ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

LÊ THANH TÂM

PHÂN LOẠI ẢNH DỰA TRÊN HƯỚNG TIẾP CẬN KERNEL

LUẬN VĂN THẠC SĨ
NGÀNH KHOA HỌC MÁY TÍNH

Thành phố Hồ Chí Minh - 2011

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

LÊ THANH TÂM

PHÂN LOẠI ẢNH DỰA TRÊN HƯỚNG TIẾP CẬN KERNEL

Ngành: KHOA HỌC MÁY TÍNH
Mã số: 60.48.01

LUẬN VĂN THẠC SĨ
(Chuyên ngành Tin học)

NGƯỜI HƯỚNG DẪN KHOA HỌC:
PGS. TS. NGUYỄN ĐÌNH THÚC
TS. TRẦN THÁI SƠN

Thành phố Hồ Chí Minh - 2011

LỜI CẢM ƠN

Trước tiên, tôi xin chân thành cảm ơn PGS.TS. Nguyễn Đình Thúc và TS. Trần Thái Sơn đã hướng dẫn tận tình cho tôi trong suốt thời gian thực hiện luận văn.

Tôi xin cảm ơn GS. Akihiro Sugimoto (National Institute of Informatics, Tokyo, Japan) và TS. Yousun Kang (National Institute of Informatics, Tokyo, Japan) đã chỉ dẫn và cho tôi những góp ý quý báu về nội dung luận văn trong thời gian thực tập 6 tháng ở Viện Tin học Quốc gia Nhật Bản (National Institute of Informatics, Tokyo, Japan).

Tôi xin cảm ơn GS. Seiichi Mita (Toyota Technological Institutue, Nagoya, Japan) đã tận tình hỗ trợ, hướng dẫn và giúp tôi có những kinh nghiệm thực tiễn trong quá trình thực tập 3 tháng ở Học viện Kỹ thuật Toyota, Nagoya, Nhật Bản (Toyota Technological Institute, Nagoya, Japan).

Tôi xin cảm ơn GS. D. McAllister (Toyota Technological Institute, Chicago, USA) và GS. L. El Ghaoui (University of California, Bekerley, USA) đã tận tình giảng dạy cho tôi những nền tảng cơ bản về máy học, tối ưu và thị giác máy tính.

Tôi xin cảm ơn ThS. Trần Lê Hồng Dũ và nghiên cứu sinh M. Kloft (University of California, Bekerley, USA) đã trao đổi, thảo luận và truyền đạt những kinh nghiệm quý báu trong quá trình thực nghiệm đề tài.

Tôi cũng xin gởi lời cảm ơn quý thầy cô, anh chị và bạn bè trong khoa Công nghệ thông tin, Trường Đại Học Khoa Học Tự Nhiên TP.HCM, những người đã giúp đỡ cũng như cung cấp cho tôi những kiến thức, kinh nghiệm.

Con xin cảm ơn ba mẹ và gia đình luôn yêu thương, hỗ trợ con trong suốt thời gian học tập, giúp con có thêm tự tin để thực hiện tốt công việc.

Xin chân thành cảm ơn!

Người thực hiện

Lê Thanh Tâm

MỤC LỤC

Danh mục các kí hiệu và chữ viết tắt

BoF	Bag of feature
C2F	Coarse to fine
MKL	Multiple Kernel Learning
HSMK	Hierarchical Spatial Matching Kernel
PMK	Pyramid Matching Kernel
SPM	Spatial Pyramid Matching
SPMK	Spatial Pyramid Matching Kernel
SVM	Support Vector Machine

Danh mục các bảng

Danh mục các hình vẽ, đồ thị

MỞ ĐẦU

Với sự bùng nổ của dữ liệu ảnh, việc phân loại các ảnh ra thành các lớp ngữ nghĩa là một trong những nhu cầu cơ bản cho việc quản lý và truy vấn ảnh dựa trên nội dung của ảnh. Thêm nữa, phân loại ảnh là một trong những bài toán cơ bản trong lĩnh vực thị giác máy tính và ứng dụng máy học và nhận được sự quan tâm của nhiều nhà khoa học trên thế giới. Bài toán phân loại ảnh có rất nhiều thách thức từ việc ảnh được chụp dưới nhiều góc độ khác nhau, điều kiện chiếu sáng khác nhau, sự đa dạng các thể hiện của cùng một lớp ngữ nghĩa cũng như sự phức tạp của thông tin nền trong ảnh. Để giải quyết bài toán phân loại ảnh thì có hai hướng tiếp cận chính là dựa trên đặc trưng hoặc dựa trên phương pháp học. Trong đó, hướng tiếp cận dựa trên phương pháp học mà đặc biệt là nhánh tiếp cận dựa trên phương pháp kernel là một trong những phương pháp được áp dụng rất rộng rãi và mang lại kết quả cao trong bài toán phân loại ảnh nói riêng và trong lĩnh vực thị giác máy tính nói chung, do tính mềm dẻo khi mô tả ảnh trong những điều kiện phức tạp như trên. Do vậy, trong luận văn này, tôi đề xuất kernel mới, đặt tên là Hierarchical Spatial Matching Kernel (HSMK) và áp dụng cho bài toán phân loại ảnh. HSMK là mô hình cải tiến từ mô hình Spatial Pyramid Maching (SPM), nhưng thay vì sử dụng mô hình Bag-of-Feature (BoF) để mô hình cho các vùng con (subregions), HSMK sử dụng mô hình thô mịn (coarse to fine – C2F) cho các vùng con mà được hiện thực hóa bằng phương pháp multiresolution (tạm dịch nhiều loại phân giải), tức xem xét vùng con trên một chuỗi các độ phân giải (resolution) khác nhau, do vậy, nó có thể miêu tả được thông tin tổng quát của vùng con từ những độ phân giải thô, cũng như những thông tin chi tiết của vùng con ở những độ phân giải mịn hơn như cách thức xem xét một vùng trên bản đồ, để có thể đạt được độ đo tương đồng tốt hơn trên các vùng con này. Từ thí nghiệm cho thấy, kernel đề xuất - HSMK cho hiệu quả rất tốt cho bài toán phân loại ảnh và đạt được kết quả tối ưu (state-of-the-art) trên nhiều cơ sở dữ liệu chuẩn cho bài toán phân loại ảnh.

Chương 1 Giới thiệu

1.1 Mục tiêu

Trong luận văn này, tôi nghiên cứu việc xây dựng kernel cho thuật toán phân lớp trong lĩnh vực máy học, cụ thể là thuật toán phân lớp Support Vector Machine (SVM). SVM thực hiện việc phân lớp bằng cách tìm siêu phẳng (hyperplane) mà cho phép cực đại hóa khoảng cách biên (maximize margins). Trong khi đó, kernel của SVM dùng để đo độ tương đồng giữa các mẫu học, việc này đóng góp lớn vào hiệu quả phân lớp của thuật toán SVM. Thêm nữa, SVM là thuật toán phân lớp hiệu quả và được sử dụng rất rộng rãi trong nhiều lĩnh vực, đặc biệt trong lĩnh vực thị giác máy tính. Từ kernel tuyến tính (linear kernel) mà sử dụng hàm tương quan (correlation), hay tích nội (inner product) để tính độ tương đồng trong việc phân chia lớp ở thời gian đầu khi thuật toán SVM được đề xuất. Các nhà nghiên cứu nhận thấy rằng, dữ liệu ngày càng phong phú và đa dạng, việc này đòi hỏi cần phải sử dụng các kernel phi tuyến (non-linear kernel) để có thể tìm được siêu phẳng hiệu quả hơn. Do vậy, nghiên cứu xây dựng kernel là một trong những chủ đề được nhiều nhà nghiên cứu trên thế giới quan tâm.

Để đánh giá sự hiệu quả của kernel đề xuất, tôi áp dụng kernel đề xuất vào bài toán phân loại ảnh trong lĩnh vực thị giác máy tính. Trong đó, bài toán phân loại đối tượng và phân loại cảnh là hai thể hiện cụ thể của bài toán phân loại ảnh được thực nghiệm dựa trên việc áp dụng kernel đề xuất để phân lớp.

1.2 Đóng góp của luận văn

1.2.1 Xây dựng kernel cho thuật toán SVM

Luận văn đề xuất Hierarchical Spatial Matching Kernel (HSMK), tạm dịch kernel so khớp có tính không gian và phân cấp. HSMK là sự cải tiến của Sptial Pyramid Matching Kernel – SPMK (tạm dịch kernel so khớp dạng tháp) dựa trên mô hình thô mịn (coarse to fine – C2F). SPMK được đề xuất bởi Lazebnik và các

đồng sự [19] thực hiện việc chia ảnh trên một chuỗi các lưới có kích thước khác nhau thành các vùng con (subregions), sau đó áp dụng mô hình túi đặc trưng (Bag of features – BoF) [6] để mô hình cho các vùng con này. Kernel đề xuất - HSMK cũng thực hiện việc chia ảnh dựa trên một chuỗi các lưới có kích thước khác nhau như trong SPMK, nhưng thay vì sử dụng mô hình BoF mà được biết hạn chế trong việc mô hình vùng để có thể đo được độ tương đồng tối ưu, HSMK sử dụng mô hình C2F để có thể xem xét vùng trên nhiều kích cỡ khác nhau, việc này có thể cho phép HSMK đạt được sự xấp xỉ độ tương đồng tối ưu tốt hơn khi sử dụng BoF như trong SPMK.

HSMK được tôi và các đồng sự công bố trong bài báo "Hiearchical Spatial Matching Kernel for Image Categorization" ở hội nghị quốc tế về phân tích và nhận dạng ảnh (International Conference on Image Analysis and Recognition – ICIAR) ở Burnaby, British Columbia, Canada vào năm 2011.

1.2.2 Áp dụng kernel xây dựng cho bài toán phân loại ảnh

Để cho thấy sự hiệu quả của kernel đề xuất - HSMK, tôi áp dụng vào bài toán phân loại ảnh thông qua hai thể hiện là bài toán phân loại đối tượng và phân loại cảnh. Từ thực nghiệm trên nhiều cơ sở dữ liệu ảnh chuẩn (benchmark dataset) cho bài toán phân loại đối tượng như Oxford Flower, CALTECH-101, CALTECH-256, cũng như cho bài toán phân loại cảnh như MIT Scene, UIUC Scene. HSMK cho kết quả vượt trội so với SPMK, với lưu ý rằng, SPMK được biết như kernel tốt nhất được dùng mô hình đối tượng cho việc tính toán độ tương đồng trong nhiều bài toán của lĩnh vực thị giác máy tính, đặc biệt là bài toán phân loại ảnh.

Thêm nữa, việc sử dụng kernel đề xuất - HSMK cũng cho kết quả cao nhất (state of the art) hoặc ngang với các cách tiếp cận khác trên các cơ sở dữ liệu chuẩn này. Mặt khác, hướng tiếp cận sử dụng HSMK chỉ sử dụng một kernel phi tuyến với SVM trên một loại đặc trưng, trong khi các phương pháp đạt kết quả cao nhất khác trên các cơ sở dữ liệu chuẩn trên thường sử dụng trên nhiều loại đặc trưng, cũng như sử dụng các phương pháp học phức tạp như học với nhiều kernel (multiple

kernel learning – MKL), tối ưu tuyến tính kết hợp boosting (linear programming boosting – LP-B).

1.3 Các đóng góp khác liên quan

Luận văn không trình bày tất cả các đóng góp được công bố của tôi trong thời gian là một học viên cao học. Trong phần này, tôi trình bày tóm tắt đóng góp khác liên quan đến hướng của luận văn – về máy học và thị giác máy tính.

Tôi đề xuất thuật toán phân đoạn (segmentation) màu cho ảnh biển báo giao thông dựa trên thuật toán phân lớp SVM. Thay vì xử lý trên từng điểm ảnh (pixel) như cách tiếp cận truyền thống, thuật toán đề xuất xử lý trên một vùng các điểm ảnh để có thể sử dụng các thông tin lân cận, nâng cao hiệu quả phân đoạn màu trong ảnh giao thông. Thuật toán này được áp dụng vào việc phát hiện biển báo giao thông cho hệ thống lái xe tự động trong đề án "Hệ thống lái xe tự động" (Autonomous driving system) của Học Viện Công Nghệ Toyota, Nagoya, Nhật Bản (Toyota Technological Institute, Nagoya, Japan). Công trình này được công bố trong bài báo "Realtime Traffic Sign Detection Using Color and Shape-Based Features" ở hội nghị lần hai về hệ thống cơ sở dữ liệu và hệ thống thông tin thông minh (Asian Conference on Intelligent Information and Database Systems – ACIIDS) ở Huế, Việt Nam, 2010.

1.4 Cấu trúc của luận văn

Trong chương 2, tôi trình bày khái quát nền tảng lý thuyết của thuật toán phân lớp dựa trên Support Vector Machine (SVM), từ SVM truyền thống với việc học dựa trên một kernel tới dạng học nhiều kernel của SVM, hay được biết với tên gọi bài toán Multiple Kernel Learning (MKL) cũng như lý thuyết về kernel được sử dụng trong SVM cũng như trong MKL. Tiếp đó, trong chương 3, tôi trình bày phương pháp học dựa trên kernel mà được xem là một trong những hướng tiếp cận chính và hiệu quả cho bài toán phân loại ảnh và trong chương 4, tôi trình bày kernel mà luận văn đề xuất - Hiearchical Spatial Matching Kernel (HSMK). Cuối cùng,

chương 5 trình bày việc áp dụng HSMK vào bài toán phân loại ảnh mà cụ thể là bài toán phân loại đối tượng và bài toán phân loại cảnh trên những cơ sở dữ liệu chuẩn như: Oxford Flower, CALTECH-101, CALTECH-256, MIT Scene và UIUC Scene.

Chương 2 Thuật toán phân lớp dựa trên SVM

Trong chương này, tôi trình bày khái quát lý thuyết phân lớp của thuật toán Support Vector Machine (SVM). Tôi cũng nhắc lại lý thuyết kernel áp dụng cho thuật toán SVM trong chương này. Cuối cùng là một hướng nghiên cứu đang được cộng đồng nghiên cứu máy học rất quan tâm là việc học với nhiều kernel cho SVM, hay được biết với tên gọi bài toán Multiple Kernel Learning (MKL).

2.1 Học với một kernel – Support Vector Machine (SVM)

2.1.1 Thuật toán phân lớp SVM

Thuật toán phân lớp SVM được đề xuất bởi Cortes và Vapnik vào năm 1995 [3]. Nhưng những ý tưởng chính của thuật toán phân lớp SVM bắt nguồn từ hai công trình của Vapnik và Lerner vào năm 1963 [31] và Vapnik và Chervonenkis vào năm 1964 [32].

Thuật toán SVM là một bộ phân lớp nhị phân được xây dựng cho một tập dữ liệu huấn luyện như sau:

Gọi $X = \{x_1, x_2, ..., x_N\}$ với $x_i \in \mathbb{R}^n$ là tập dữ liệu nhập và $Y = \{y_1, y_2, ..., y_N\}$ tương ứng là tập dữ liệu xuất, hay còn gọi là nhãn của các mẫu dữ liệu nhập với $y_i \in \{-1, +1\}$. $D_{train} = (X, Y)$ được gọi là tập dữ liệu huấn luyện cho thuật toán phân lớp SVM.

