

字跡辨識專題報告
B10901163 張顯譽

摘要：

中文筆跡鑑定的目的在於分辨某文件是否由當事人所寫，還是被他人偽造。筆跡鑑定在法務鑑定中扮演了非常重要的角色，應用範圍廣泛，包括信用卡簽單、帳單、遺囑等文件的鑑定。要準確完成筆跡鑑定是一項極為耗費眼力、需要高度細心且費時的工作。其核心在於辨別筆跡之間的差異，以確認是否為同一人所寫。然而，由於同一人的字跡也會有自然變異，這增加了筆跡真偽辨識的難度。



目標：

透過 DSP 方法讀取字跡，並且取以下重要的 feature 送進 SVM，預測字跡是否偽造

Dataset：50 個真跡，50 個偽造

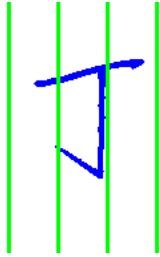

平分成兩堆，一堆當 training (25 個真的、25 個偽造)，另一堆當 testing (25 個真的、25 個偽造)

Input 圖像處理

如何判斷該 pixel 是否為筆畫	
用 python (cv2) 將圖讀進來，並用公式 $y = 0.299R + 0.587G + 0.114B$ If $y < 220 \rightarrow$ stroke (筆畫點) Else background	
 原圖	 輸出的圖

Projection feature

將一張圖切成橫的五等分，和直的五等分，計算每一個區域有多少個 stroke 點，接著每一張圖取平均和標準差，做 normalization，共 10 個 feature(p_1, p_2, \dots, p_{10})
--

 <p>p1: 0 strokes p2: 88 strokes p3: 676 strokes p4: 145 strokes p5: 0 strokes</p>	 <p>p6: 0 strokes p7: 523 strokes p8: 169 strokes p9: 217 strokes p10: 0 strokes</p>
這十個 feature 可以做到 90.22%精準度	

Moment feature

<p>將一張圖座標化，並假設二值化，$B(i,j) = 1$ if stroke else = 0</p> $m_0 = \frac{\sum_{i=0}^m \sum_{j=0}^n m \cdot B(i,j)}{\sum_{i=0}^m \sum_{j=0}^n B(i,j)} \quad n_0 = \frac{\sum_{i=0}^m \sum_{j=0}^n n \cdot B(i,j)}{\sum_{i=0}^m \sum_{j=0}^n B(i,j)}$ $m_{a,b} = \frac{\sum_{i=0}^m \sum_{j=0}^n (m - m_0)^a \cdot (n - n_0)^b \cdot B(i,j)}{\sum_{i=0}^m \sum_{j=0}^n B(i,j)}$ <p>取 $m_0, n_0, m_{2,0}, m_{0,2}, m_{1,1}, m_{3,0}, m_{2,1}, m_{1,2}, m_{0,3}$ 共九個 feature</p>	這九個 feature 可以做到 89.55%精準度
--	----------------------------

