Pattern Recognition
Term Project Report

Using Visual Words for Image Classification

u9562171, 雷禹恆

1. Abstract

Visual words 近年來在image retrieval領域被大量使用。它是基於文字上的textual words，套用在影像上的類比，因此可將過去在text retrieval領域的技巧直接利用於image retrieval，也有助於large-scale影像搜尋系統的效率。Visual words的擷取大致上是將影像的SIFT features，在keypoint feature space上做K-means clustering的結果，以histogram來表示，是一種bag-of-features。除了可用於retrieval外，visual words也被用於image classification。本專題的目的就是將visual words作為影像特徵，並套用於multi-class image classification。

2. Introduction

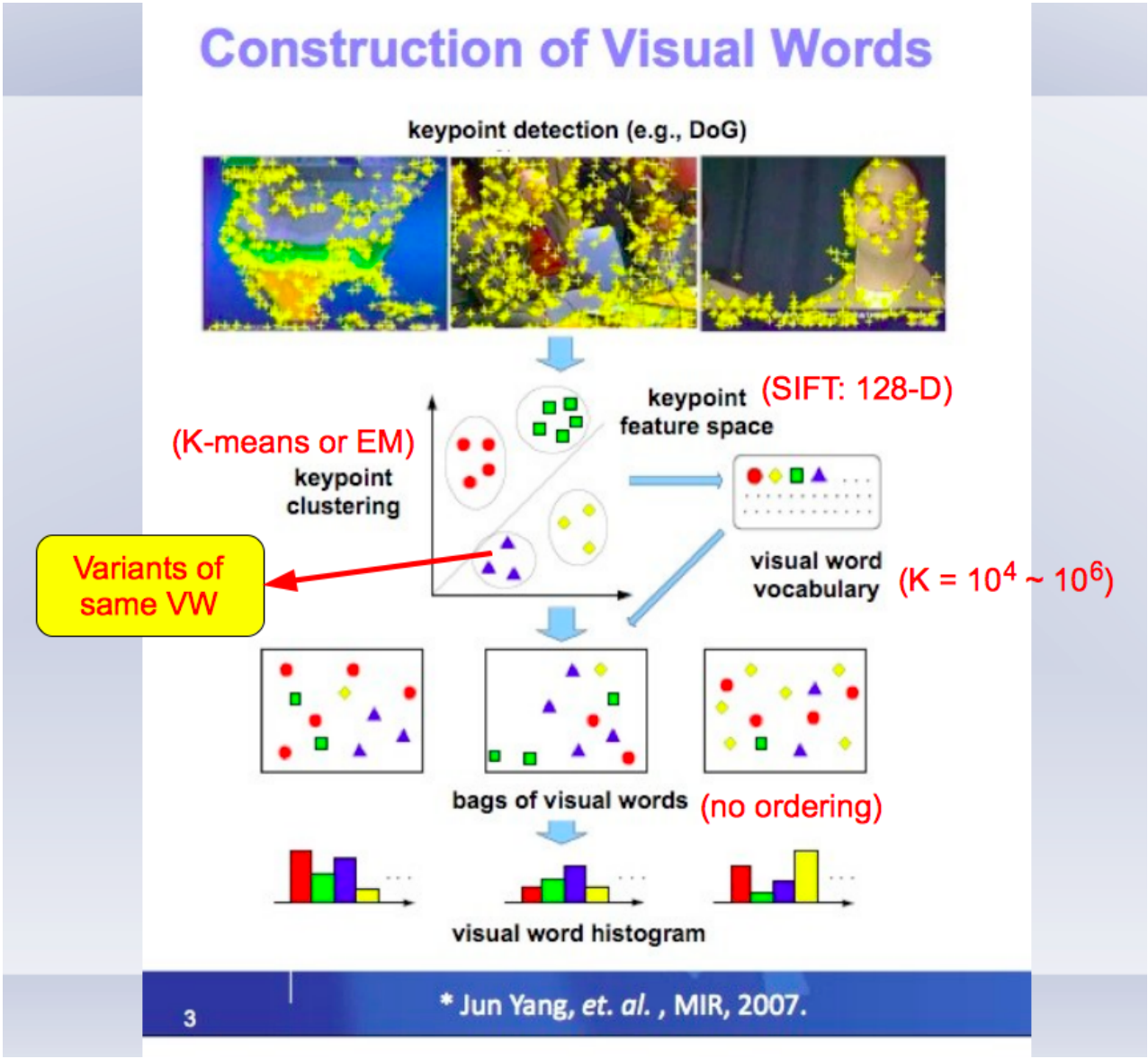
Visual words (簡稱VWs) 近年來在image retrieval領域被大量使用，其motivation其實是從text retrieval領域而來，是基於文字上的textual words，套用在影像上的類比。如同於一篇文章是由許多文字 (textual words) 組合而成，若我們也能將一張影像表示成由許多 “visual words” 組合而成，就能將過去在text retrieval領域的技巧直接利用於image retrieval；而以文字搜尋系統現今的效率，將影像的表示法「文字化」也有助於large-scale影像搜尋系統的效率。