Bộ phân lớp tuyến tính được mô hình như sau:

$$y(x) = sign(w^T x + b) \tag{2.1}$$

Với $w \in \mathbb{R}^n$ là vector trọng số và $b \in \mathbb{R}$. Khi đó, ta có ràng buộc dữ liệu cho thuật toán học SVM như sau:

$$\begin{cases} w^T x_k + b \geq +1 & y_k = +1 \\ w^T x_k + b \leq -1 & y_k = -1 \end{cases} \tag{2.2}$$

Ta có thể kết hợp tập điều kiện (2.2) thành:

$$y_k (w^T x_k + b) \geq 1 \qquad k = 1, \ldots, N \tag{2.3}$$

Với điều kiện ràng buộc như trong (2.3), với các tập dữ liệu không thể phân tách được trên tất cả các mẫu học thì lời giải cho thuật toán phân lớp SVM là rỗng, điều này rất dễ xảy ra trong thực tế, do dữ liệu huấn luyện luôn có nhiễu. Để giải quyết cho trường hợp này, Cortes và Vapnik [3] đã thay đổi công thức (2.3) thành:

$$y_k(w^T x_k + b) \geq 1 - \xi_k \qquad k = 1, \ldots, N \tag{2.4}$$

Với biến slack $\xi_k > 0$ để giải quyết cho trường hợp một số mẫu trong tập dữ liệu huấn luyện vi phạm điều kiện phân lớp. Ta có thể thấy những mẫu có $\xi_k > 1$ là những mẫu vi phạm điều kiện phân lớp so với ràng buộc trong (2.3).

Công thức tối ưu dạng nguyên thủy (primal problem) theo không gian trọng số của SVM có dạng như sau:

$$\begin{aligned} &\min_{w,b,\xi} J_P(w,\xi) = \frac{1}{2} w^T w + C\sum_{k=1}^{N} \xi_k \\ &s.t. \qquad y_k(w^T x_k + b) \geq 1 - \xi_k, \qquad k = 1, \ldots, N \\ &\qquad\quad \xi_k \geq 0, \qquad k = 1, \ldots, N \end{aligned} \tag{2.5}$$

Với C là một số nguyên dương, được sử dụng để điều khiển giữa việc tối ưu hàm mục tiêu và những mẫu vi phạm ràng buộc phân lớp của SVM trong (2.3).

Từ (2.5), ta có biểu thức Lagrangian tương ứng là:

$$L(w,b,\xi;\alpha,v) = J_p(w,\xi) - \sum_{k=1}^{N} \alpha_k (y_k(w^T x_k + b) - 1 + \xi_k) - \sum_{k=1}^{N} v_k \xi_k \tag{2.6}$$

Với các hệ số Lagrangian $\alpha_k \geq 0, v_k \geq 0$ với $k = 1, \ldots, N$. Từ biểu thức (2.6), ta có lời giải của vấn đề tương ứng với việc giải bài toán:

$$\max_{\alpha,v} \min_{w,b,\xi} L(w,b,\xi;\alpha,v) \tag{2.7}$$

Lấy đạo hàm từng phần cho mỗi biến của hàm Lagrangian L trong (2.6), ta có:

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{k=1}^{N} \alpha_k y_k x_k \\ \dfrac{\partial L}{\partial b} = 0 \quad \rightarrow \quad \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \dfrac{\partial L}{\partial \xi_k} = 0 \quad \rightarrow \quad 0 \le \alpha_k \le C, k = 1,...,N \end{cases} \tag{2.8}$$

Thay (2.8) vào (2.6) ta có bài toán đối ngẫu dạng tối ưu bậc hai (Dual Quadratic Programming) cho bài toán SVM như sau:

$$\begin{aligned} &\max_{\alpha} J_D(\alpha) = -\frac{1}{2} \sum_{k,l=1}^{N} y_k y_l x_k^T x_l \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k \\ &s.t. \qquad \sum_{k=1}^{N} \alpha_k y_k = 0 \\ &\qquad\qquad 0 \le \alpha_k \le C, \quad k = 1,...,N \end{aligned} \tag{2.9}$$

Do biểu thức (2.9) là dạng bài toán tối ưu bậc hai (Quadratic Programming), do vậy có thể sử dụng các bộ giải tối ưu (optimization solvers) để tìm lời giải.

2.1.2 Kernel trong thuật toán phân lớp SVM

2.1.2.1 Đo độ tương đồng sử dụng kernel

Để mở rộng khả năng phân lớp của thuật toán SVM, thay vì sử dùng hàm tích nội (inner product) để đo độ tương đồng giữa 2 mẫu x_i, x_j trong không gian dữ liệu nhập huấn luyện, khái niệm kernel được đưa ra.

Đầu tiên, dữ liệu nhập sẽ được chuyển sang không gian H bằng hàm ánh xạ như sau:

$$\Phi : X \rightarrow H, \qquad x \mapsto \Phi(x) \tag{2.10}$$

Để tính độ tương đồng giữa các mẫu học trong H, ta có thể sử dụng hàm tích nội tương ứng trong không gian H, ký hiệu $\langle .,. \rangle_H$. Để tiện lợi, ta định nghĩa hàm tương ứng như sau:

$$k : X \times X \rightarrow \mathbb{R}, \qquad (x, x') \mapsto k(x, x') \tag{2.11}$$

mà thỏa điều kiện:

$$k(x,x') = \left\langle \Phi(x), \Phi(x') \right\rangle_H , \qquad \forall x, x' \in X \tag{2.12}$$

Hàm như trong (2.12) được gọi là hàm kernel.

2.1.2.2 Kernel xác định dương (Positive Definite Kernel)

Hàm được định nghĩa như (2.12) thuộc lớp kernel xác định dương (Positive Definite Kernel). Điều này cho phép thuật toán SVM, khi tính tích nội có thể sử dụng bất kỳ hàm kernel xác định dương để thay thế cho $\left\langle \Phi(x), \Phi(x') \right\rangle_H$ khi tính toán cho kernel $k(x,x')$. Kỹ thuật này được biết với tên gọi mẹo kernel (kernel trick). Điều này dẫn tới với hàm kernel xác định dương, ta không cần biết dạng tường minh của dạng hàm chuyển không gian từ không gian dữ liệu nhập vào không gian H, mà điều này đã được định nghĩa không tường minh thông qua hàm kernel.

Để làm rõ hàm kernel xác định dương, tôi nhắc lại một số định nghĩa sau:

Định nghĩa 1 (ma trận Gram)

Cho một kernel $K : X \times X \to \mathbb{R}$ và dãy dữ liệu $x_1, ..., x_n \in X$. Ta gọi ma trận K có chiều $n \times n$ có chứa các phần tử như sau:

$$K_{ij} = k(x_i, x_j) \tag{2.13}$$

là ma trận Gram hay ma trận kernel k cho dãy dữ liệu $x_1, ..., x_n$.

Định nghĩa 2 (ma trận xác định dương)

Ma trận đối xứng các số thực có chiều $n \times n$ được gọi là xác định dương khi và chỉ khi với $\forall c_1, ..., c_n \in \mathbb{R}$, ta có:

$$\sum_{i,j=1}^{n} c_i c_j K_{ij} \geq 0 \tag{2.14}$$

Với dấu bằng trong (2.14) xảy ra khi $c_1 = ... = c_n = 0$, khi đó ma trận được gọi là xác định dương ngặt (strictly positive definite).

Định nghĩa 3 (Kernel xác định dương)

Nếu $\forall n \in \mathbb{N}$ và $\forall x_1, ..., x_n \in X$, ma trận Gram $K_{ij} = k(x_i, x_j)$ là xác định dương thì ta gọi kernel là kernel xác định dương.

Trong phương pháp học SVM với kernel, ta có định đề quan trọng sau:

Định đề kernel

Một hàm $k : X \times X \to \mathbb{R}$ là một kernel xác định dương khi và chỉ khi tồn tại một không gian Hilbert H và một hàm ánh xạ $\Phi : X \to H$ thỏa điều kiện $\forall x, x' \in X$, ta có $k(x, x') = \langle \Phi(x), \Phi(x') \rangle_H$.

Chứng minh

"$\Leftarrow$": Giả sử kernel được viết dưới dạng (2.12), ta có:

$$\sum_{i,j=1}^{n} c_i c_j \langle \Phi(x_i), \Phi(x_j) \rangle_H = \left\langle \sum_{i=1}^{n} c_i \Phi(x_i), \sum_{j=1}^{n} c_j \Phi(x_j) \right\rangle_H = \left\| \sum_{i=1}^{n} c_i \Phi(x_i) \right\|_H^2 \geq 0. \quad (2.15)$$

"$\Rightarrow$" được trình bày trong 2.1.2.3, tức xây dựng không gian Hilbert và hàm ánh xạ Φ cũng như các tính chất mong muốn từ kernel xác định dương.

2.1.2.3 Xây dựng không gian tái sinh kernel Hibert (Reproducting Kernel Hibert Space – RKHS)

Trong phần này, tôi trình bày cách xây dựng không gian Hilbert mà mỗi phần tử của không gian là một hàm kernel xác định dương.

Cho kernel k, ta thành lập tập hợp F như sau:

$$F = \left\{ f(.) = \sum_{i=1}^{n} \alpha_i k(., x_i); \quad n \in \mathbb{N}, \alpha_i \in \mathbb{R}, x_i \in X \right\} \subseteq \mathbb{R}^X \quad (2.16)$$

Với $k(., x) : X \to \mathbb{R}$ là hàm và cũng là một phần tử trong F. Ta thấy rằng tập hợp F trên sẽ tạo thành một không gian vector nếu ta gắn với hai phép toán cộng $(f + g)(x) = f(x) + g(x)$ và phép nhân với số thực $(\lambda f)(x) = \lambda f(x), \lambda \in \mathbb{R}$.

Ta định nghĩa phép tích nội của hai phần tử trong không gian này, cho:

$$f(.) = \sum_{i=1}^{n} \alpha_i k(., x_i) \qquad g(.) = \sum_{j=1}^{n'} \beta_j k(., x'_j) \quad (2.17)$$

Với $n, n' \in \mathbb{N}, \alpha_i, \beta_j \in \mathbb{R}, x_i, x'_j \in X$, thì tích nội có dạng như sau:

$$\langle f, g \rangle_F := \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \beta_j k(x_i, x'_j) \quad (2.18)$$

Với ghi chú rằng, chúng ta có thể sử dụng tính chất đối xứng của kernel để viết lại như sau:

$$\sum_{j=1}^{n'} \beta_j f(x'_j) = \langle f, g \rangle_F = \sum_{i=1}^{n} \alpha_i g(x_i) \tag{2.19}$$

Từ tính chất xác định dương của kernel k, ta có:

$$\langle f, f \rangle_F = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \tag{2.20}$$

Từ biểu thức (2.20), ta suy ra rằng với mọi hàm $f_1, ..., f_p \in F$ và mọi hệ số $c_1, ..., c_p \in \mathbb{R}$, ta có:

$$\sum_{i,j=1}^{p} c_i c_j \langle f_i, f_j \rangle_F = \left\langle \sum_{i=1}^{p} c_i f_i, \sum_{j=1}^{p} c_j f_j \right\rangle_F \geq 0 \tag{2.21}$$

Do vậy, $\langle .,. \rangle_F$ là kernel xác định dương trong không gian vector của tập hàm F.

Thêm nữa, khi $g(.) = k(., x)$ thì theo định nghĩa tích nội trong (2.18), ta có:

$$\langle f, k(., x) \rangle_F = \sum_{i=1}^{n} \alpha_i k(x_i, x) = f(x), \qquad \forall x \in X \tag{2.22}$$

Tương tự (2.22), ta có trường hợp đặc biệt như sau:

$$\langle k(., x), k(., x') \rangle_F = k(x, x') \tag{2.23}$$

Tính chất này được biết với tên gọi tính chất tái sinh (reproducing property) của kernel. Tức một hàm f có thể được biểu diễn như một hàm tuyến tính được định nghĩa bằng tích nội trong không gian vector của tập hàm F (như biểu thức (2.22)).

Để chứng minh tính xác định (definiteness property) của tích nội, tôi nhắc lại bất đẳng thức Cauchy-Schwarz:

Định đề: bất đẳng thức Cauchy-Schwarz

Nếu k là kernel xác định dương và $x_1, x_2 \in X$ thì ta có:

$$k(x_1, x_2)^2 \leq k(x_1, x_1) k(x_2, x_2) \tag{2.24}$$

Chứng minh

Do k là kernel xác định dương nên ma trận Gram $K_{ij} = k(x_i, x_j)$ với kích cỡ 2×2 cũng xác định dương, hay các trị riêng (eigenvalues) của ma trận Gram là không âm, dẫn đến định thức ma trận Gram $\det(K) \geq 0$. Khai triển det(K) ta có điều phải chứng minh.

$$0 \leq \det(K) = k(x_1, x_1)k(x_2, x_2) - k(x_1, x_2)^2 \tag{2.25}$$

Từ bất đẳng thức Cauchy-Schwarz trong (2.24) và tính chất tái sinh (reproducing property) trong (2.22), ta có:

$$\left|f(x)\right|^2 = \left|\left\langle k(.,x), f\right\rangle_F\right|^2 \leq k(x,x).\left\langle f, f\right\rangle_F \tag{2.26}$$

Điều này chứng minh rằng: $\left\langle f, f\right\rangle_F = 0 \Leftrightarrow f = 0$.

Do tính chất (2.22), nên không gian vector được thành lập trên tập F được gọi là không gian tái sinh kernel Hilbert (reproducing kernel Hibert space – RKHS) từ kernel xác định dương k. Thêm nữa, RKHS xác định duy nhất kernel xác định dương k và ngược lại, điều này được trình bày trong định lý Moore-Aronszajn [1].

Định lý Moore-Aronszajn [1]

Đối với mỗi kernel xác định dương k, thì tồn tại duy nhất không gian tái sinh kernel Hibert H mà kernel của nó là k và ngược lại.

Chi tiết hơn về việc xây dựng không gian tái sinh kernel Hilbert xin tham khảo [1].

2.2 Học với nhiều kernel – Multiple Kernel Learning (MKL)

Để giải quyết cho vấn đề chọn kernel như thế nào cho một bài toán cụ thể trong thực tế, trong [18], Lanckriet và các đồng sự đã đề xuất việc học SVM với nhiều kernel. Hai ý tưởng chính của bài toán Multiple Kernel Learning (MKL) là: (i) tham số hóa hàm kernel là sự kết hợp tuyến tính của nhiều kernel xác định dương, (ii) hệ số của việc kết hợp tuyến tính các kernel được tối ưu hóa trong quá trình huấn luyện sử dụng thuật toán mở rộng của SVM.

Gọi η là số lượng kernel dùng cho việc học SVM, các kernel xác định dương dùng cho việc học tương ứng là k_1, k_2, …k_η, thì kết hợp tuyến tính kernel cho việc học SVM được biểu diễn là:

$$\begin{aligned} k(.,.) &= \sum_{\alpha=1}^{\eta} \theta_\alpha k_\alpha(.,.) \\ s.t. \quad & \sum_{\alpha=1}^{\eta} \theta_\alpha = 1, \theta_\alpha \geq 0 \qquad \forall \alpha = 1,...,\eta \end{aligned} \tag{2.27}$$

Với θ_α là hệ số kết hợp tuyến tính của kernel xác định dương $k_\alpha(.,.)$.

Dạng nguyên thủy (Primal) cho bài toán học MKL như sau:

$$\begin{aligned} & \min_{w'_\alpha, w_0, \xi, \theta} \frac{1}{2}\left(\sum_{\alpha=1}^{\eta} \theta_\alpha \left\| w'_\alpha \right\|_2\right)^2 + C\sum_{i=1}^{N} \xi_i \\ s.t. \quad & y_i\left(\sum_{\alpha=1}^{\eta} \theta_\alpha \left\langle w'_\alpha, \Phi_\alpha(x_i) \right\rangle + w_0\right) \geq 1 - \xi_i \\ & \sum_{\alpha=1}^{\eta} \theta_\alpha = 1, \theta \geq 0, \xi \geq 0, \end{aligned} \tag{2.28}$$

Với (x_i, y_i) là dữ liệu huấn luyện, $\Phi_\alpha(x_i)$ là hàm chuyển dữ liệu nhập x_i của kernel $k_\alpha(.,.)$ vào không gian RKHS tương ứng. Trong công thức (2.28), do ta có lượng nhân giữa hai biến primal là θ_α và w'_α nên công thức là không lồi. Bằng cách đặt $w_\alpha = \theta_\alpha w'_\alpha$ ta sẽ chuyển về được dạng bài toán tìm cực tiểu của hàm lồi (convex minimization problem) [2].

