

台科大 110 學年度「彩色影像處理」作業三：利用方向梯度直方圖做英文大寫字母辨識

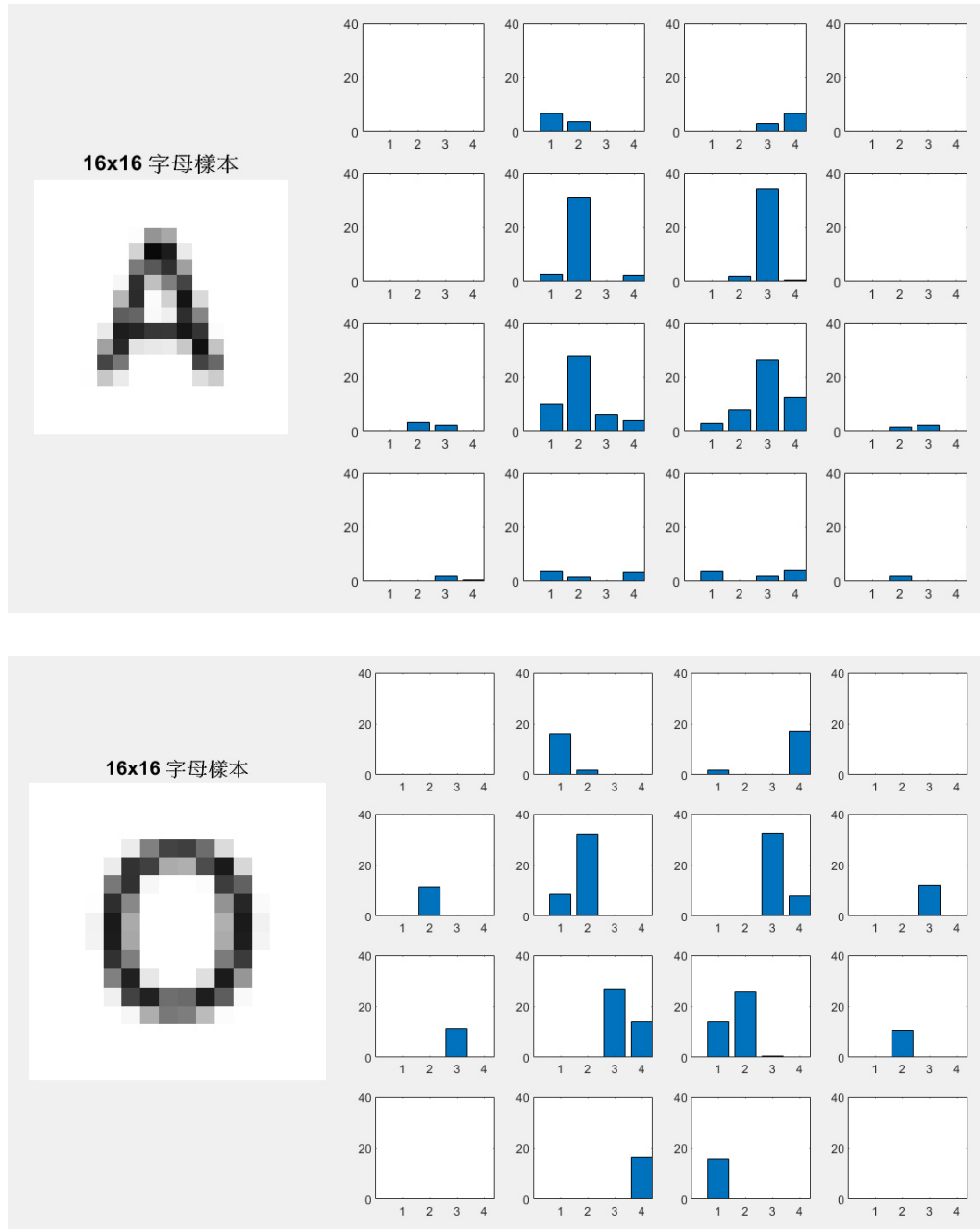
注意事項：

1. 不得使用現成 HOG 相關函式。
2. 程式語言可用 Matlab, Python, C++, Java, VB
3. 報告形式：將程式碼與詳細註解以文字形式貼入 Word 檔，連同執行結果截圖，轉成 PDF 檔。檔案名稱以 HW3_學號命名，例如 HW3_M11001234.pdf。將程式碼、說明 PDF 檔、測試影像壓成 zip 檔。檔案名稱以 HW3_學號命名，例如 HW3_M11001234.zip。
4. 繳交期限：12 月 23 日 24:00 前上傳至 Moodle 作業區。