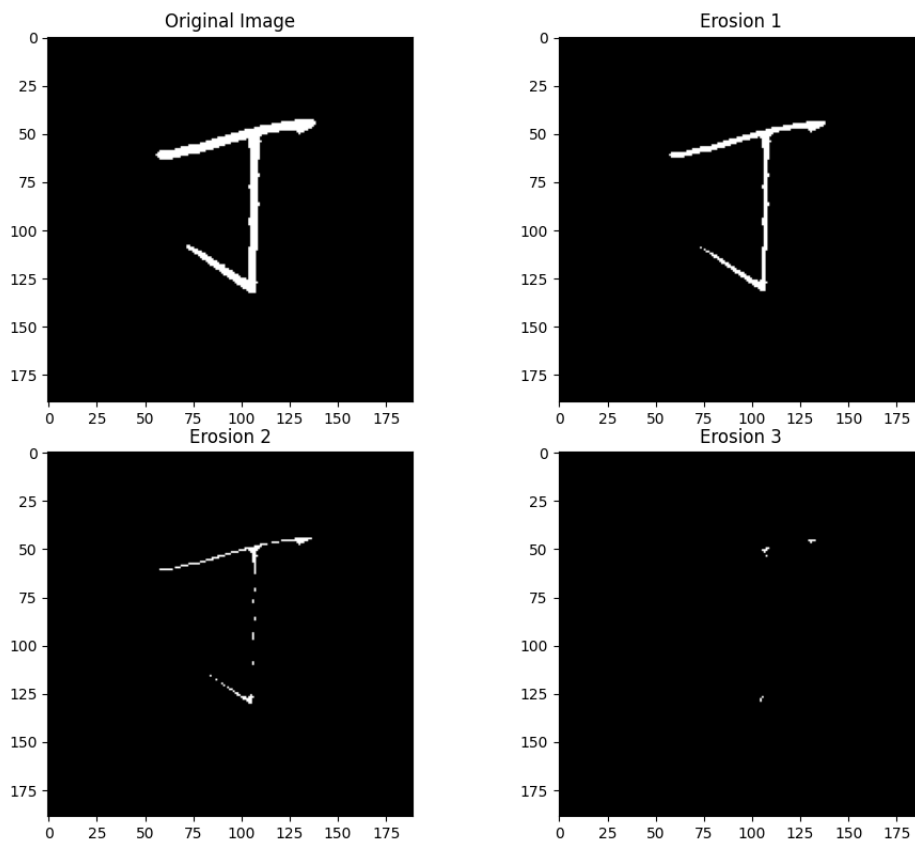
Intensity feature

<p>計算每一個 stroke pixel 的強度的平均及標準差 ($y = 0.299R + 0.587G + 0.114B$)</p> <p>取 mean, std 共兩個 feature</p>	這兩個 feature 可以做到 88.89%精準度
---	----------------------------

Stroke after erosion feature

<p>將一張圖座標化，並假設二值化，$B(i,j) = 1$ if stroke else = 0</p> <p>Initial</p> <p>$Y_0(i,j) = B(i,j)$</p> <p>$Y_k(i,j) = Y_k(i,j) \&\& Y_{k-1}(i-1,j) \&\& Y_{k-1}(i,j-1) \&\& Y_{k-1}(i+1,j) \&\& Y_{k-1}(i,j+1)$</p>	
---	--

令 $r_k = \text{num}(\text{stroke pixel after } k \text{ erosion}) / \text{num}(\text{stroke pixel})$



取 r_1, r_2, r_3 共 3 個 feature

這三個 feature 可以做到 87.33%精準度

Orientation feature

將所有的 stroke pixel 的座標取平均(x_0, y_0)，得一 $p \times 2$ 矩陣 Z

($p = \text{stroke 的數量}$)

每一列是($x_n - x_0, y_n - y_0$)

$$Z = EDE^T$$

$$= \begin{pmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_{11} & e_{21} \\ e_{12} & e_{22} \end{pmatrix}$$

$\lambda_1, \lambda_2 = \text{eigenvalue}$

(e_{12}, e_{21}), (e_{12}, e_{22}) 是兩組 eigenvector

計算這兩組 eigenvector 和橫軸的角度，比較水平的 eigenvector 對應到 eigenvalue 設為 $\lambda_{\text{horizontal}}$ ，另一 eigenvalue 為 $\lambda_{\text{vertical}}$ ，再將比較水平的夾角也送進去 SVM，取 $\theta, \lambda_{\text{horizontal}}, \lambda_{\text{vertical}}$ 共 3 個 feature

這三個 feature 可以做到 76%精準度

在資料庫中，總共有 9 種字（丁、建、均、五、十、伍、拾、務、實）我把字分成筆畫小於八劃的（丁、均、五、十、伍）和大於八劃的（建、拾、務、實），選定適合的 feature 達到最高的 accuracy

實驗結果

Projection feature + Moment feature + Intensity feature + Stroke after erosion feature	
丁	94%
均	96%
五	92%
十	94%
伍	96%
Projection feature + Moment feature + Intensity feature + Stroke after erosion feature+ Orientation feature	
建	90%
拾	92%
務	94%
實	94%

Conclusion

我做出一套針對中文字的字跡真偽辨識演算法。比起其他語言中的文字，中文字擁有比較複雜的結構。因此，演算法使用了幾種不同的 feature，Projection feature, Moment feature, Intensity feature, Stroke after erosion feature+, Orientation feature。我們結合這幾種特徵可以

更完整的描述整個文字。考量到在做真偽辨識時，不同的特徵對於不同的字有不同的重要性，因此我以筆畫多寡去分 feature 重要性。SVM 則會根據組合特徵驗證該手寫文字是否為偽造的文字。實驗結果顯示，我們的演算法在字跡真偽辨識中的準確率達到 93.57%，並明顯優於其他現有的辨識方法。