Để giải bài toán MKL, các nhà nghiên cứu máy học đã mô hình hóa bài toán về nhiều dạng khác nhau, trong đó hai thuật giải: SILP của Sonnenburg [30] và SimpleMKL của Rakotomamonjy [28] mà sử dụng SVM truyền thống như một phần của thuật giải được sử dụng rộng rãi hơn cả.

2.2.1 SILP

Trong [30], Sonnenburg đã chuyển (2.28) về dạng bài toán SILP (semi-infinitive linear program), hay là bài toán tối ưu tuyến tính hàm mục tiêu với số lượng ràng buộc là vô tận. Dạng SILP của bài toán MKL được Sonnenburg đưa ra như sau:

$$\begin{aligned} \max_{\gamma,\theta} \quad & \gamma \\ s.t. \quad & \gamma \in \mathbb{R} \\ & \sum_{p=1}^{\eta} \theta_p = 1, \theta_p \geq 0 \qquad \forall p = 1,...,\eta \\ & \sum_{p=1}^{\eta} \theta_p S_p(\alpha) \geq \gamma, \qquad \forall \alpha \in Z \\ & Z = \left\{ \alpha \in \mathbb{R}^n \middle| 0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0 \right\} \end{aligned} \tag{2.29}$$

Với hàm $S_p(\alpha)$ được định nghĩa như sau:

$$S_p(\alpha) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k_p(x_i, x_j) - \sum_{i=1}^{n} \alpha_i \tag{2.30}$$

Đây là bài toán tối ưu tuyến tính (linear program) do γ, θ chỉ được ràng buộc bởi điều kiện tuyến tính. Nhưng do $\alpha \in Z$ nên ta có vô hạn điều kiện ràng buộc cần được thỏa khi tìm lời giải tối ưu.

Thuật toán SILP cho MKL

Nhập: Tham số C>0 (parameter regularization), tập kernel, tập dữ liệu huấn luyện.

Xuất: Các tham số α, b, θ

1: Khởi tạo trọng số cho các kernel: $\theta_p \leftarrow \frac{1}{\eta}, \forall p = 1,...,\eta$

2: $(\alpha^0, b^0) \leftarrow$ Từ việc giải SVM với θ

3: $t \leftarrow 0$ (gắn t = 0, t dùng để xác định số vòng lặp của thuật toán SILP)

4: **while** (Điều kiện dừng chưa thỏa) **do**

5: $(\theta^t, \gamma^t) \leftarrow$ Từ việc giải (2.29) với tập ràng buộc $\{\alpha^0, ..., \alpha^t\}$

6: $\alpha^{t+1} \leftarrow$ Từ việc giải $\alpha^* = \arg\max_{\alpha \in Z} \quad \gamma \quad -\sum_{p=1}^{\eta} \theta_p S_p(\alpha)$ với γ, θ

7: **if** $\sum_{p=1}^{\eta} \theta_p^t S_p(\alpha^{t+1}) \geq \gamma^t$ **then**

8: **break**

9: **end if**

10: $t \leftarrow t+1$

11: **end while**

Chi tiết về thuật toán SILP, xin tham khảo thêm trong [30]. Thuật toán SILP được cài đặt và công bố trong Shogun Toolbox ở địa chỉ: http://www.shogun-toolbox.org

2.2.2 SimpleMKL

Trong [28], Rakotomamonjy đã chuyển (2.28) về dạng tối ưu như sau:

$$\min_{\theta} \quad g(\theta) = \min\{\frac{1}{2}\sum_{p=1}^{\eta}\frac{1}{\theta_p}\|v_p\|^2 + C\sum_{i=1}^{n} L(y_i, \sum_{p=1}^{\eta}\langle v_p, \Phi_p(x_i)\rangle_{H_p} + b)\}$$
$$\{v|v_p=\theta_p w_p\},b$$

$$s.t. \quad \sum_{\alpha=1}^{\eta}\theta_\alpha = 1, \theta \geq 0 \tag{2.31}$$

Rakotomamonjy sử dụng phương pháp giải gradient descent (tạm dịch là giảm dần theo hướng xác định bởi hướng đạo hàm của hàm mục tiêu theo biến tối ưu) để giải cho dạng tối ưu MKL được miêu tả trong (2.31). Để có thể thực hiện, Rakotomamonjy đưa ra công thức tính đạo hàm của tham số SVM theo vector trọng số của kernel như sau:

$$\frac{\partial g}{\partial \theta_p} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i^*\alpha_j^* y_i y_j k_p(x_i, x_j) \tag{2.32}$$

Với (α^*, b^*) là điểm cực đại của hàm g, người ta cũng chứng minh được rằng các điểm này thỏa:

$$\frac{\partial \alpha}{\partial \theta_p}(\alpha^*) = 0 \qquad \frac{\partial b}{\partial \theta_p}(b^*) = 0 \tag{2.33}$$

Thuật toán SimpleMKL cho MKL

Nhập: Tham số C>0 (parameter regularization), tập kernel, tập dữ liệu huấn luyện.

Xuất: Các tham số α, b, θ

1: Khởi tạo trọng số cho các kernel: $\theta_p \leftarrow \frac{1}{\eta}, \forall p = 1,...,\eta$

2: **while** (Điều kiện dừng không thỏa) **do**

3: $(\alpha, b) \leftarrow$ Từ việc giải SVM với kernel $k(.,.) = \sum_{p=1}^{\eta} \theta_p k_p(.,.)$

4: Cập nhật trong số θ, sử dụng bước gradient trong thuật toán gradient descent.

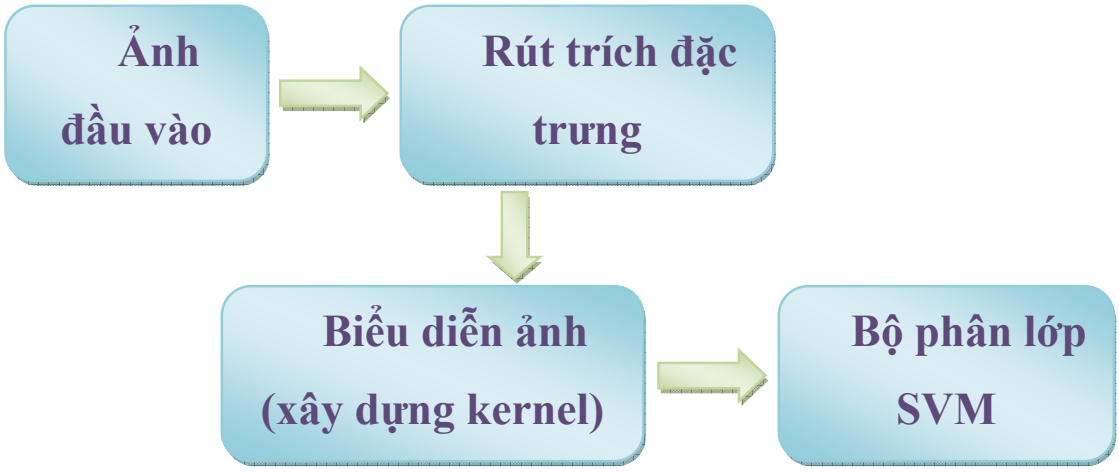
5: **end while**

Chi tiết về thuật toán SimpleMKL, xin tham khảo thêm trong [28]. Thuật toán SimpleMKL được cài đặt và công bố trong SimpleMKL Toolbox ở địa chỉ: http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html.

Chương 3 Phương pháp kernel

Trong các hướng tiếp cận cho bài toán phân loại ảnh, phương pháp kernel là một trong những phương pháp hiệu quả và được áp dụng phổ biến. Phương pháp kernel sử dụng hàm phi tuyến (non-linear) để tính độ tương đồng của hai mẫu thay vì sử dụng tích nội (inner product) – hàm tuyến tính (linear) để cực đại khoảng cách biên (maximize margine) thông qua việc giải bài toán tối ưu bậc hai (quadratic optimization), tiêu biểu là thuật toán học giám sát Support Vector Machines (SVM).

Phương pháp kernel thường bao gồm các bước sau: từ ảnh đầu vào, thực hiện rút trích các đặc trưng, có thể là đặc trưng cục bộ hay đặc trưng toàn cục, tập hợp những đặc trưng này đại diện cho ảnh, kế tiếp tập đặc trưng sẽ được biểu diễn theo một phương pháp biểu diễn ảnh thích hợp để có thể áp dụng hàm tính độ tương đồng, kết quả này được đưa vào bộ phân lớp SVM để xác định mẫu thuộc phân lớp nào như được tóm tắt trong Hình 1.



Hình 1: Mô hình tổng quát cho phương pháp kernel

Trong đó, luận văn tập trung nghiên cứu phần biểu diễn ảnh (xây dựng kernel) cho bài toán phân loại ảnh.

3.1 Mô hình túi đặc trưng (Bag-of-feature model – BoF)

Để xây dựng hàm tính toán độ tương đồng giữa hai ảnh được biểu diễn dưới dạng tập các đặc trưng, mô hình túi đặc trưng BoF [6] mượn ý tưởng mô hình túi từ (Bag-of-word model) từ lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing) trong bài toán tìm chủ đề của văn bản. Mỗi ảnh sẽ tương ứng là mỗi văn bản, các đặc trưng trong ảnh tương ứng với các từ trong văn bản, và chủ đề của văn bản tương ứng là lớp ngữ nghĩa hay nhãn cho ảnh cần được phân loại.

Mô hình BoF được thực hiện gồm hai pha chính: xây dựng từ điển từ các đặc trưng, sử dụng từ điển để mô hình ảnh cho việc tính toán độ tương đồng.

Xây dựng từ điển (codeword dictionary): ý tưởng chính là lượng tử hóa (quantized) các đặc trưng về không gian nhỏ hơn để giảm nhiễu. Trong bước này, toàn bộ các đặc trưng của tất cả các ảnh trong tập dữ liệu học được chọn, ta đặt tên là tập đặc trưng của tập học D_{train} (để cho tiện lợi về mặt thời gian và phù hợp kích cỡ bộ nhớ, thường người ta sẽ chọn tập con D_{set} được chọn ngẫu nhiên từ D_{train}), sau đó thuật toán phân cụm không giám sát Kmeans (unsupervised learning Kmeans clustering algorithm) được áp dụng, để phân tập đặc trưng về N_w cụm, mỗi từ vựng (codeword) trong bộ từ điển (codebook hay dictionary) được đỉnh nghĩa là các tâm (centroid) của cụm thu được từ thuật toán Kmeans. Tập hợp tất cả các từ vựng tạo thành bộ từ điển cho mô hình BoF.

Từ tập đặc trưng đại diện cho mỗi ảnh, với mỗi đặc trưng, ta tìm một từ vựng tương ứng trong bộ từ điển. Bước này ta tính khoảng cách từ mỗi đặc trưng đến mỗi từ vựng trong bộ từ điển, đặc trưng có khoảng cách ngắn nhất đến từ vựng nào, thì đặc trưng sẽ tương ứng với từ vựng đó.Thông thường thì khoảng cách Euclid được sử dụng trong bước này.

Mô hình toán cho việc ánh xạ đặc trưng thành từ vựng như sau: gọi D là từ điển gồm N_w từ vựng D = {W_1, W_2, …, W_{Nw}} và $\Im$ là hàm tính khoảng cách được sử dụng, mỗi đặc trưng x_f sẽ tương ứng với từ vựng có chỉ số là:

$$id_{x_f} = \arg\min_{D} \Im(x_f, W_i) \tag{3.1}$$

Như vậy, từ tập đặc trưng của ảnh, ta chuyển thành tập những từ vựng đại diện cho ảnh. Sau đó, ta lấy histogram của từ vựng – ghi nhận tổng số lần xuất hiện của các từ vựng trong ảnh.

Từ thực nghiệm cho thấy để đạt được độ tương đồng giữa hai ảnh tốt, thì phải áp dụng với kernel phi tuyến như intersection kernel hoặc χ^2 (chi-square) kernel trong phân lớp sử dụng thuật toán SVM.

3.2 Các cải tiến của mô hình BoF

Nhiều phương pháp nghiên cứu gần đây được đề xuất để cải tiến mô hình BoF truyền thống. Hướng tiếp cận sử dụng mô hình phát sinh (generative model) [4][7] để mô hình sự đồng hiện của các từ vựng, hoặc thay vì sử dụng Kmeans để lượng tử hóa xây dựng từ vựng cho từ điển, trong [24][35], các tác giả thực hiện việc học để tìm bộ từ vựng cho độ phân biệt cao giữa các lớp ngữ nghĩa để nâng cao hiệu quả phân lớp. Mặc khác, hướng tiếp cận sử dụng biểu diễn thưa (sparse coding) thay cho Kmeans để xây dựng bộ từ điển cũng đạt được nhiều thành công như trong trong các công bố [22][36]. Biểu diễn thưa có cách xây dựng bộ từ điển tương tự với Kmeans, đều thực hiện việc giải bài toán tối ưu (optimization), nhưng biểu diễn thưa sử dụng ràng buộc mềm hơn so với Kmeans, do vậy sẽ nhận ít lỗi hơn khi thực hiện việc tái tạo lại đặc trưng ban đầu (error reconstruction), cũng như đạt được bộ từ vựng tốt hơn, được trình bày chi tiết trong phần 3.3.

Một trong những điểm yếu chính của mô hình BoF là bỏ qua thông tin không gian của đặc trưng cục bộ trong ảnh (spatial information), để khắc phục điều này, Lazebnik và các cộng sự [18] đã đề xuất mô hình tháp không gian (spatial pyramid kernel), một cách mở rộng của mô hình BoF, SPM sử dụng một chuỗi các lưới có kích thước khác nhau để chia ảnh thành các vùng con (subregion) và sau đó sử dụng mô hình BoF để thống kê tổng hợp (aggregated statistics) đặc trưng cục bộ trên các vùng con (subregions) cố định thay vì chỉ sử dụng trên toàn ảnh như trong mô hình BoF cổ điển, cuối cùng tập hợp các mô hình BoF trên các vùng con được nối lại theo thứ tự được định nghĩa trước để mô hình cho ảnh.

Trong các cải tiến từ mô hình BoF thì SPM mang lại hiệu quả cao và đơn giản khi thực hiện. Do vậy, SPM được sử dụng như một thành phần chính trong nhiều hệ thống đạt kết quả tốt nhất (state-of-the-art) trong lĩnh vực phân loại ảnh [12].

Cũng như mô hình BoF, thì SPM sẽ mang lại hiệu quả tốt nhất khi được sử dụng với kernel phi tuyến như intersection kernel hoặc χ^2 (chi-square) kernel. Những kernel phi tuyến này có độ phức tạp tính toán cao cũng như không gian lưu trữ lớn so với phương pháp tuyến tính. Để giải quyết vấn đề này, Maji và các đồng sự [23] đưa ra một phương pháp tính toán xấp xỉ để nâng cao hiệu quả xây dựng histogram intersection kernel, giảm độ phức tạp tính toán, nhưng hiệu quả trên chỉ đạt được bằng cách sử dụng bảng phụ được tính toán trước, mà được xem như một loại tính toán trước cho huấn luyện SVM phi tuyến. Để xử lý cho dữ liệu lớn, Yang và các đồng sự [36] đưa ra mô hình tuyến tính SPM với biểu diễn thưa (spare coding) (ScSPM) trong đó tích nội (kernel tuyến tính) được sử dụng thay vì kernel phi tuyến dựa trên tính chất tuyến tính của dữ liệu thưa. Wang & Wang [34] đề xuất mô hình học trên nhiều kích cỡ (multiscale learning - MSL) bằng cách sử dụng multiple kernel learning (MKL) để xác định các hệ số cho mô hình SPM thay vì sử dụng hệ số được xác định trước của mô hình SPM nguyên thủy.

