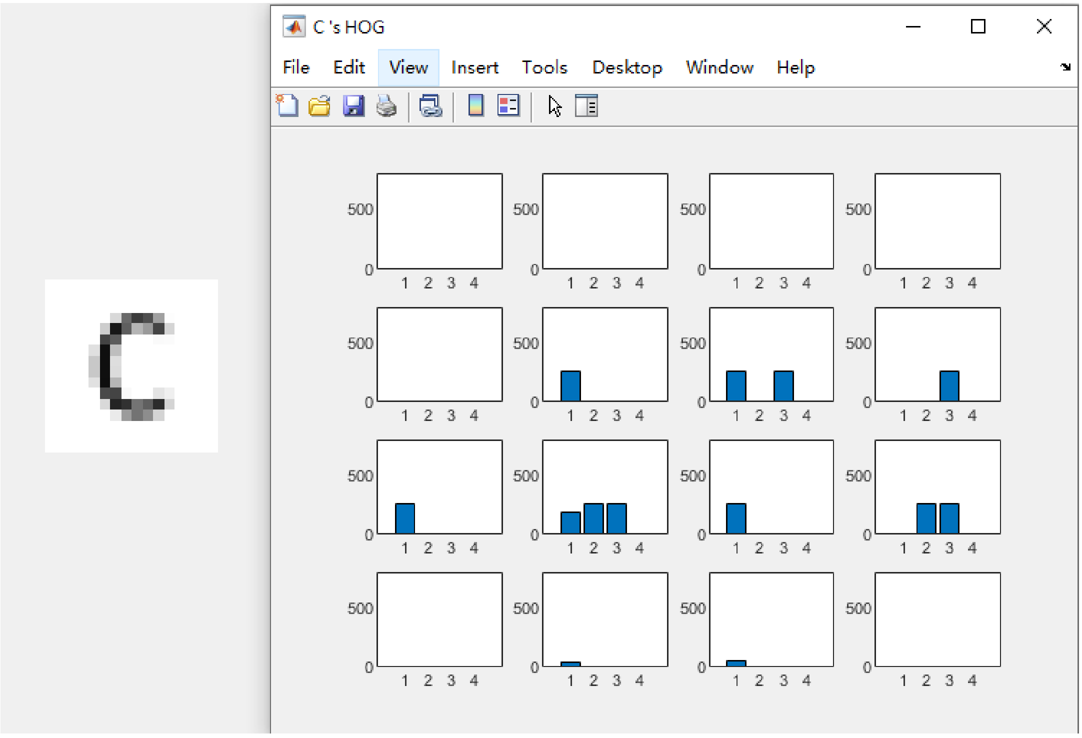
程式碼部分：

```
clc,clear,mainFunc;

function mainFunc()
select = input("請輸入要讀的圖檔編號,01~26:\n",'s'); %輸入要測試的字母編號，型態為字串
file_name = strcat("ABC",select,".jpg"); %合併字串
letter =
["A" ,"B" ,"C" ,"D" ,"E" ,"F" ,"G" ,"H" ,"I" ,"J" ,"K" ,"L" ,"M" ,"N" ,"O"
,"P" ,"Q" ,"R" ,"S" ,"T" ,"U" ,"V" ,"W" ,"X" ,"Y" ,"Z"];

img = imread(file_name); %讀檔
img = im2gray(img); %轉灰階
img = imresize(img, [16,16]); %改圖大小

kernal_x = [-1 0 1]; %sobel y & x 方向側邊算子
kernal_y = transpose(kernal_x);

G_x = imfilter(img, kernal_x, 'replicate'); %用 kernal_y & kernal_x 做卷積
G_y = imfilter(img, kernal_y, 'replicate');

M = abs(G_x) + abs(G_y); %gradient magnitude
% M = double(M)/255*5;

[row, col] = size(img);

theta = atan2d(double(G_y) ,double(G_x)); %gradient 的方向(角度)
theta = mod(theta ,180);
theta =floor(theta/45)+1;

step = 4; %cell_size = step*step
bins = 4; %gradient 方向分四個
number = 1;

%會在視窗顯示 input 的 letter
figure(Name= letter(str2double(select)) + " 's HOG",NumberTitle="off")
```