在文獻 [1] 有提到motivation of visual words的細節。首先我們先回顧text retrieval的過程：1. 一篇文章被parse成許多文字，2. 每個文字是由它的「主幹(stem)」來表示的。例如以 ‘walk’ 這個字來說，‘walk’、‘walking’、‘walks’ 等variants同屬於 ‘walk’ 這個主幹，在text retrieval system裡被視為同一個字。3. 排除掉每篇文章都有的極端常見字，例如 ‘the’ 和 ‘an’。4. 一篇文章文章的表示法，即以每個字出現頻率的histogram vector來表示。5. 在此histogram中，對於每個字其實都有給一個某種形式weight，例如Google利用PageRank [2] 的方式來做weighting。6. 在執行文字搜尋時，回傳和此query vector最接近(以角度衡量)的文章。

下一節將解釋影像中VW的特徵擷取流程，及其與此段之1.、2.、4.的類比。

3. Construction of Visual words

Visual words的建構流程可以用圖(1)來說明：



圖(1)：Construction of visual words

步驟**1**：偵測影像中的SIFT keypoints，並計算keypoint descriptors。例如在原始SIFT文獻 [3] 使用Difference of Gaussians (DoG) 來偵測keypoints，而以一個128-D的向量作為descriptor。偵測keypoints的動作相當於上一節所說的1. 將文章parse成一個一個的文字。

步驟**2**：將所有訓練影像的所有keypoint descriptors，散佈於一個128-D的keypoint feature space中，再執行一個clustering algorithm，例如K-means或是這學期教過的EM。在image retrieval領域中，cluster數 K 常訂為$10^4 \sim 10^6$。同一個cluster裡的keypoints，相當於是同一個 “visual word stem” 的variants，在retrieval / classification系統中被視為同一個VW，因此K也被稱為系統裡的 “vocabulary size”。clustering後的結果相當於上一節所說的2. 同一個word stem下有許多的variants。

步驟**3**：最後，一張影像可看成由許多VW（原先是keypoints）組成。因為在retrieval領域，我們並不在意文字的排列順序，只在意文章中文字的出現頻率，同樣

的道理，一張影像中我們只在乎每個VW stem的出現頻率。以這種概念構成的特徵被稱為 “Bag-of-features”，只在乎袋子裡有什麼物品，而不是物品的排列順序。因此影像特徵的表示法為根據VW出現頻率的 “visual word histogram”。這種概念與上一節4.的文字出現頻率histogram相同。

結論：Visual words可看作是將影像中local的keypoint descriptors，套上clustering algorithm後，變成整張影像的global feature。

Visual words 除了大量用於image retrieval外，也可以直接作為features用於image classification。本專題的目的就是將visual words作為影像特徵，並套用於multi-class image classification。