Trong luận văn này, tôi đề xuất hàm kernel mới dựa trên hướng tiếp cận của mô hình thô mịn (coarse to fine – C2F) cho các vùng con (subregion) trong mô hình SPM, và đặt tên là Hierarchical Spatial Matching Kernel (HSMK). Mô hình C2F giúp cho vùng con được xem xét ở nhiều mức độ khác nhau, có thể hình tượng như khi xem bản đồ, ở mức thô cho phép quan sát toàn cảnh, thêm nữa, ở mức mịn thì cho phép quan sát các chi tiết. Do vậy, HSMK không chỉ giúp mô tả thông tin thứ tự không gian của đặc trưng cục bộ mà còn có thể đo chính xác độ tương đồng giữa các tập hợp của đặc trưng cục bộ không thứ tự lấy từ các vùng con. Trong HSMK, việc áp dụng mô hình C2F trên các vùng con được hiện thực hóa bằng cách sử dụng nhiều độ phân giải (multi-resolution). Do vậy, đặc trưng cục bộ có thể miêu tả thông tin chi tiết của ảnh hoặc đối tượng từ vùng con ở độ phân giải mịn (fine resolution) và cả thông tin toàn cục của vùng con ở độ phân giải thô hơn. Thêm nữa, việc so

khớp dựa trên mô hình C2F là quá trình phân cấp (hierarchical), điều này có nghĩa là đặc trưng mà không tìm được sự so khớp ở độ phân giải mịn có khả năng được so khớp ở độ phân giải thô hơn. Như vậy, kernel được đề xuất có thể đạt được sự xấp xỉ so khớp tối ưu (optimal matching) tốt hơn giữa các vùng con so với SPM. Tóm lại, HSMK chú trọng vào việc cải thiện độ đo tương đồng giữa các vùng con bằng cách sử dụng mô hình C2F, được hiện thực hóa bằng cách sử dụng nhiều độ phân giải (multi-resolution), thay vì sử dụng mô hình BoF trên các vùng con như trong SPM. Việc xem xét vùng con bằng cách sử dụng một chuỗi các độ phân giải (resolution) khác nhau tương tự như trong kernel so khớp dạng tháp (pyramid matching kernel) [13], nhưng thay vì sử dụng vector trọng số được định nghĩa trước cho các intersection kernel cơ bản trên các vùng con cho việc kết hợp trên nhiều độ phân giải (resolution) khác nhau, tôi chuyển bài toán về dạng học trên nhiều kernel có phân bố đồng nhất (uniform multiple kernel learning – uniform MKL) để tìm vector trọng số hiệu quả hơn. Ưu điểm của HSMK là nó có thể được dùng trên tập hợp các đặc trưng không thứ tự có số phần tử khác nhau bằng cách áp dụng chuẩn hóa căn bậc hai theo đường chéo (square root diagonal normalization) [28] cho các intersection kernel cơ bản trên vùng con mà điều này không được xem xét trong PMK [13].

3.3 Phương pháp biểu diễn thưa (Sparse Coding)

Gọi $X = \{x_1, x_2, \ldots, x_M\}$ là tập hợp các đặc trưng, với x_i thuộc không gian $\mathbb{R}^d$. Lượng tử hóa (quantization) bằng cách áp dụng K-means có thể mô hình như sau:

$$\min_{V} \sum_{m=1}^{M} \min_{k=1..K} \left\| x_m - v_k \right\|^2 \tag{3.2}$$

Với $V = \{v_1, v_2, \ldots, v_K\}$ là tập hợp K cluster tìm được bởi phương pháp Kmeans (hay còn gọi là codebook – hay từ điển) và $\| \cdot \|$ là L2-norm của vector. Hay ta có thể diễn đạt là với vector x_m ta tìm vector v_k (codeword – từ vựng) tương ứng trong từ điển bằng cách tìm vector v_k sao cho khoảng cách từ x_m tới v_k là ngắn nhất (thông thường thì hàm Euclide được sử dụng để tính khoảng cách). Ta có thể chuyển công

thức tối ưu (optimization) (3.2) về bài toán phân tích ma trận (matrix factorization) như sau:

Ta gọi u_j là vector xác định từ vựng của x_j trong từ điển (codebook), với $Card(u_j) = 1$, tức x_j thuộc về từ vựng thứ p thì thành phần thứ p trong u_j bằng 1, còn các thành phần còn lại bằng 0, và tập $U = \{u_1, u_2, \ldots, u_M\}$ là tập hợp xác định từ vựng của vector đặc trưng tương ứng với tập X. Do vậy, ta có thể đưa công thức K-means về dạng sau:

$$\min_{U,V} \sum_{m=1}^{M} \left\| x_m - u_m V \right\|^2 \tag{3.3}$$

Thỏa: $$Card\left(u_m\right) = 1, \left|u_m\right| = 1, u_m \geq 0, \forall m,$$

Với | . | là L1-norm của vector và vector $u \geq 0$ được dùng kí hiệu vector không âm tức các thành phần của vector đều không âm. Ràng buộc $Card(u_m) = 1$ là ràng buộc mạnh, ta làm yếu ràng buộc này bằng cách thay bằng L1-norm regularization của u_m. Điều này làm cho kết quả u_m khi tối ưu (optimization) sẽ có rất ít thành phần khác 0 trong u_m. Công thức tối ưu (3.3) trở thành:

$$\min_{U,V} \sum_{m=1}^{M} \left\| x_m - u_m V \right\|^2 + \lambda \left|u_m\right| \tag{3.4}$$

Thỏa: $$\left\|v_k\right\| \leq 1, \forall k.$$

Điều kiện L2-norm trên v_k để tránh lời giải bất kì. Bởi vì trong công thức của hàm mục tiêu (objective function) trong công thức tối ưu (3.4), ta có lượng $u_m.V$, do vậy ta có thể giảm u_m xuống t lần và tăng V lên t lần thì ta có thể giảm hàm mục tiêu và bài toán tối ưu như trong công thức (3.4) còn được gọi là biểu diễn thưa (sparse coding).

Chương 4 Hierarchical Spatial Matching Kernel

Trong chương này, đầu tiên tôi mô tả công thức gốc của SPM làm cơ sở cho việc giới thiệu kernel mới HSMK mà sử dụng mô hình thô mịn (coarse to fine – C2F) trên các vùng con như là một cơ sở để cải tiến độ hiệu quả so với kernel SPM.

4.1 Kernel tháp không gian (Spatial Pyramid Matching Kernel – SPMK)

Mỗi ảnh được đại diện là tập hợp các vector đặc trưng trong không gian d chiều. Các đặc trưng được lượng tử hóa (quantized) thành các thành phần rời rạc được gọi là từ hình ảnh (visual words) bằng cách sử dụng thuật toán học phân cụm không giám sát K-means (unsupervised clustering algorithm – Kmeans) hoặc bằng phương pháp biểu diễn thưa (sparse coding). Việc so khớp giữa các đặc trưng cục bộ chuyển thành việc so khớp trên miền rời rạc của các từ hình ảnh (visual words) tương ứng. Điều này có nghĩa là các từ hình ảnh được so khớp khi chúng giống nhau và không so khớp khi chúng không giống nhau.

SPM thực hiện trên một chuỗi các tỉ lệ (scale) khác nhau với l= 0, 1, 2, …,L của ảnh đầu vào. Trên mỗi tỉ lệ co dãn, nó chia ảnh thành 2^l x 2^l các vùng con theo chiều dọc và chiều ngang của ảnh và áp dụng mô hình BoF để đo độ tương đồng giữa các vùng con này. Gọi X, Y là hai tập hợp chứa các vector trong không gian D chiều. Sự tương đồng giữa hai tập hợp trên tỉ lệ l là tổng độ tương đồng giữa tất cả các vùng con tương ứng của ảnh ở tỉ lệ đó.

$$K_l(X,Y) = \sum_{i=1}^{2^{2l}} I(X_i^l, Y_i^l), \tag{4.1}$$

Với X_i^l là tập hợp của đặc trưng của vùng con thứ i ở tỉ lệ l của tập hợp vector ảnh X. Intersection kernel I giữa vùng con X_i^l và Y_i^l được định nghĩa như sau:

$$I(X_i^l, Y_i^l) = \sum_{j=1}^{V} \min(H_{X_i^l}(j), H_{Y_i^l}(j)), \tag{4.2}$$

Với V là tổng số từ hình ảnh được dùng trong bộ từ điển đã xây dựng, $H_\alpha(j)$ là số lần xuất hiện của từ hình ảnh thứ j mà thu được từ việc lượng tử hóa của đặc trưng cục bộ trong tập α. Cuối cùng, kernel SPM được tính là tổng có trọng số được định nghĩa trước của độ tương đồng trên chuỗi các tỉ lệ được áp dụng:

$$K(X, Y) = \frac{1}{2^L} K_0(X, Y) + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} K_l(X, Y). \tag{4.3}$$

Trọng số $\frac{1}{2^{L-l+1}}$ tương ứng ở tỉ lệ l là nghịch đảo tỉ lệ chiều rộng giữa các vùng con được định nghĩa ở tỉ lệ tương ứng. Trọng số này được dùng để bù cho việc so khớp trên nhiều vùng có tỉ lệ khác nhau bởi vì đặc trưng cục bộ dễ tìm thấy sự so khớp ở những vùng rộng lớn hơn. Thêm nữa, những cặp đặc trưng được so khớp ở tỉ lệ l cũng xuất hiện ở tỉ lệ mịn hơn $(l - \zeta)$ của resolution với $\zeta > 0$.

4.2 Kernel đề xuất: Hierarchical Spatial Matching Kernel

Để cải thiện tính hiệu quả trong việc tính toán độ tương đồng giữa các vùng con, tôi đề xuất việc sử dụng mô hình thô mịn (coarse to fine – C2F) trên các vùng con bằng cách thực hiện trên một chuỗi các độ phân giải (resolution) khác nhau (2^{-r} x 2^{-r}) với r = 0, 1, 2, …,R như trong PMK [13].

Với X_i^l và Y_i^l lần lượt là tập hợp của các đặc trưng cục bộ của những vùng con thứ i ở tỉ lệ l của tập hợp vector trong ảnh X, Y. Ở mỗi độ phân giải (resolution) r, tôi áp dụng intersection kernel F^r được chuẩn hóa bằng phương pháp chuẩn hóa căn bậc hai theo đường chéo (square root diagonal normalization) để đo độ tương đồng giữa chúng như sau:

$$F^r(X_i^l, Y_i^l) = \frac{I(X_i^l(r), Y_i^l(r))}{\sqrt{I(X_i^l(r), X_i^l(r)) I(Y_i^l(r), Y_i^l(r))}}, \tag{4.4}$$

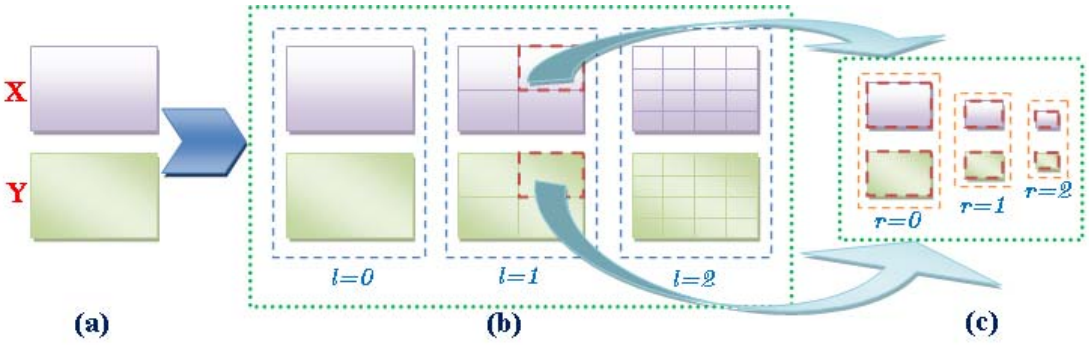
Với $X_i^l(r)$ và $Y_i^l(r)$ lần lượt là tập hợp X_i^l và Y_i^l ở độ phân giải (resolution) r. Thêm nữa, histogram intersection của một tập hợp bất kỳ và chính nó bằng với số phần tử của tập hợp đó (cardinality). Do vậy, gọi $\aleph_{X_i^l(r)}$ và $\aleph_{Y_i^l(r)}$ lần lượt là số phần tử của tập hợp $X_i^l(r)$ và $Y_i^l(r)$ công thức (4.4) trở thành như sau:

$$F^r(X_i^l, Y_i^l) = \frac{I(X_i^l(r), Y_i^l(r))}{\sqrt{\aleph_{X_i^l(r)} \aleph_{Y_i^l(r)}}}. \tag{4.5}$$

Việc chuẩn hóa căn bậc hai theo đường chéo cho intersection kernel không chỉ duy trì việc thỏa điều kiện Mercer về kernel mà còn giúp bù cho việc khác nhau về số lượng phần tử giữa các tập hợp như thể hiện trong công thức (4.5). Để đạt được độ tương đồng tổng hợp của mô hình C2F, tôi định nghĩa sự kết hợp tuyến tính (linear combination) trên một chuỗi các kernel cục bộ, với mỗi thành phần kernel cục bộ được tính toán theo công thức (4.5) ở mỗi độ phân giải. Từ đó, hàm kernel F giữa tập hợp X_i^l và tập hợp Y_i^l trong mô hình C2F được định nghĩa như sau:

$$F(X_i^l, Y_i^l) = \sum_{r=0}^{R} \theta_r F^r(X_i^l, Y_i^l) \tag{4.6}$$

Với: $$\sum_{r=0}^{R} \theta_r = 1, \theta_r \geq 0, \forall r = 0, 1, 2, \ldots, R.$$



Hình 2: Minh họa kernel HSMK được áp dụng trên ảnh X và Y với L=2 và R=2 (a). Đầu tiên, HSMK chia ảnh ra thành 2^l x 2^l các vùng con với l=0, 1, 2

như SPMK (b). Tuy nhiên, HSMK sử dụng mô hình coarse-to-fine cho mỗi vùng con bằng cách tính toán độ tương đồng trên một chuỗi các resolution khác nhau 2^{-r} x 2^{-r} với r = 0, 1, 2 (c). Công thức (4.8) mà vector trọng số được tính từ MKL với kernel cơ bản có phân bố đồng nhất được sử dụng để xấp xỉ độ so khớp tối ưu giữa các vùng con thay vì sử dụng mô hình BoF như trong SPMK

Hơn nữa, khi các kernel cục bộ được kết hợp tuyến tính đưa vào thuật toán SVM, ta có thể chuyển bài toán về dạng MKL, trong đó các kernel cơ bản được định nghĩa như trong công thức (4.5) trên các độ phân giải khác nhau của các vùng con như sau:

$$\begin{aligned} &\min_{w_\alpha, w_0, \xi, \theta} \frac{1}{2}(\sum_{\alpha=1}^{\eta} \theta_\alpha \left\| w_\alpha \right\|_2)^2 + C\sum_{i=1}^{N} \xi_i \\ &s.t. \quad y_i(\sum_{\alpha=1}^{\eta} \theta_\alpha \left\langle w_\alpha, \Phi_\alpha(x_i) \right\rangle + w_0) \geq 1 - \xi_i \\ &\sum_{\alpha=1}^{\eta} \theta_\alpha = 1, \theta \geq 0, \xi \geq 0, \end{aligned} \tag{4.7}$$