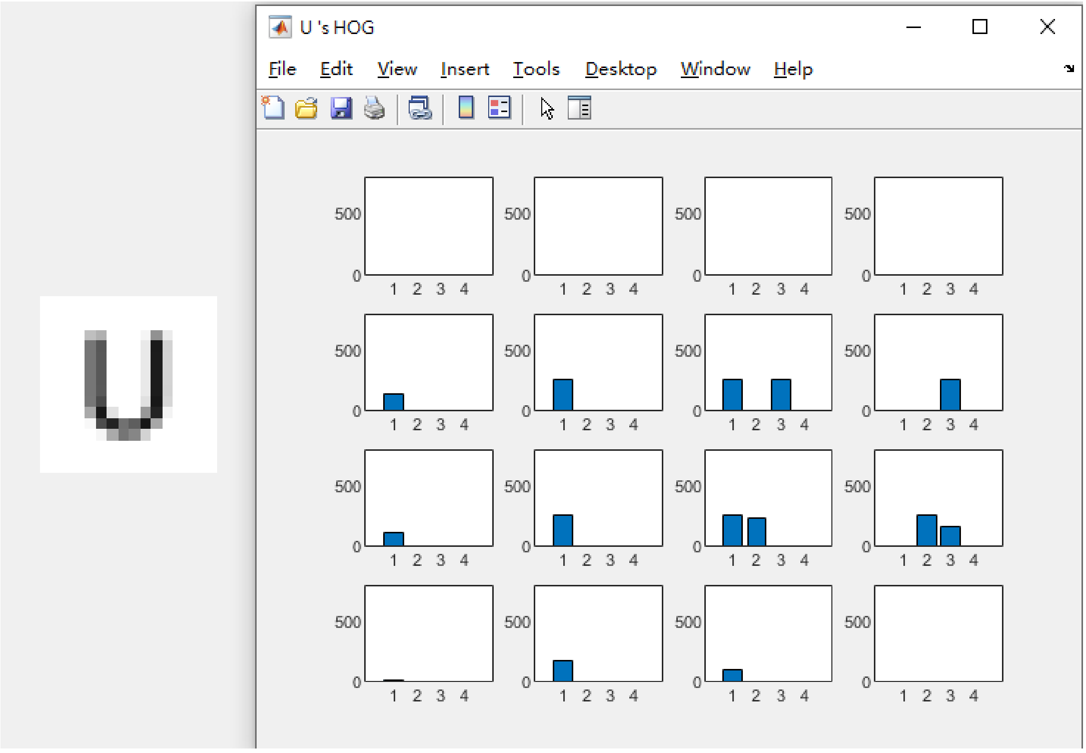
```
%取得每一個cell左上角的位置，去做HOG
for j = 1:step:col
    for i = 1:step:row
        [x,y] = bulidHOG(theta ,M ,step ,bins ,i ,j);
        subplot(4,4,number);
        bar(x,y); %畫直方圖
        ylim([0 800]); %限制y軸大小
        number = number +1;
    end
end
figure(Name= letter(str2double(select)) ,NumberTitle="off")
imshow(img); %要改跟HOG顯示一起



%ROI
step = 16;

test_img = imread("t1.jpg");
test_img = imresize(test_img ,[128,128]);
test_rgb = cat(3 ,test_img ,test_img ,test_img); %將圖存一個3通道的版本好上色

test_x = imfilter(test_img ,kernal_x ,'replicate'); %做卷積
test_y = imfilter(test_img ,kernal_y ,'replicate');
test_M = abs(test_x) + abs(test_y);

[test_row ,test_col] = size(test_img);

test_size = 128/16;
test_size = test_size*test_size;
array = zeros(1 ,test_size);
number = 1;

for j = 1:step:test_col %對圖每16*16依序做L2 norm
    for i = 1:step:test_row
        region = test_M(i:i+step-1 ,j:j+step-1 ,:);
        L2_error = norm(double(M-region) ,2);
        array(number) = L2_error;
        number = number+1;
    end
end

[min_three ,sequence] = sort(array ,'ascend'); %找出最小的幾個

for i = 1:2:5
    init_x = floor(sequence(i)/8)*16+1;
%     init_y = (mod(sequence(i) ,8)-1)*16+1;
    init_y = (mod(sequence(i) ,8));
    if init_y == 0
        init_y = 7*16+1;
    else
        init_y = (init_y - 1)*16+1;
    end
    test_rgb = getMARK(test_rgb ,init_x ,init_y); %最小的幾個要上色
end

for i = 1:3
    init_x = floor(sequence(i)/8)*16+1;
%     init_y = (mod(sequence(i) ,8)-1)*16+1;
    init_y = (mod(sequence(i) ,8));
    if init_y == 0
        init_y = 7*16+1;
    else
        init_y = (init_y - 1)*16+1;
    end
    test_rgb = getMARK(test_rgb ,init_x ,init_y); %最小的幾個要上色
end

%秀出上色結果
figure(Name="identify "+letter(str2double(select)) ,NumberTitle="off");
imshow(test_rgb);

end

%為每個 cell 繪製 HOG
function [x_axis ,y_axis] = bulidHOG(the ,mag ,step ,bins ,init_i ,init_j)
```