4. Dataset

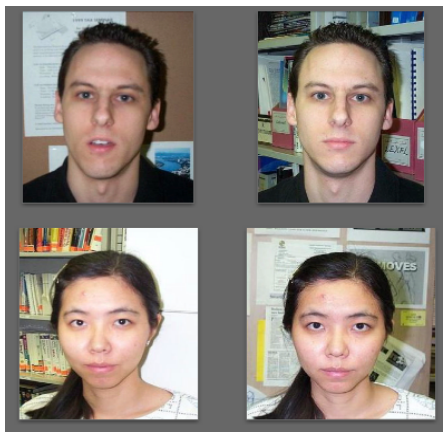
網路上屬於multi-class，每個class大量，但一張影像只有單一物件的dataset沒有想像中得好找。後來選擇了**Caltech 101** [4] 的subset作為本次實驗的dataset。Caltech 101雖然有101種object class，但class裡夠多張影像的只有5種，資料夾名稱(重新命名名稱)分別是：airplanes(Airplane)、BACKGROUND_Google(None)、Faces_easy(Face)、Faces(未使用)、Motorbikes(Motorbike)。Faces和Faces_easy我只選擇切割比較好的Faces_easy來使用。另外我也挑掉不少label有瑕疵的影像，尤其是None類別雖然號稱為背景，裡面卻有很多時候出現人臉。還有極少數張取VW失敗的影像也被剔除。

最後剩下的dataset，我對於4個class分別以training : testing = 9 : 1的方式分配比例。各類的張數統計如表(1)，可看出Airplane和Motorbike的量約為Face和None的兩倍。影像範例如圖(2)，可看出None類別比較困難，因為各式各樣的內容都有，但其他三個類別就簡單許多，物體也少有被干擾的情況。

Label	Class	Total	Training	Testing
1	Airplane	800	720	80
2	Face	435	391	44
3	Motorbike	797	717	80
4	None	368	331	37
	Σ	2400	2159	241

表(1)：dataset 中各類影像的張數統計

(a) Airplane



(b) Face

(c) Motorbike

(d) None

圖(2)：dataset 中 4 類影像的範例

5. Experiment Results and Observations

本次實驗使用MATLAB R2009b (64-bit)，系統環境是Mac OS X 10.6.3，Intel Cuo 2 Duo 2.16GHz processor，3G RAM。分類器必須配合libSVM [5] 的主package與MATLAB interface。詳細使用說明見README.txt。

而使用的features: visual word histograms，是把原始圖片給實驗室學長姊後，請她們用實驗室的tools幫我轉成VW文字檔的，其中偵測keypoints的方式是Hessian Affine (HA)，而非SIFT文獻上的Difference of Gaussians (DoG)。也就是說，feature extraction 的部分是training和testing皆事先完成。Vocabulary size分別有K = 10,000 (10^4)，1,000 (10^3)，及125 (5^3)。

Attempt 1：一開始我只使用作業學過的MATLAB內建SVM來分類。由於只支援 binary classification，我用one-against-one的方式訓練了 C * (C-1) / 2 = 6個分類器，再用majority vote決定output值。使用人造data跑起來還算合理，但套進真實資料：K = 10,000的VW以後，外加手動測試了一些SVM參數，結果卻非常離譜地差。training與testing的confusion matrices如表(2)：

$$\underset{training}{\begin{bmatrix} 720 & 0 & 0 & 0 \\ 0 & 391 & 0 & 0 \\ 0 & 0 & 717 & 0 \\ 0 & 0 & 0 & 331 \end{bmatrix}} \quad \underset{testing}{\begin{bmatrix} 80 & 0 & 0 & 0 \\ 44 & 0 & 0 & 0 \\ 80 & 0 & 0 & 0 \\ 37 & 0 & 0 & 0 \end{bmatrix}}$$

表(2)：Attempt 1 實驗的 confusion matrices

可以看出training意外地完全被分對，testing卻完全被分到class 1(Airplane)。檢查之下發現在SVM train的過程中，所有的點都被歸類為support vectors，也就是本次實驗的結果極端地fit我的training data，對testing data幾乎沒有classification的能力。

Attempt 2：接著我改用libSVM的multi-class classifier來分類，配上其提供的parameter selection tools (grid.py)。實驗的結果好很多，整體accuracy可達90%以上。嘗試了K = 10,000和K = 1,000，其confusion matrix，每個class的recall與precision如表(3)：

$$confusion = \begin{bmatrix} 76 & 0 & 1 & 3 \\ 0 & 43 & 0 & 1 \\ 0 & 0 & 79 & 1 \\ 4 & 1 & 0 & 32 \end{bmatrix}, recall = \begin{bmatrix} 0.95 \\ 0.98 \\ 0.99 \\ 0.86 \end{bmatrix}, precision = \begin{bmatrix} 0.95 \\ 0.98 \\ 0.99 \\ 0.87 \end{bmatrix}$$