Với x_i là mẫu ảnh học, y_i là nhãn lớp tương ứng của x_i, N là số mẫu dùng cho việc học, (w_α , w_0 , ξ) là các tham số của SVM, C là tham số xác định biên mềm (soft margin) được định nghĩa trước để bù cho lỗi của mẫu học trong thuật toán SMV, θ là vector trọng số cho các kernel cục bộ cơ bản, η là số lượng kernel cục bộ cơ bản của các vùng con trên một chuỗi các resolution khác nhau, $\theta \geq 0$ nghĩa là bất kỳ thành phần nào của vector đều không âm (hay vector θ còn được gọi là vector không âm), $\Phi(x)$ là hàm chuyển vector x vào không gian tái sinh kernel Hibert (RKHS) và $\langle .,. \rangle$ ký hiệu cho tích nội (inner product). MKL tìm các tham số cho SVM đồng thời xác định vector trọng số cho các kernel cục bộ cơ bản. Thêm nữa, những kernel cục bộ cơ bản này được định nghĩa trên nhiều độ phân giải khác nhau của cùng vùng con. Do vậy, độ trùng lắp thông tin giữa chúng là cao. Từ thí nghiệm của Gehler và Nowozin [12] và đặc biệt là của Kloft và các đồng sự [16] đã chứng tỏ rằng MKL với kernel cơ bản có phân bố đồng nhất - phương pháp xấp xỉ

để chuyển bài toán MKL về dạng SVM truyền thống với kernel phi tuyến – là phương pháp hiệu quả nhất xét trên khía cạnh của độ chính xác cũng như thời gian tính toán. Do vậy, công thức (4.6) với các hệ số kết hợp tuyến tính đạt được từ MKL với kernel cơ bản phân bố đồng nhất trở thành (Uniform MKL):

$$F(X_i^l, Y_i^l) = \frac{1}{R+1} \sum_{r=0}^{R} F^r(X_i^l, Y_i^l). \tag{4.8}$$

Hình 2 minh họa cách thực hiện của HSMK với L=2 và R=2. HSMK cũng thực hiện việc xem xét vùng con trên một chuỗi các độ phân giải khác nhau như trong PMK để có thể đạt được độ đo tương đồng tốt hơn. Tuy nhiên, HSMK tính toán vector trọng số dựa trên MKL với kernel cơ bản phân bố đồng nhất, do vậy đạt được hiệu quả hơn, cũng như có thể giải thích về mặt lý thuyết thay vì dùng vector trọng số định nghĩa trước như trong PMK. Thêm nữa việc áp dụng chuẩn hóa kernel cơ bản bằng phương pháp chuẩn hóa căn bậc hai theo đường chéo giúp cho HSMK thực hiện tốt trên các tập vector có số phần tử khác nhau mà không được xem xét trong PMK. HSMK được định nghĩa dựa trên việc tính toán SPM trong mô hình C2F, điều này mang đến sự hiệu quả khi thực hiện trên tập hợp vector không thứ tự, thậm chí các tập vector này có số phần tử khác nhau. Về mặt toán học, công thức của HSMK được định nghĩa như sau:

$$K(X,Y) = \frac{1}{2^L} F_0(X,Y) + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} F_l(X,Y) \tag{4.9}$$

Với: $$F_l(X,Y) = \sum_{i=1}^{2^{2l}} F(X_i^l, Y_i^l) = \frac{1}{R+1} \sum_{i=1}^{2^{2l}} \sum_{r=0}^{R} F^r(X_i^l, Y_i^l).$$

Tóm lại, HSMK sử dụng thuật toán cây kd (kd-tree) để chuyển đặc trưng cục bộ thành các từ hình ảnh rời rạc và sau đó intersection kernel được chuẩn hóa bằng phương pháp chuẩn hóa căn bậc hai theo đường chéo dùng để đo độ tương đồng trên histogram có V từ hình ảnh. Mỗi ảnh là tập hợp gồm $\aleph$ đặc trưng cục bộ trong không gian D chiều và thuật toán cây kd có độ phức tạp log(V) để thực hiện việc chuyển đặc trưng cục bộ. Do vậy, độ phức tạp của HSMK là O(DMlog(V)) với

M=max($\aleph_X$, $\aleph_Y$). Vơi lưu ý rằng, độ phức tạp của thuật toán so khớp tối ưu [17] là $O(DM^3)$.

Chương 5 Thực nghiệm

Trong chương này, tôi trình bày việc áp dụng HSMK vào bài toán phân loại ảnh. HSMK được thực nghiệm trên hai bài toán cụ thể cho phân loại ảnh là phân loại đối tượng và phân loại cảnh trên những cơ sở dữ liệu chuẩn như Oxford Flower, CALTECH-101, CALTECH-256, MIT Scene, UIUC Scene.

5.1 Phân loại ảnh (Image categorization)

5.1.1 Giới thiệu bài toán phân loại ảnh

Phân loại ảnh là bài toán phân loại mỗi ảnh đã cho vào một lớp ngữ nghĩa cụ thể (semantic class). Lớp ngữ nghĩa được định nghĩa dựa trên việc ảnh mô tả loại phong cảnh gì, ví dụ: núi, bờ biển hay tòa nhà - trong trường hợp này, đây là bài toán con của phân loại ảnh, còn được biết dưới tên phân loại cảnh (scene categorization). Ngoài ra, lớp ngữ nghĩa có thể được định nghĩa là ảnh chứa đối tượng quan tâm, ví dụ: ghế, du thuyền hay gấu trúc – trường hợp này, bài toán còn được gọi dưới tên phân loại đối tượng (object categorization). Với ghi chú, mỗi ảnh chỉ thuộc về một lớp xác định đã được định nghĩa trước.

Phân loại ảnh là một trong những bài toán trong lĩnh vực thị giác máy tính nhận được sự quan tâm lớn nhất trong cộng đồng nghiên cứu, là một trong những chủ đề chính thảo luận chính của các hội nghị hàng đầu thế giới như CVPR, ICCV, ECCV… Phân loại ảnh là bài toán cho trước tập dữ liệu huấn luyện gồm nhiều lớp với số lượng cố định, mỗi ảnh được đánh nhãn thuộc về một lớp nhất định, yêu cầu của bài toán là xây dựng mô hình ảnh cũng như phương pháp học để có thể xác định chính xác nhãn của các ảnh trong tập dữ liệu kiểm tra. Tập dữ liệu huấn liệu và tập dữ liệu kiểm tra là hai tập dữ liệu riêng biệt, không có phần chung. Trong quá trình huấn luyện, các ảnh huấn luyện sẽ có nhãn tương ứng đi kèm. Ngược lại, trong quá trình kiểm tra, chương trình sẽ xác định nhãn của từng ảnh đã được bỏ nhãn đi kèm, kết quả sẽ được so sánh với nhãn đi kèm tương ứng của từng ảnh, nếu giống nhau tức kết quả dự đoán của chương trình là chính xác, ngược lại là sai.

Mô hình toán cho bài toán phân loại ảnh có thể diễn đạt như sau: cho tập dữ liệu (dataset) D chứa M ảnh $X = \{X_1, X_2, \ldots, X_M\}$ được định nghĩa trên N lớp ngữ nghĩa $Y = \{Y_1, Y_2, \ldots, Y_N\}$, trong đó mỗi ảnh X_i thuộc X được phân loại vào một lớp duy nhất $Y_i \in Y$ hay ta có thể nói ảnh X_i được gán nhãn Y_i.

Chọn ngẫu nhiên k ảnh từ mỗi lớp ngữ nghĩa cho trước (N lớp) để tạo tập dữ liệu huấn luyện D_{Train}. Ghi chú, mỗi lớp ta có thể chọn số lượng ảnh khác nhau làm dữ liệu huấn luyện, nhưng thông thường người ta thường chọn số lượng ảnh của mỗi lớp là bằng nhau để tránh đi hiện tượng bias dữ liệu (tạm dịch hiện tượng chiếm ưu thế về số lượng của một lớp so với lớp khác) không cần thiết. Tập các ảnh còn lại được gọi là tập dữ liệu kiểm tra D_{Test}. Mục tiêu của bài toán là từ tập ảnh dữ liệu huấn luyện D_{Train}, ta tìm bộ phân lớp F nhận thông tin đầu vào là X_i và trả về Y_i tương ứng sao cho khi thực hiện việc kiểm tra bộ phân lớp F trên tập dữ liệu D_{Test} đạt độ chính xác cao nhất có thể.

Giai đoạn huấn luyện: từ tập dữ liệu $D_{Train} = \{(X_i, Y_i) \mid X_i$ thuộc tập ảnh được chọn, Y_i là nhãn tương ứng của $Y_i\}$. Mục tiêu là học bộ phân lớp $F: X \rightarrow Y$.

Giai đoạn kiểm tra: trên tập dữ liệu $D_{Test} = \{(X_j, Y_j) \mid X_j$ thuộc tập ảnh không được chọn huấn luyện, Y_j là nhãn tương ứng của $X_j\}$. Dùng bộ phân lớp F đã được học từ tập huấn luyện và ảnh kiểm tra X_j, ta thu được kết quả phân lớp là $Z_j = F(X_j)$. Nếu $Z_j=Y_j$ thì kết luận là việc phân lớp cho ảnh X_j là đúng, ngược lại nếu $Z_j \neq Y_j$ thì việc phân lớp cho ảnh X_j là sai.

5.1.2 Ứng dụng của phân loại ảnh

Từ tập dữ liệu học – tập dữ liệu được gán nhãn do con người xây dựng, hệ thống có thể học để thực hiện việc phân lớp cho một lượng vô hạn các ảnh mới cần được gán nhãn. Nói cách khác, ta chỉ thực hiện gán nhãn thủ công trên một số lượng nhỏ ảnh (xây dựng tập dữ liệu học) hay có thể xem thao tác này như sự định nghĩa về mặt ngữ nghĩa cho hệ thống hoạt động sau này, từ đó hệ thống sẽ giúp ta gán nhãn tự động cho số lượng ảnh mới tùy ý.

Phân loại ảnh là một phần quan trọng trong việc xây dựng hệ thống truy vấn ảnh dựa trên thông tin ảnh. Đây là một trong những ứng dụng quan trọng, do số lượng

ảnh càng ngày càng lớn, phát triển theo cấp số mũ như có thể thấy qua các website chia sẽ ảnh như Flickr, Picasa v.v…, trong khi ở thời điểm luận văn được viết, các máy tìm kiếm như Google, Bing, Yahoo v.v… chỉ có thể dựa trên thông tin văn bản (text), các hỗ trợ truy vấn trên ảnh thực ra chỉ được thực hiện bằng cách truy vấn văn bản trên những đánh dấu ngữ nghĩa (tag) bằng văn bản của người sử dụng tạo cho ảnh, hoặc tên của tập tin ảnh thay vì nội dung mà ảnh chứa.

Hơn nữa, giải quyết bài toán phân loại ảnh tức giải quyết bài toán làm thế nào để biểu diễn đối tượng trong ảnh tốt, cũng như tìm được độ đo tương đồng thích hợp giữa các đối tượng, do vậy nó có thể mở rộng để giải các bài toán quan trọng khác trong lĩnh vực thị giác máy tính như phát hiện đối tượng, nhận dạng đối tượng v.v…

5.1.3 Những thách thức của bài toán phân loại ảnh

Phân loại ảnh là bài toán có rất nhiều thách thức. Do mỗi ảnh chỉ được gán nhãn thuộc về một lớp nhất định, và trong tự nhiên, các đối tượng cũng như phong cảnh tự nhiên rất đa dạng và phong phú, do vậy thông tin về nhãn chỉ cung cấp như phần thông tin khái quát về ảnh, nói một cách khác, đây là ngữ nghĩa của ảnh. Trong khi, mỗi ảnh là một thể hiện cụ thể của lớp ngữ nghĩa và được lưu trữ theo từng pixel, do vậy sẽ có những thách thức sau:

- Khác biệt về góc nhìn
- Khác biệt về chiếu sáng
- Đối tượng bị che phủ một phần
- Khác biệt về độ lớn của đối tượng
- Sự sai biệt của đối tượng (hình được vẽ lại)
- Đối tượng bị nhiễu do thông tin nền
- Sự đa dạng về thể hiện của đối tượng

5.1.4 Các hướng tiếp cận

Có hai hướng tiếp cận chính cho bài toán phân loại ảnh: (i) dựa trên đặc trưng, (ii) dựa trên phương pháp học để có thể phân loại các đối tượng.

5.1.4.1 Hướng tiếp cận dựa trên đặc trưng

Trong công trình [26], Nilsback và các đồng sự đã khảo sát nhiều loại đặc trưng cho bài toán phân loại đối tượng trên cơ sở dữ liệu Oxford Flower, từ màu sắc đền hình dạng của hoa, cũng như xây dựng những đặc trưng dựa trên việc phân đoạn (segmentation) ảnh, để rút trích ra các đối tượng, loại bỏ vùng nền (background) có thể gây nhiễu cho việc phân loại, và xây dựng đặc trưng trên vùng đối tượng được phân đoạn này: như SIFT-Internal, SIFT-Boundary (cả hai đặc trưng này dựa trên đặc trưng SIFT được công bố bởi Lowe trong [20][21]). Thêm nữa, SIFT-Internal được biết là đặc trưng tốt nhất cho bài toán phân loại đối tượng trên cơ sở dữ liệu Oxford Flower.

Trong công trình [27], Oliva và Torralba đã đề xuất đặc trưng toàn cục – GIST để mô hình cho các ảnh chứa cảnh để giải quyết bài toán phân loại cảnh như trên cơ sở dữ liệu MIT Scene hay UIUC Scene (UIUC Scene là cơ sở dữ liệu mở rộng từ cơ sở dữ liệu MIT Scene). GIST được biết là đặc trưng tốt nhất cho bài toán phân loại cảnh.

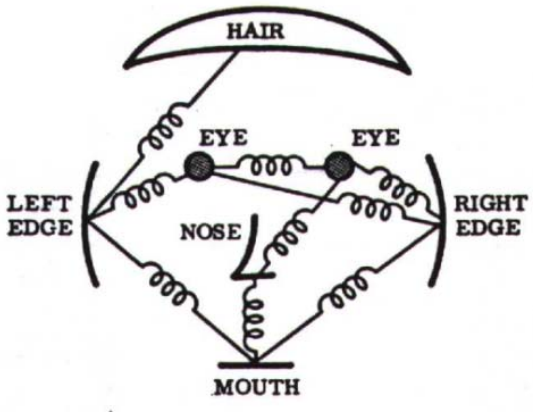
Ngoài ra, trong các công trình [4][12][15] cũng trình bày các thí nghiệm trong việc kết hợp nhiều loại đặc trưng lại với nhau để miêu tả được ảnh tốt hơn. Nhưng việc kết hợp của nhiều loại đặc trưng yêu cầu chọn lựa bộ trọng số cho mỗi đặc trưng để khi kết hợp lại với nhau đạt hiệu quả tốt nhất. Vấn đề này dẫn đến các bài toán liên quan đến việc xây dựng phương pháp học, Gehler và Nowozin [12] đã đề xuất sử dụng phương pháp học MKL hoặc LP-B (Linear Programming Boosting) để xác định bộ trọng cho các đặc trưng ngay trong quá trình học, thêm nữa MKL cũng được dùng để kết hợp các đặc trưng như trong [33][37].