● 第一部份：輸入任意字母編號，查詢圖 t 中的字母

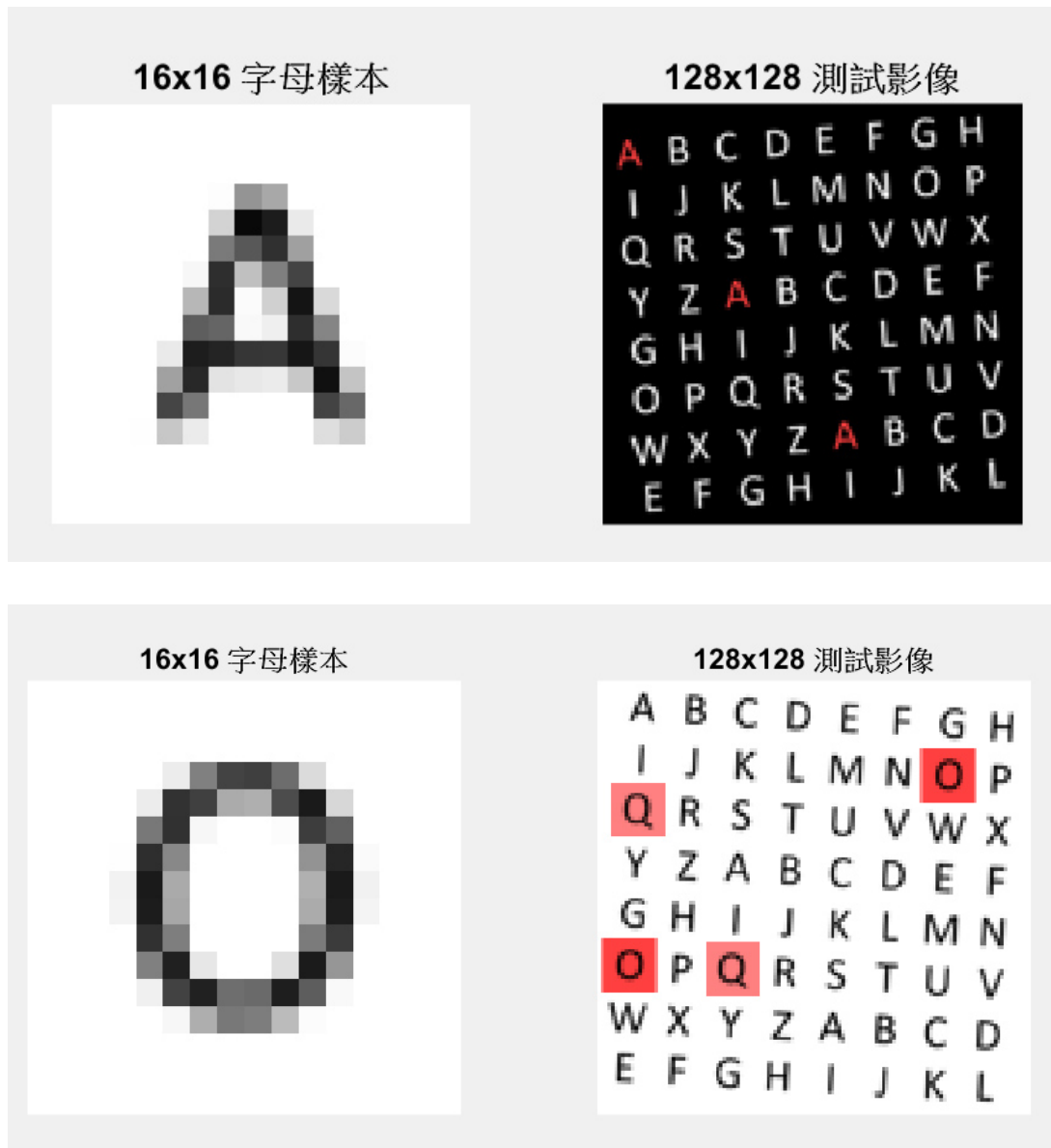
- 1 撰寫產生 4×4 方向梯度直方圖 (Histogram of oriented gradient, HOG) 描述子的程式。
 - 1.1 讀入灰階字母樣本，縮小至 16×16 像素尺寸，將影像分成 4×4 區域，每個區域有 16 (= 4×4) 個像素。算出這 16 個像素的梯度絕對值 M 與方向 α 。
 - 1.2 $M = |G_x| + |G_y|$ 或 $\sqrt{G_x^2 + G_y^2}$ ，其中 G_x 與 G_y 分別代表 Sobel 垂直與水平邊緣檢測的結果(不要做二值化)，注意 `imfilter()` 的邊緣處理要選 'replicate'。
 - 1.3 梯度方向角 α 可用 `atan2d()` 計算， α 超過 [0 180] 度範圍的，用 `mod(α , 180)` 限制其範圍。
 - 1.4 以 45 度為間隔，算出 [0 180] 度內四個區間的梯度直方圖。將 4×4 區域的直方圖一一算出，獲得長度 64 (= 4×4×4) 的向量。
- 2 撰寫主程式
 - 2.1 用 `input()` 讀取字母編號，範圍是 [1 26] 的整數。
 - 2.2 生成檔名字串：以 'A' 為例，檔名是 ABC01.jpg。
 - 2.3 讀取灰階字母樣本，並將樣本影像縮小至 16×16 像素。
 - 2.4 套用前述的 HOG 函式，獲得字母樣本 HOG 向量。