(a) K = 10,000：Accuracy = 95.4357% (230/241)

$$confusion = \begin{bmatrix} 76 & 0 & 1 & 3 \\ 0 & 44 & 0 & 0 \\ 0 & 0 & 79 & 1 \\ 4 & 2 & 1 & 30 \end{bmatrix}, recall = \begin{bmatrix} 0.95 \\ 1.00 \\ 0.99 \\ 0.81 \end{bmatrix}, precision = \begin{bmatrix} 0.95 \\ 0.96 \\ 0.98 \\ 0.88 \end{bmatrix}$$

(b) K = 1,000：Accuracy = 95.0207% (229/241)

表(3)：Attempt 2 的實驗數據

Attempt 2的結果遠優於Attempt 1，我認為可能是MATLAB內建的SVM能力無法應付本次的高維度dataset，或是只用人工測試少數SVM參數遠比用tools的效果差。另外從表(3)可看出，class 4 (None) 的recall和precision皆最差，符合預期，因為此類別的影像內容什麼都有，本身就比較困難，其他3個類別相對簡單很多，所以數據都蠻不錯的。本報告之後實驗結果，各個class的表現也符合這樣的現象。

Attempt 3：雖然Attempt 2的結果已經很好，dataset仍然存在著一個問題：我的dataset實際的dimensionality (K)是否真的有1,000、10,000那麼高？因此我接著多使用PCA (Principal Component Analysis) 來做轉換和降維，因為K = 10,000的PCA超出我自己電腦的運算能力，Attempt 3只測試K = 1,000。目標維度我利用保留多少百分比的total variance來決定，分別測試100%、99%、98%、97%、96%、95%。實驗數據如表(4)：

Total Variance	Reduced Dimension	Overall Accuracy	Confusion (C), Recall (R), Precision (P)
100%	1,000	95.0207% (229/241)	$C=\begin{bmatrix}76&0&1&3\\0&44&0&0\\0&0&79&1\\4&2&1&30\end{bmatrix}, R=\begin{bmatrix}0.95\\1.00\\0.99\\0.81\end{bmatrix}, P=\begin{bmatrix}0.95\\0.96\\0.98\\0.88\end{bmatrix}$
99%	348	95.0207% (229/241)	$C=\begin{bmatrix}75&0&1&4\\0&44&0&0\\0&0&79&1\\4&2&0&31\end{bmatrix}, R=\begin{bmatrix}0.94\\1.00\\0.99\\0.84\end{bmatrix}, P=\begin{bmatrix}0.95\\0.96\\0.99\\0.86\end{bmatrix}$
98%	195	95.8506% (231/241)	$C=\begin{bmatrix}77&0&1&2\\0&44&0&0\\0&0&80&0\\4&1&2&30\end{bmatrix}, R=\begin{bmatrix}0.96\\1.00\\1.00\\0.81\end{bmatrix}, P=\begin{bmatrix}0.95\\0.98\\0.96\\0.94\end{bmatrix}$
97%	118	95.4357% (230/241)	$C=\begin{bmatrix}77&0&1&2\\0&44&0&0\\0&1&79&0\\4&1&2&30\end{bmatrix}, R=\begin{bmatrix}0.96\\1.00\\0.99\\0.81\end{bmatrix}, P=\begin{bmatrix}0.95\\0.96\\0.96\\0.94\end{bmatrix}$
96%	75	95.4357% (230/241)	$C=\begin{bmatrix}76&0&1&3\\0&44&0&0\\0&0&79&1\\4&1&1&31\end{bmatrix}, R=\begin{bmatrix}0.95\\1.00\\0.99\\0.84\end{bmatrix}, P=\begin{bmatrix}0.95\\0.98\\0.98\\0.89\end{bmatrix}$
95%	49	94.1909% (227/241)	$C=\begin{bmatrix}77&0&1&2\\0&44&0&0\\0&0&79&1\\6&2&2&27\end{bmatrix}, R=\begin{bmatrix}0.96\\1.00\\0.99\\0.73\end{bmatrix}, P=\begin{bmatrix}0.93\\0.96\\0.96\\0.90\end{bmatrix}$