Dựa trên việc khảo sát các công trình công bố trên bài toán phân loại ảnh như trong [7][8][13][14][18][34][36], đặc trưng SIFT là đặc trưng được sử dụng phổ biến nhất và đạt hiệu quả cao cho bài toán phân loại ảnh.

5.1.4.2 Hướng tiếp cận dựa trên phương pháp học

Fergus và các đồng sự [10] sử dụng tập hợp các thành phần khác nhau của đối tượng và xây dựng đồ thị để mô hình đối tượng dựa trên mối quan hệ về vị trí giữa

các thành phần cũng như sự hiện diện của các thành phần đối tượng trong ảnh. Ý tưởng này phát triển từ ý tưởng của mô hình mối liên hệ giữa các thành phần (pictorial) được sử dụng trong công trình của Fischler và Elschlager [11] như được minh họa trong Hình 3, điểm khác là cách tiếp cận của Fergus chỉ giữ lại những phần mà cần thiết để có thể phân biệt giữa các lớp đối tượng với nhau, do vậy nó có thể tránh việc mô hình những phần khác nhau có tính toàn cục (tức vấn đề cùng một lớp đối tượng, nhưng các đối tượng lại có nhiều thể hiện khác nhau hay sự đa dạng của thể hiện của cùng một đối tượng), mô hình này còn được gọi là mô hình chòm sao (constellation model).



Hình 3: Mô hình mối liên hệ giữa các thành phần (Pictorial)

Felzenszwalb và các đồng sự [9] đề xuất phương pháp latent-SVM để xây dựng mô hình quyết định cho việc mô hình đối tượng mà sử dụng tập của các phần đối tượng ở nhiều độ phân giải khác nhau dựa trên HoG [5] (deformable part model).

Ngoài ra, xây dựng bộ học dựa trên kernel để đo độ tương đồng của các đối tượng hỗ trợ cho việc phân loại là hướng tiếp cận được nhiều nhà nghiên cứu quan tâm [13][18][34][36]. Nổi bật trong hướng tiếp cận này là mô hình BoF và SPM như đã được trình bày trong Chương 3.

Kernel đề xuất – HSMK theo hướng tiếp cận kernel, áp dụng cho bài toán phân loại ảnh. HSMK là sự cải tiến của SPMK để có thể tính toán độ tương đồng giữa ảnh tốt hơn dưới nhiều thách thức của bài toán phân loại ảnh.

5.2 Thực nghiệm

Hầu hết các phương pháp tiếp cận gần đây cho bài toán phân loại ảnh (image categorization) đều sử dụng đặc trưng cục bộ bất biến để miêu tả ảnh, bởi vì các đặc trưng này dễ miêu tả và so khớp khi đối tượng hoặc cảnh trong hình có nhiều thay đổi về góc nhìn (viewpoints), được chiếu sáng khác nhau, do chụp ở các thời điểm khác nhau trong ngày (illuminations) hoặc sự phức tạp của thông tin nền trong ảnh (background clutter). Trong những đặc trưng bất biến cục bộ, SIFT [20] là một trong những đặc trưng hiệu quả và được sử dụng phổ biến nhất. Để đạt được tính phân biệt cao, tôi sử dụng SIFT với mật độ dày (dense SIFT) bằng cách sử dụng bộ miêu tả SIFT trên vùng chữ nhật con 16 x 16 và tính toán trên tất cả các điểm ảnh (pixel) thay vì chỉ tính trên các điểm quan trọng (key points) như trong [20] hay trên một lưới các điểm như trong [18]. Thêm nữa, để tăng khả năng mở rộng, dense SIFT được tính toán trên ảnh mức xám. Đặc trưng cục bộ với phân bố dày mang đến khả năng miêu tả thông tin của những vùng đồng nhất, ví dụ như bầu trời, nước hay cỏ mà ở những trường hợp này điểm quan trọng (key points) không tồn tại. Hơn nữa, sự kết hợp của đặc trưng phân bố dày (dense features) và mô hình C2F cho phép ảnh được miêu tả chính xác hơn bởi vì những đặc trưng cục bộ có thể miêu tả đầy đủ hơn thông tin của những vùng lân cận (neighbor information) trên nhiều cấp khác nhau của resolution ảnh. Sau đó, thuật toán phân cụm không giám sát Kmeans được sử dụng trên tập con ngẫu nhiên chứa các đặc trưng SIFT để xây dựng tập các từ hình ảnh, còn được gọi là từ điển. Từ các kết quả trong các công trình đã được công bố, tôi thực hiện thí nghiệm với hai kính cỡ khác nhau của từ điển là M bằng 400 và 800.

Để chứng minh sự hiệu quả của HSMK, tôi thực hiện trên 2 thể hiện khác nhau của bài toán phân loại ảnh: phân loại đối tượng (object categorization) và phân loại cảnh (scene categorization). Đối với bài toán phân loại đối tượng, chúng tôi sử dụng cơ sở dữ liệu Oxford Flower, và để chứng minh khả năng mở rộng cũng như tính hiệu quả của HSMK trên cở sở dữ liệu lớn, tôi cũng thực hiện thí nghiệm trên hai cơ sở dữ liệu lớn cho bài toán phân loại đối tượng là: CALTECH-101 [7], và

CALTECH-256 [14]. Đối với bài toán phân loại cảnh, tôi đánh giá độ hiệu quả của HSMK trên cở sở dữ liệu MIT scene [27] và UIUC scene [18].

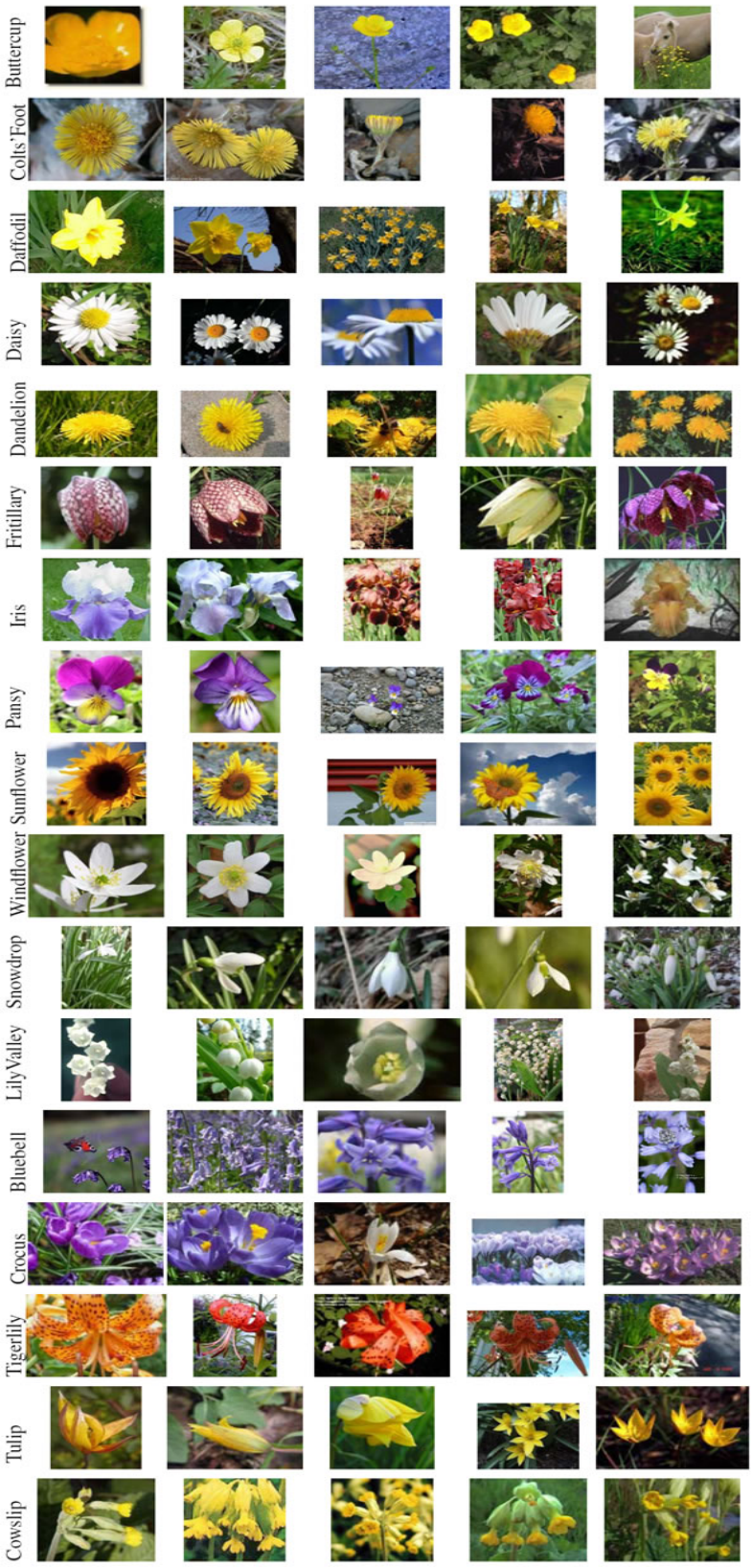
5.2.1 Phân loại đối tượng

5.2.1.1 Cơ sở dữ liệu Oxford Flowers:

Cơ sở dữ liệu này chứa 17 lớp các loại hoa thông dụng của Vương quốc Anh, được thu thập bởi Nilsback và các đồng sự [25]. Mỗi lớp chứ 80 ảnh hoa được chụp với các kích thước khác nhau, góc nhìn khác nhau cũng như có điều kiện chiếu sáng khác nhau. Hơn nữa, hoa trong cùng 1 lớp như Irises, Fritillaries và Pansies có sự đa dạng rất lớn về màu sắc cũng như hình dạng, và trong một số trường hợp độ tương đồng giữa các lớp hoa khác nhau rất gần nhau, ví dụ như giữa Dandelion và Colts'Foot. Hình 4 minh họa một số mẫu hoa trong cơ sở dữ liệu Oxford Flower. Để thực hiện thí nghiệm, tôi sử dụng cách thiết lập của Gehler và Nowozin [12], chọn ngẫu nhiên 40 mẫu từ mỗi lớp để học (training set) và dùng phần còn lại cho việc kiểm tra (testing set), và không sử dụng tập kiểm thử (validation set) như trong [25][26] để chọn tham số tối ưu cho hệ thống.

Bảng 5.1: Bảng so sánh độ chính xác phân lớp (%) khi sử dụng một đặc trưng trên cở sở dữ liệu Oxford Flower (với NN ký hiệu cho thuật toán phân lớp láng giềng gần nhất: Nearest Neighbour)

Phương pháp	Độ chính xác (%)
HSV (NN) [26]	43.0
SIFT-Internal (NN) [26]	55.1
SIFT-Boundary (NN) [26]	32.0
HOG [26]	49.6
HSV (SVM) [12]	61.3
SIFT-Internal (SVM) [12]	70.6
SIFT-Boundary (SVM) [12]	59.4
HOG (SVM) [12]	58.5
SIFT (MSL) [34]	65.3
Dense SIFT (HSMK)	**72.9**


Buttercup
Colts'Foot
Daffodil
Daisy
Dandelion
Fritillary
Iris
Pansy
Sunflower
Windflower
Snowdrop
LilyValley
Bluebell
Crocus
Tigerlily
Tulip
Cowslip

Hình 4: Minh họa cơ sở dữ liệu Oxford Flower (17 lớp)[1]

Bảng 5.1 cho thấy rằng HSMK đạt được kết quả tốt nhất (state-of-the-art result) khi sử dụng một loại đặc trưng so với các hướng tiếp cận đã có. Nó không chỉ cho kết quả tốt hơn SIFT-Internal [26] – mà được biết là loại đặc trưng tốt nhất cho cơ sở dữ liệu này, với lưu ý là SIFT-Internel được tính toán trên ảnh đã được segmentation, mà còn tốt hơn cả SPM với hệ số tối ưu bằng hệ thống học tỉ lệ MSL [34]. Thêm nữa, Bảng 5.2 cho thấy rằng kết quả đạt được từ HSMK cũng tốt hơn so với SPMK.

Bảng 5.2: Bảng so sánh độ chính xác phân lớp (%) giữa HSMK và SPMK trên cơ sở dữ liệu Oxford Flower

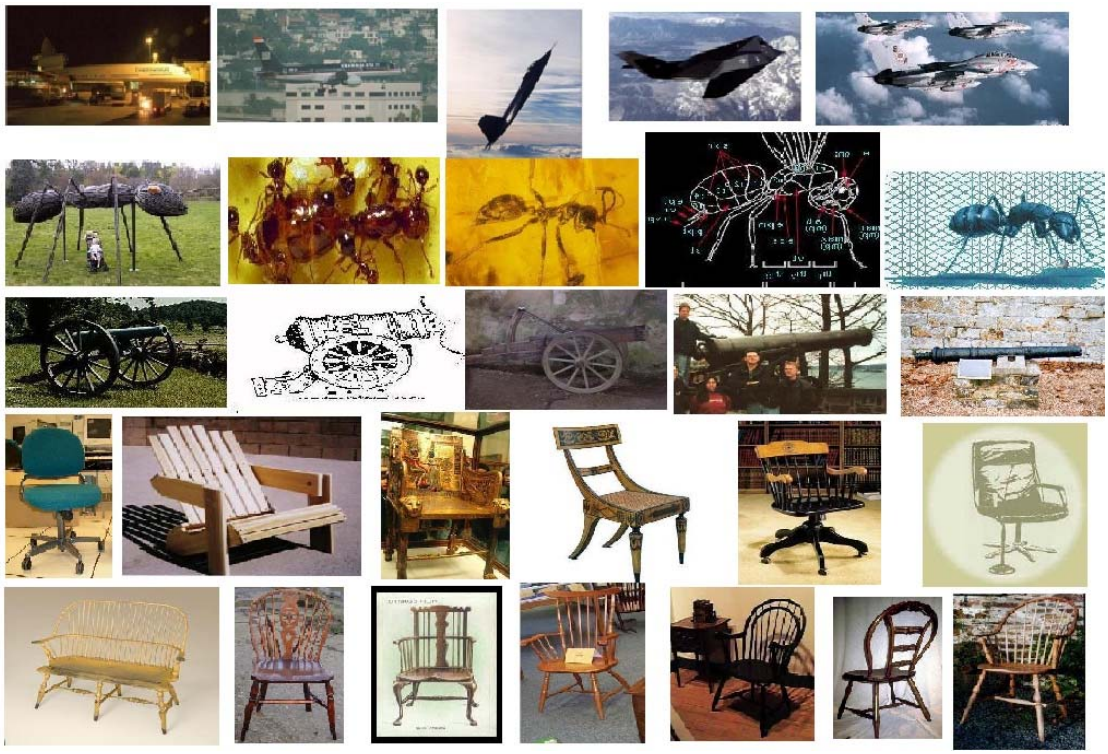
Kernel	M = 400	M = 800
SPMK	68.09%	69.12%
HSMK	**71.76%**	**72.94%**

5.2.1.2 Cơ sở dữ liệu CALTECH:

Để cho thấy tính hiệu quả cũng như khả năng mở rộng, tôi cũng đánh giá HSMK trên cơ sở dữ liệu lớn CALTECH-101 và CALTECH-256.Những cơ sở dữ liệu này có tính đa dạng các thể hiện trong cùng một lớp rất lớn, cũng như sự đa dạng về góc nhìn và cả sự phức tạp của nền trong ảnh. Thêm nữa, như trong Hình 5 minh họa một số mẫu trong cơ sở dữ liệu CALTECH-101, mỗi hàng minh họa một lớp trong cơ sở dữ liệu, ta có thể nhận thấy ở hang thứ 4 thể hiện lớp chair và hàng thứ 5 thể hiện lớp Windsor_chair rất giống nhau về hình dáng và cả độ đa dạng trong cùng lớp. Đối với cơ sở dữ liệu CALTECH-101, tôi thực hiện thí nghiệm khi sử dụng 5, 10, 15, 20, 25, 30 mẫu học để huấn luyện cho mỗi lớp, bao gồm cả lớp nền (background class) và sử dụng đến 50 mẫu mỗi lớp cho kiểm tra. Bảng 5.3 so sánh kết quả phân lớp dựa trên HSMK và các cách tiếp cận khác. Có thể thấy rằng, HSMK đạt kết quả tương ứng (comparable result) với kết quả tốt nhất (state-of-the-art result) thậm chí khi chỉ sử dụng một loại đặc trưng trong khi các cách tiếp cận

[1] Cơ sở dữ liệu Oxford Flower được cung cấp cho nghiên cứu khoa học ở địa chỉ: http://www.robots.ox.ac.uk/%7Evgg/data/flowers/17/17flowers.tgz

khác sử dụng kết hợp của nhiều loại đặc trưng và thuật toán học phức tạp như MKL và linear programming boosting (LP-B) [12].