- 2.5 以 4×4 影像陣列，分別繪製每個子區域的方向梯度直方圖。注意 16 個直方圖的垂直尺度(垂直範圍)需一致。答案不一定要跟老師的範例一致，openCV 的 xy 的軸向與 Matlab 相反。



- 2.6 讀取灰階測試影像，檔名是 t1.jpg (可比較更有挑戰性的 t2.jpg 歪斜，t3.jpg 歪斜+反白)。將影像縮小至 128×128 像素。
- 2.7 用迴圈讀取測試影像不同位置的 ROI，ROI 的大小為 16×16 像素。
- 2.8 套用前述的 HOG 函式，獲得 ROI 區域的 HOG 向量。
- 2.9 計算字母樣本與 ROI 區域 HOG 的 L2 距離(誤差的平方和)。

2.10 將 L2 距離最小的前六名 ROI，用顏色標示。以下的作法，是將該 ROI 區域的藍綠兩通道乘上原區塊亮度的 0.5 倍。如果前六名中只選中一次，會呈現淡紅色，若選中一次以上，則呈現鮮紅色。由紅色的鮮豔度可看出該區域 HOG 特徵的強健度。旁邊標示 16×16 像素的字母樣本影像。



● **第二部份：辨識圖 p 中的字母，輸出字串**

1. 將 26 個字母樣本依序讀入，縮小成 16×16 像素之後，分別算出 HOG 特徵向量。
2. 輸入測試影像 p.jpg (例如下圖)，將影像縮小至適當大小。



3. 將字串影像二值化後，再對連通區域分離(可利用現成函式，例如 `bwlabel()`，`regionprops()`)。
4. 由左至右切個成多個字母區塊。
5. 依序算出每個字母區域的 HOG 特徵向量，並跟第一個步驟 26 個字母的 HOG 特徵向量一一匹配，將 L2 距離最小的字母選出。
6. 輸出字串辨識結果。例如：'JDJCLGADOBMI'
7. 用文字說明這種方法有什麼不足之處？如何改善？