表(4)：Attempt 3 的實驗數據 (K = 1,000 + PCA)

total variance = 100%其實和Attempt 2的K = 1,000是一樣的結果。從表(4)可發現，total variance降到95%，也就是維度一路降到49時，都有差不多水準的表現。後來測更低的90% (13維)，accuracy就只剩下88.80%了。

總之，經過PCA之後的accuracy和Attempt 2比沒有增加，但是降維後的執行速度，不論是training或testing，皆比Attempt 2快很多。

Attempt 4：由之前PCA的經驗，我推斷此dataset最適合的vocabulary size (K)可能比設定的1,000、10,000小很多，介於K = 100 ~ 200之間。因此我再用一份 K = 125 (5^3) 的dataset跑一次實驗，只做SVM parameter selection，不做PCA，發現也能得到很好的結果，如表(5)：

$$confusion = \begin{bmatrix} 75 & 0 & 1 & 4 \\ 0 & 44 & 0 & 0 \\ 0 & 0 & 80 & 0 \\ 4 & 2 & 2 & 29 \end{bmatrix}, recall = \begin{bmatrix} 0.94 \\ 1.00 \\ 1.00 \\ 0.79 \end{bmatrix}, precision = \begin{bmatrix} 0.95 \\ 0.96 \\ 0.96 \\ 0.88 \end{bmatrix}$$

Accuracy = 94.6058% (228/241)

表(5)：Attempt 4 的實驗數據 (K = 125, no PCA)

因此，本次visual word dataset的實際vocabulary size，可能比retrieval常用的 $10^4 \sim 10^6$ 低很多。

6. Discussions

在實驗過程中，有幾個步驟面臨到需要決定參數問題，分別是：

1. SVM parameters (C, γ)
2. PCA target dimension (% of total variance retained)
3. Vocabulary size (K in K-means)

針對1.，因為有現成的libSVM parameter selection tools，所以只要讓電腦去選擇即可。針對2.，我的原則是，在能達到同樣classification水準下，選擇維度最低的。實驗發現在K = 1,000的dataset中，以95% of total variance降到49維都還有很好的表現。至於針對3.，由於我對實驗室擷取VW的tools不熟悉，資料都是事先請學長姊擷取好的，沒有辦法真的去嘗試每一種K，只用了K = 10,000、1,000、125進行實驗，算是一個缺點。

我在後期有想到一點，文獻 [1] 提到在text retrieval會把 “very common words”，例如 ‘a’、‘the’、‘is’、‘and’，從database剔除，或是想辦法降低它們的weights。這點在我的dataset裡並沒有做到，因為時間關係一時也想不到簡單又無缺點的方法。這或許是accuracy無法超越96%的原因之一。

最後，本次專題其實訂的問題算簡單，還有一些挑戰等待克服。例如本次的dataset: Caltech 101，我挑出來的4類有3類是將物體切割出來，容易分類的。未來可嘗試更困難的dataset，例如Caltech 256 [6]，再組合其他網路上的dataset讓object classes更多。除此之外，在object detection的問題中，快速localization或recognition [7] 也是很重要的。物體不會永遠佔滿影像的大半面積，以不同大小的sliding window尋找物體位置是很沒效率的做法；而類別增加時，一個一個去詢問SVM的output也不是很有效率。這些都是未來努力的方向。

7. Project hosting

本專題以open source的形式托管於Google Code，網址如下：

http://code.google.com/p/search

網站上除了有本次專題的程式碼、dataset、程式執行的demo影片、說明文件、本書面報告、投影片之外，未來的研究如屬於可公開的部分，也會繼續在專案網站上更新。

8. References

[1] Sivic, Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos,” *Proc. IEEE International Conference on Computer Vision, 2003*.

[2] Brin, Page, “The anatomy of a large-scale hypertextual web search engine,” *7th Int. WWW Conference, 1998*.

[3] Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] Caltech 101 dataset, http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[5] C. C. Chang(張治中) and C. J. Lin(林智仁), “LIBSVM -- A Library for Support Vector Machines,” http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[6] Caltech 256 dataset, http://www.vision.caltech.edu/Image_Datasets/Caltech256/

[7] Yeh, Lee, Darrell, “Fast concurrent object localization and recognition,” *IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.