Hình 5: Minh họa cơ sở dữ liệu CALTECH-101[2]

Bảng 5.4 cho thấy rằng kết quả đạt được từ HSMK cũng tốt hơn so với kết quả đạt từ SPMK. Thêm nữa, khi thực hiện thí nghiệm nếu bỏ lớp nền thì HSMK đạt được độ chính xác 78.4% cho trường hợp dùng mỗi lớp 30 mẫu để học. Điều này chứng tỏ, cách tiếp cận đề xuất có tính hiệu quả cao mặc dù được thực hiện đơn giản hơn so với các cách tiếp cận đạt kết quả cao khác.

Bảng 5.3: Bảng so sánh kết quả phân lớp trên cơ sở dữ liệu CALTECH-101

	5 (mẫu học)	10 (mẫu học)	15 (mẫu học)	20 (mẫu học)	25 (mẫu học)	30 (mẫu học)
Grauman & Darrell [13]	34.8%	44.0%	50.0%	53.5%	55.5%	58.2%
Wang & Wang [34]	-	-	61.4%	-	-	-
Lazebnik et al. [18]	-	-	56.4%	-	-	64.6%

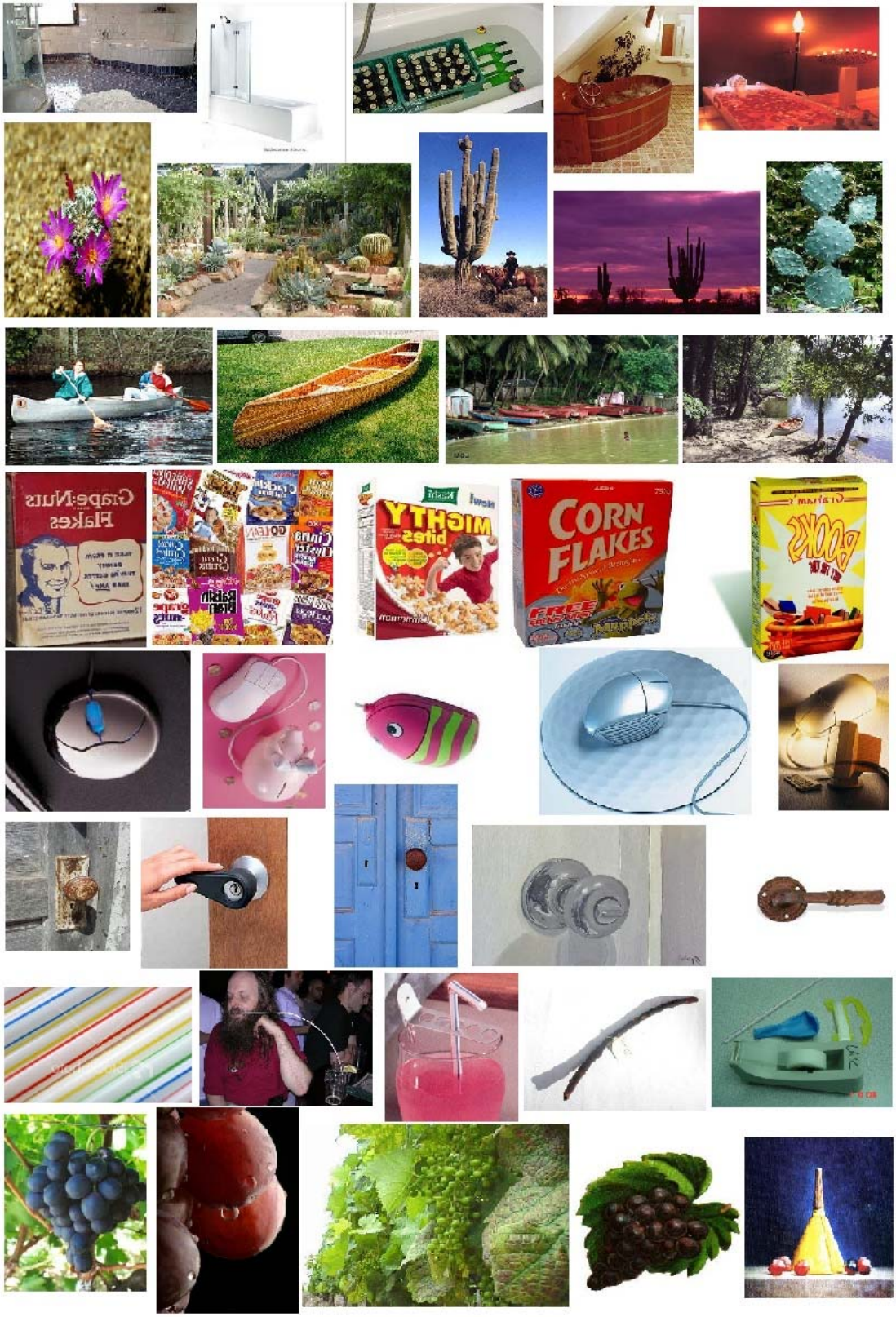
[2] Cơ sở dữ liệu CALTECH-101 được cung cấp ở địa chỉ: http://www.vision.caltech.edu/Image_Datasets/Caltech101/101_ObjectCategories.tar.gz

Yang et al. [36]	-	-	67.0%	-	-	73.2%
Boimann et al. [4]	56.9%	-	72.8%	-	-	79.1%
Gehler & Nowozin (MKL) [12]	42.1%	55.1%	62.3%	67.1%	70.5%	73.7%
Gehler & Nowozin (LP-Beta) [12]	54.2%	65.0%	70.4%	73.6%	75.5%	77.8%
Gehler & Nowozin (LP-B) [12]	46.5%	59.7%	66.7%	71.1%	73.8%	77.2%
Phương pháp đề xuất (HSMK)	50.5%	62.2%	69.0%	72.3%	74.4%	77.3%

Bảng 5.4: Bảng so sánh độ chính xác phân lớp của HSMK và SPMK trên cở sở dữ liệu CALTECH-101

	5 (mẫu học)	10 (mẫu học)	15 (mẫu học)	20 (mẫu học)	25 (mẫu học)	30 (mẫu học)
SPMK (M = 400)	48.18%	58.86%	65.34%	69.35%	71.95%	73.46%
HSMK (M = 400)	**50.68%**	**61.97%**	**67.91%**	**71.35%**	**73.92%**	**75.59%**
SPMK (M = 800)	48.11%	59.70%	66.84%	69.98%	72.62%	75.13%
HSMK (M = 800)	**50.48%**	**62.17%**	**68.95%**	**72.32%**	**74.36%**	**77.33%**

Hình 6 minh họa sự đa dạng về thể hiện của các đối tượng trong cơ sở dữ liệu CALTECH-256, mỗi hàng là một lớp trong cơ sở dữ liệu, CALTECH-256 là phiên bản mở rộng của CALTECH-101, nhưng không được chuẩn hóa như trong CALTECH-101 nên sự phức tạp về nền là rất lớn. Và trên cơ sở dữ liệu CALTECH-256, tôi thực hiện thí nghiệm với HSMK khi sử dụng 15 và 30 mẫu từ mỗi lớp cho việc học, bao gồm cả lớp nền (clutter class) và 25 mẫu cho mỗi lớp cho việc kiểm tra, các mẫu đều được chọn ngẫu nhiên từ cơ sở dữ liệu CALTECH-256. Tôi cũng lập trình lại thuật toán SPMK [14] nhưng sử dụng đặc trưng SIFT với phân bố dày từ thí nghiệm của tôi để có thể so sánh công bằng về sự hiệu quả của HSMK và SPMK. Như trong Bảng 5.5, HSMK cho độ chính xác phân lớp hơn 3 phần trăm so với độ chính xác của SPMK.


Grape-Nuts
Flakes
MIGHTY
bites
CORN
FLAKES

Hình 6: Minh họa cơ sở dữ liệu CALTECH-256[3]

Bảng 5.5: Bảng so sánh kết quả phân lớp trên cơ sở dữ liệu CALTECH-256

Kernel	15 (mẫu học)	30 (mẫu học)
Griffin et al. (SPMK) [14]	28.4%	34.2%
Yang et al. (ScSPM) [36]	27.7%	34.0%
Gehler & Nowozin (MKL) [12]	30.6%	35.6%
SPMK (với Dense SIFT)	25.3%	31.3%
Phương pháp đề xuất (HSMK)	**27.2%**	**34.1%**

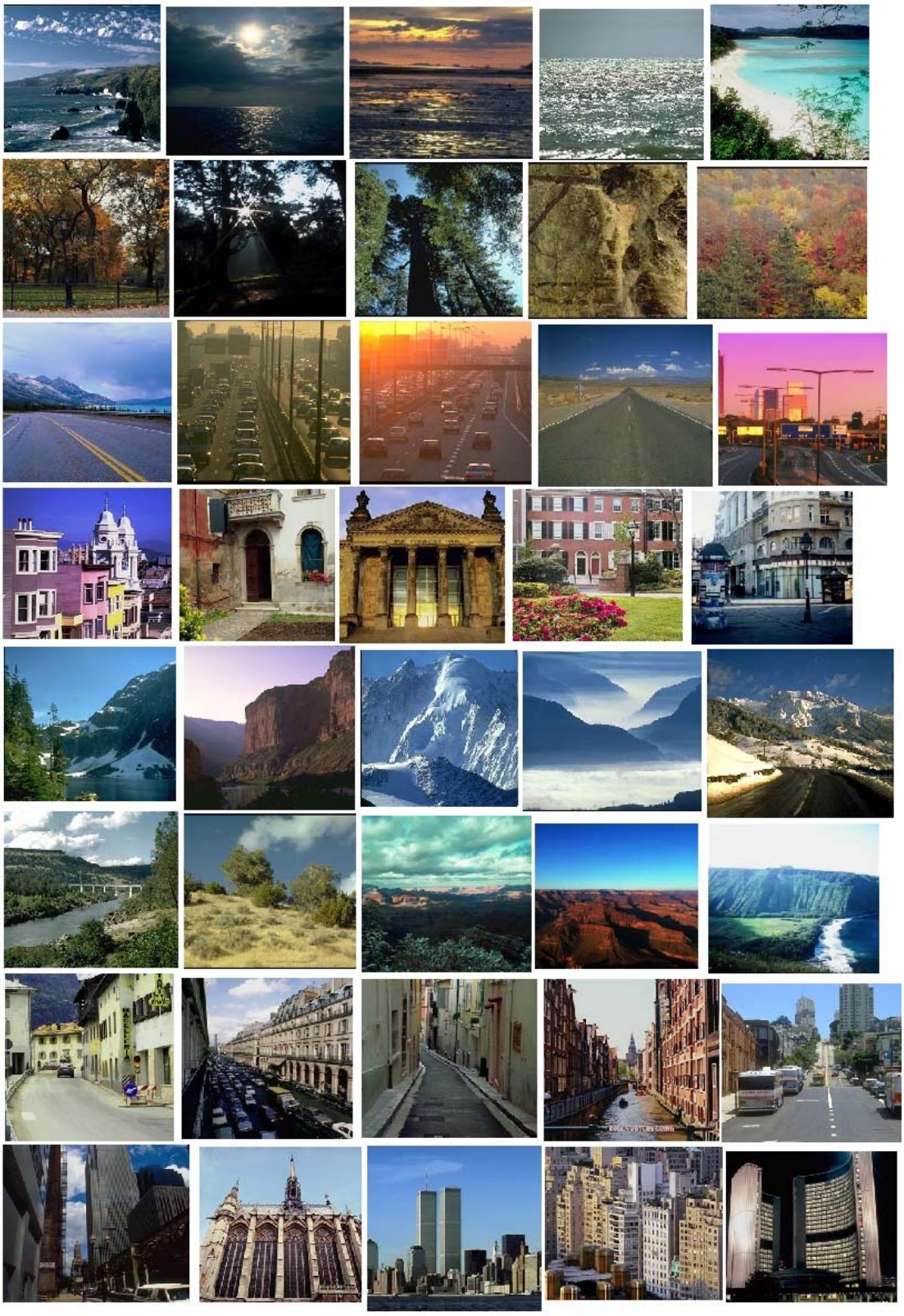
5.2.2 Phân loại cảnh (scene categorization)

HSMK cũng được thí nghiệm trên cở sở dữ liệu MIT Scene (gồm 8 lớp) và UIUC Scene (gồm 15 lớp). Trên những cơ sở dữ liệu này, tôi chọn kích cở của bộ từ điển là M = 400. Hình 7 minh họa một số mẫu trong cở sở dữ liệu MIT Scene, mỗi hàng là một lớp, cơ sở dữ liệu UIUC Scene là sự mở rộng của MIT Scene, nó bao gồm 8 lớp của MIT Scene và bổ sung thêm 7 lớp nữa, nhưng UIUC Scene các ảnh là ảnh mức xám, trong khi MIT Scene thì chứa ảnh màu. Trên cở sở dữ liệu MIT Scene, tôi chọn ngẫu nhiên 100 mẫu từ mỗi lớp cho việc huấn luyện và chọn ngẫu nhiên 100 mẫu khác trên mỗi lớp cho việc kiểm thử. Như trong Bảng 5.6, tỉ lệ phân lớp của HSMK cao hơn 2.5 phần trăm so với SPMK. Cách tiếp cận được đề xuất cũng cho kết quả cao hơn sơ với các cách tiếp cận khác sử dụng đặc trưng cục bộ [15] cũng như sự kết hợp của nhiều đặc trưng cục bộ [15] trên 10 phần trăm, và cũng tốt hơn so với cách sử dụng đặc trưng toàn cục GIST [27] mà được biết như đặc trưng tốt nhất trong việc phân loại cảnh.

Bảng 5.6: Bảng so sánh kết quả phân lớp trên cơ sở dữ liệu MIT Scene (8 lớp)

Phương pháp	Độ chính xác (%)
GIST [27]	83.7
Đặc trưng cục bộ [15]	77.2
Dense SIFT (SPMK)	85.8
Dense SIFT (HSMK)	**88.3**

[3] Cơ sở dữ liệu CALTECH-256 được cung cấp ở địa chỉ:
http://www.vision.caltech.edu/Image_Datasets/Caltech256/



Hình 7: Minh họa cơ sở dữ liệu MIT-Scene (8 lớp)[4]

Trên cơ sở dữ liệu UIUC Scene[5], tôi thiết lập thí nghiệm như miêu tả trong công trình của Lazebnik và các đồng sự [18]. Chọn ngẫu nhiên 100 mẫu từ mỗi lớp cho việc học và kiểm tra tất cả các mẫu còn lại trong cơ sở dữ liệu. Từ Bảng 5.7 cho thấy, kết quả từ HSMK cũng tốt hơn so với kết quả của SPMK [18] và SPM dựa trên biểu diễn thưa (sparse coding) [36].