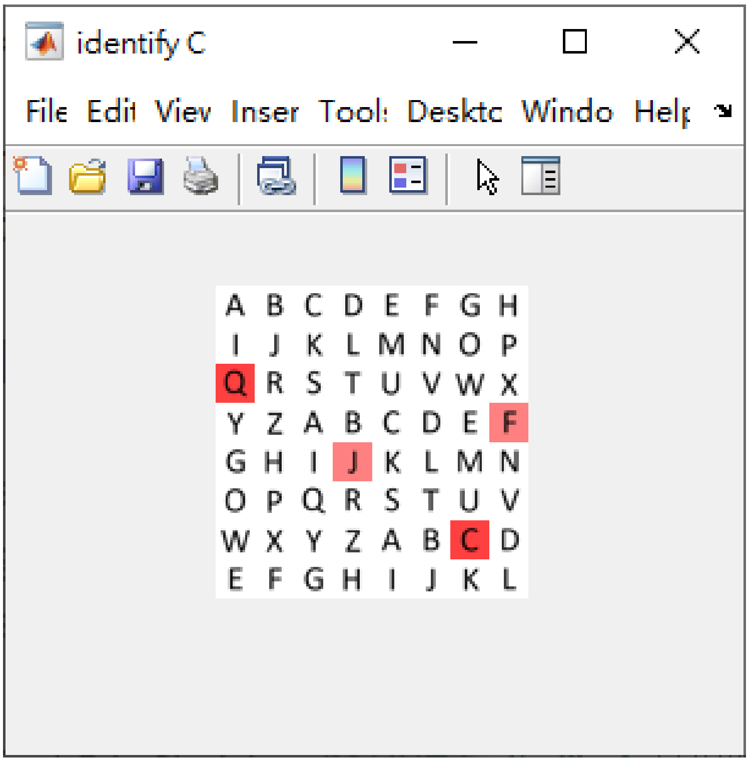
```
    x_axis = (1:bins);
    y_axis = zeros(1,bins);

    for j = init_j:init_j+step-1
        for i = init_i:init_i+step-1
            y_axis(the(i ,j)) = y_axis(the(i ,j)) + mag(i ,j);
        end
    end
end

%將 error 最低的字母上色
function out_img = getMARK(img ,x ,y)
    img(x:x+15 ,y:y+15 ,2) = img(x:x+15 ,y:y+15 ,2)*0.5;
    img(x:x+15 ,y:y+15 ,3) = img(x:x+15 ,y:y+15 ,3)*0.5;

    out_img = img;
end
```

輸出結果:

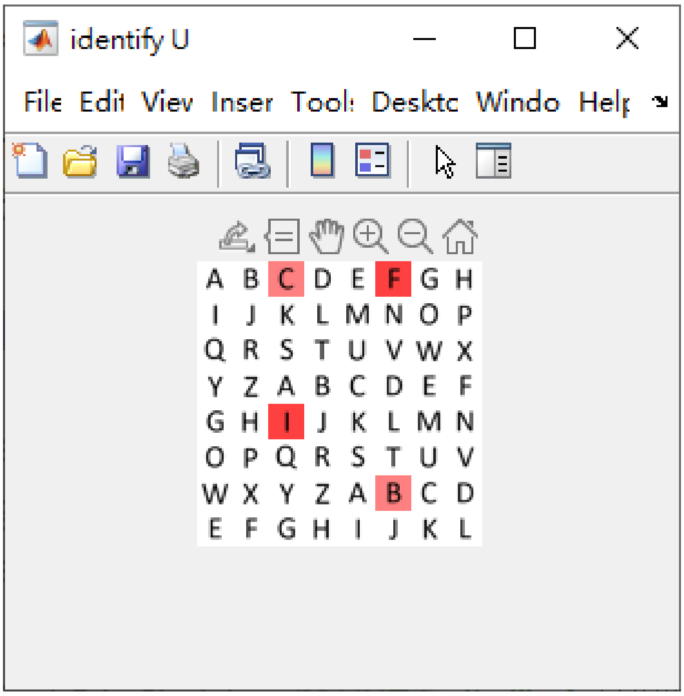


HW3_M11015069_劉宇哲





A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	A	B	C	D	E	F
G	H	I	J	K	L	M	N
O	P	Q	R	S	T	U	V
W	X	Y	Z	A	B	C	D
E	F	G	H	I	J	K	L



A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	A	B	C	D	E	F
G	H	I	J	K	L	M	N
O	P	Q	R	S	T	U	V
W	X	Y	Z	A	B	C	D
E	F	G	H	I	J	K	L

慘痛的事實，在做 ABCXX.jpg 的時候是一個一個 cell 去跑函式，所以沒存下所有的 gradient 方向，所以做 identify 時判斷依據只有 gradient magnitude，根本就不準。

此方法的缺點在於對付有歪斜的、大小不同的很可能會造成畔對結果失準，所以有可能需要額外加一些旋轉平移矩陣來變化圖片，或者是增加可以判斷的資料集，像機器學習般，如果只有每個字母都只有一張來驗證，很可能會失準。