Bảng 5.7: Bảng so sánh kết quả phân lớp trên cơ sở dữ liệu MIT Scene

Phương pháp	Độ chính xác (%)
Lazebnik et al. [18]	81.4
Yang et al. [36]	80.3
SPMK	79.9
Phương pháp đề xuất (HSMK)	**82.2**

5.2.3 Thí nghiệm Sparse Coding cho Hierarchical Spatial Matching Kernel (ScHSMK)

Để nâng cao hiệu quả phân lớp, thay vì sử dụng Kmeans để thành lập từ điển, và thông kê từ vựng, tôi thí nghiệm sử dụng mô hình biểu diễn thưa (sparse coding) kết hợp với HSMK trên hai cơ sở dữ liệu phân loại đối tượng là Oxford Flower và CALTECH-101.

5.2.3.1 ScHSMK trên cơ sở dữ liệu Oxford Flower

Đối với cơ sở dữ liệu Oxford Flower, tôi sử dụng kích cỡ của từ điển là M=800, và trong quá trình tính HSMK, tôi thí nghiệm với trường hợp dùng kernel tuyến tính (tích nội) thay cho công thức (4.2) – intersection kernel và gọi là Linear Hierarchial Spatial Matching Kernel (HSMK-L), các tham số thí nghiệm khác được sử dụng như trong phần 5.2.1.1. Bảng 5.8 cho thấy kết quả phân lớp sử dụng biểu diễn thưa

[4] Cơ sở dữ liệu MIT-Scene được cung cấp ở địa chỉ:
http://people.csail.mit.edu/torralba/code/spatialenvelope/spatial_envelope_256x256_static_8outdoorcategories.zip

[5] Cơ sở dữ liệu UIUC-Scene được cung cấp ở địa chỉ:
http://www-cvr.ai.uiuc.edu/ponce_grp/data/scene_categories/scene_categories.zip

(Sparse Coding) luôn cho kết quả tốt hơn so với khi sử dụng lượng tử hóa vector (vector quantization) (ví dụ như sử dụng thuật toán Kmeans). Trong trường hợp sử dụng biểu diễn thưa (sparse coding) thì HSMK cũng tốt hơn so với SPMK khoảng 2 phần trăm cả khi sử dụng kernel tuyến tính hay intersection kernel cho tính toán cơ bản trên các vùng con như trong công thức (4.2). Thêm nữa từ Bảng 5.8, ta có thể thấy biểu diễn thưa (sparse coding) có xu hướng làm cho các đặc trưng đạt được tính tuyến tính nhiều hơn so với lượng tử hóa vector (vector quantization), nên khi ta thay intersection kernel bằng kernel tuyến tính, kết quả không thay đổi đáng kể.

Bảng 5.8: Bảng so sánh kết quả phân lớp sử dụng Sparse Coding so với sử dụng vector quantization (Kmeans) trên Oxford Flower

Phương pháp	Độ chính xác (%)
SPMK	69.12
Sparse Code + SPMK + Linear kernel (ScSPMK-L)	71.18
Sparse Code + SPMK + Intersection kernel (ScSPMK)	73.09
HSMK	72.94
Sparse Code + HSMK + Linear Kernel (ScHSMK-L)	73.82
Sparse Code + HSMK + Intersection kernel (ScHSMK)	75.00

5.2.3.2 ScHSMK trên cơ sở dữ liệu CALTECH-101

Đối với cơ sở dữ liệu CALTECH-101, tôi sử dụng hai loại kích cỡ của từ điển là M=400 và M=800. Và thí nghiệm cho hai trường hợp về số lượng mẫu học là 15 và 30 mẫu học cho mỗi lớp, các tham số khác như trong thí nghiệm ở phần 5.2.1.2 đối với cơ sở dữ liệu CALTECH-101 .Tôi cũng thực hiện thí nghiệm với trường hợp dùng kernel tuyến tính (tích nội) thay cho công thức (4.2) – intersection kernel như trong thí nghiệm ở phần 5.2.3.1. Bảng 5.9 cho thấy HSMK với biểu diễn thưa (Sparse coding) đạt được kết quả tối ưu (state of the art) trên cơ sở dữ liệu CALTECH-101. HSMK luôn tốt hơn SPMK khoảng 2 đến 4 phần trăm với cùng phương pháp tạo từ vựng lượng tử hóa vector (vector quantization) hay biểu diễn thưa (sparse coding), sử dụng kernel tuyến tính hay intersection kernel. Khi sử dụng

biểu diễn thưa (sparse coding) kết quả được cải thiện so với khi chỉ sử dụng lượng tử hóa vector (vector quantization), điều này có thể giải thích qua công thức tối ưu như được trình bày trong phần 0. Biểu diễn thưa (Sparse coding) cũng làm cho các đặc trưng trở nên tuyến tính hơn, như có thể thấy kết quả phân lớp khi sử dụng biểu diễn thưa (sparse coding) với kernel tuyến tính, ta có thể thu được kết quả tốt hơn hoặc ngang với khi sử dụng lượng tử hóa vector (vector quantization) với intersection kernel.

Bảng 5.9: Bảng so sánh kết quả phân lớp sử dụng Sparse Coding so với sử dụng vector quantization (Kmeans) trên CALTECH-101

		30 mẫu học	15 mẫu học
SPM (M=400)	Vector quantization	73.46	65.34
	Sparse coding + linear kernel	73.54	-
	Sparse coding + intersection kernel	75.68	-
HSMK (M=400)	Vector quantization	75.59	67.91
	Sparse coding + linear kernel	77.15	-
	Sparse coding + intersection kernel	79.02	-
SPM (M=800)	Vector quantization	75.13	66.84
	Sparse coding + linear kernel	75.52	-
	Sparse coding + intersection kernel	76.96	-
HSMK (M=800)	Vector quantization	77.33	68.95
	Sparse coding + linear kernel	78.93	72.14
	Sparse coding + intersection kernel	**80.60**	**73.44**
	Boimain et al. [4]	79.1	72.8

Kết luận và kiến nghị

Kết luận

Tôi đã đề xuất kernel tốt và hiệu quả được gọi là hierarchical spatial matching kernel (HSMK). HSMK sử dụng mô hình thô mịn (coarse to fine – C2F) trên vùng con để cải thiện spatial pyramid matching kernel (SPMK), HSMK mô tả vùng con tốt hơn dựa trên nhiều thông tin hơn của các vùng lân cận thông qua một chuỗi các độ phân giải (resolution) khác nhau, do vậy có thể mô tả được thông tin tổng quát ở resolution thô, cũng như thông tin chi tiết của vùng con ở độ phân giải (resolution) mịn hơn. Thêm nữa, kernel HSMK có khả năng xử lý tốt trên tập hợp các đặc trưng không thứ tự như SPMK và pyramid matching kernel (PMK) cũng như các tập hợp có số phần tử khác nhau. Sự kết hợp của kernel đề xuất với đặc trưng cục bộ có phân bố dày (dense local feature) cho thấy đạt được sự hiệu quả rất cao. Mô hình trên cho phép đạt kết quả ít nhất là tương ứng hoặc kết quả tốt nhất (state-of-the-art) so với các cách tiếp cận khác tồn tại trên nhiều loại cơ sở dữ liệu từ phân loại đối tượng như Oxford Flower, CALTECH-101, CALTECH-256, đến các cơ sở dữ liệu phân loại cảnh như MIT Scene, UIUC Scene. Hơn nữa, phương pháp đề xuất đơn giản bởi vì nó chỉ sử dụng một loại đặc trưng cục bộ với SVM phi tuyến, trong khi cac phương pháp tiếp cận khác gần đây phức tạp hơn rất nhiều mà dựa trên multiple kernel learning (MKL) hoặc sự kết hợp của nhiều loại đặc trưng (feature combinations).

Trên các cơ sở dữ liệu chuẩn về phân loại đối tượng và phân loại cảnh, cách tiếp cận đề xuất cho kết quả tốt hơn SPMK. Thêm nữa, SPMK là một thành phần quan trọng trong nhiều hệ thống đạt kết quả tốt nhất hiện nay, ví dụ như dùng trong việc xây dựng các kernel cơ bản trong mô hình học MKL. Điều này có nghĩa là ta có thể thay thế SPMK bằng HSMK để tăng độ chính xác của hệ thống được xây dựng dựa trên các kernel cơ bản.

Khi sử dụng biểu diễn thưa (Sparse coding) thay cho lượng tử hóa vector (vector quantization) thì tính hiệu quả của HSMK được cải thiện thêm nữa, có thể đạt kết

quả tối ưu trên cơ sở dữ liệu CALTECH-101 (cơ sở dữ liệu quan trong cho việc đánh giá phân loại ảnh).

Kiến nghị

Nghiên cứu về mặt lý thuyết sự ảnh hưởng của mô hình thô mịn (coarse to fine – C2F) cho việc biểu diễn ảnh và xây dựng kernel.

Nghiên cứu về lý thuyết sự tác động của biểu diễn thưa (sparse coding) lên nhiều độ phân giải (multi-resolution) trong HSMK.

Danh mục công trình của tác giả[6]

Trong nước:

[1] Lê Thanh Tâm, Trần Thái Sơn, Seiichi Mita (2009), "Phát hiện và phân loại biển báo giao thông dựa trên SVM trong thời gian thực," *Hội nghị Công Nghệ Thông Tin và Truyền Thông (ICTFIT)*, Thành phố Hồ Chí Minh, Việt Nam.

Quốc tế:

[1] Tam T. Le, Son T. Tran, Seiichi Mita, Thuc D. Nguyen (2010), "Realtime Traffic Sign Detection Using Color and Shape-Based Features," *The 2nd Asian Conference on Intelligent Information and Database Systems*, Lecture Notes in Artificial Intelligence 5991, Hue, Vietnam.

[2] Tam T. Le, Yousun Kang, Akihiro Sugimoto, Son T. Tran, Thuc D. Nguyen (2011), "Hierarchical Spatial Matching Kernel for Image Categorization," *International Conference on Image Analysis and Recognition (ICIAR)*, Burnaby, BC, Canada. (**accepted**)

[6] Các bài báo trên được lưu trữ trên trang web nghiên cứu cá nhân: http://sites.google.com/site/lttamvn

Tài liệu tham khảo

Tiếng Anh

[1] N. Aronszajn. (1950), “Theory of reproducing kernels,” *Transaction American Mathematics Society*, vol. 68:337-404.

[2] S. Boyd, and L. Vandenberghe. (2004), “Convex Optimization,” *Cambridge University Press*, Cambridge, England.

[3] C. Cortes, and V. Vapnik. (1995), “Support Vector Networks,” in *Machine Learning*, vol. 10(3):273-297.

[4] O Boiman, E Shechtman, and M Irani. (2008),"In defense of nearest-neighbor based image classiffication," in *CVPR*.

[5] N Dalal and B Triggs. (2005),"Histograms of oriented gradients for human detection," *in Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[6] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. (2004),"Visual categorization with bags of keypoints," *in ECCV International Workshop on Statistical Learning in Computer Vision*.

[7] L Fei-Fei, R Fergus, and P Perona. (2004),"Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories," *in Workshop on Generative-Model Based Vision*.

[8] Li Fei-Fei and P Perona. (2005),"A bayesian hierarchical model for learning natural scene categories," *in Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Washington, DC, USA, p. Volume 2.

[9] P. Felzenszwalb, D. Mcallester, and D. Ramanan. (June 2008),"A discriminatively trained, multiscale, deformable part model," *in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska.

[10] R Fergus, P Perona, and A Zisserman. (2003),"Object class recognition by unsupervised scale-invariant learning," *in IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2.

[11] M Fischler and R Elschlager. (1973)"The representation and matching of pictorial structures," *IEEE Transactions on Computers*, pp. 67-92.

[12] P Gehler and S Nowozin. (2009),"On feature combination for multiclass object classiffication," *in ICCV*, pp. 221-228.

[13] K Grauman and T Darrell. (2005),"The pyramid match kernel: discriminative classiffication with sets of image features," *in ICCV*, pp. 1458-1465.

[14] G Griffin, A Holub, and P Perona (2007)"Caltech-256 object category dataset," *Technical Report 7694*, California Institute of Technology, USA.

[15] M Johnson. (2008), "Semantic Segmentation and Image Search," *PhD Thesis*, University of Cambridge, UK.

[16] M Kloft, U Brefeld, P Laskov, and S Sonnenburg. (2008),"Non-sparse multiple kernel learning," *in NIPS Workshop on Kernel Learning: Automatic Selection of Kernels*.

[17] R.I Kondor and T Jebara. (2003),"A kernel between sets of vectors," *in ICML*, pp. 361-368.

[18] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan. (2004), “Learning the Kernel Matrix with Semidefinite Programming,” in *Journal of Machine Learning Research*, vol. 5:27-72.

[19] S. Lazebnik, C. Schmid, and J. Ponce. (2006),"Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *in CVPR*, vol. 2.

[20] David G Lowe. (2004)"Distinctive Image Features from Scale-Invariant keypoints," *International Journal of Computer Vision*, vol. 60 (2): pp 91-110.

[21] David G Lowe. (1999),"Object recognition from local scale-invariant features," *in International Conference on Computer Vision*, Corfu, Greece.

[22] J Mairal, F Bach, J Ponce, and G Sapiro. (2009),"Online dictionary learning for sparse coding," *in ICML*, pp. 689-696.

[23] S Maji, A Berg, and J Malik. (2008),"Classiffication using intersection kernel support vector machines is efficient," *in CVPR*, pp. 1-8.

[24] F Moosmann, B Triggs, and F Jurie. (2008),"Randomized clustering forests for building fast and discriminative visual vocabularies," *in NIPS Workshop on Kernel Learning: Automatic Selection of Kernels*.

[25] M.E Nilsback and A Zisserman. (2006),"A visual vocabulary for ower classiffication," *in CVPR*, vol. 2, pp. 1447-1454.

[26] M.E Nilsback and A Zisserman. (2008),"Automated ower classiffication over a large number of classes," in *ICVGIP*.

[27] A Oliva and A Torralba. (2001)"Modeling the shape of the scene: A holistic representation of the spatial envelope," *in IJCV*, pp. 145-175.

[28] A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu. (2008) “SimpleMKL,” in *Journal of Machine Learning Research*, vol. 9:2491-2521.

[29] B Scholkopf, and A.J Smola. (2002) “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond,” *MIT Press*, Cambridge, MA, USA.

[30] S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf. (2006) “Large Scale Multiple Kernel Learning,” in *Journal of Machine Learning Research*.

[31] V. Vapnik, and A. Lerner. (1963), “Pattern recognition using generalized portrait method”, *in Automation and Remote Control*, 24, 774-780.

[32] V. Vapnik, and A. Chervonenkis. (1964), “A note on one class of perceptrons”, *in Automation and Remote Control*, 25.

[33] M. Varma and D. Ray. (2007),"Learning the discriminative power-invariance trade-off," *in IEEE 11th International Conference on Computer Vision*.

[34] S.C Wang and Y.C.F Wang. (2010),"A multi-scale learning framework for visual categorization," *in ACCV*.

[35] L Yang, R Jin, R Sukthankar, and F Jurie. (2008),"Unifying discriminative visual code-book generation with classiffier training for object category recognition," *in CVPR*, Los Alamitos, CA, USA, vol. 0, pp. 1-8.

[36] J Yang, K Yu, Y Gong, and T Huang. (2009),"Linear spatial pyramid matching using sparse coding for image classiffication," *in CVPR*, pp. 1794-1801.

[37] Q Yuan, A Thangali, V Ablavsky, and S Sclaroff. (2008),"Multiplicative kernels: Object detection, segmentation and pose estimation," *in Computer Vision and Pattern Recognition*.