

# HỆ TRUY VẤN ẢNH SỬ DỤNG CHỮ KÝ MỜ VÀ CÂY FS-TREE

VĂN THẾ THÀNH

*Trung tâm Công nghệ Thông tin - Trường ĐH Công nghiệp Thực phẩm Tp.HCM*

## TÓM TẮT

Việc truy vấn ảnh sẽ tìm ra các hình ảnh tương tự về nội dung với hình ảnh cần truy vấn. Vấn đề đặt ra là cần xây dựng một hệ thống tìm kiếm các hình ảnh tương tự nhưng vẫn đảm bảo về tốc độ và không gian truy vấn. Để giải quyết bài toán tìm kiếm ảnh tương tự này, bài báo sẽ trích xuất vùng đặc trưng màu sắc trên mỗi hình ảnh dựa trên phương pháp Harris-Laplace, đồng thời xây dựng cấu trúc chữ ký mờ để mô tả các đặc trưng về nội dung màu sắc của hình ảnh theo chuẩn MPEG7. Trên cơ sở chữ ký mờ, bài báo tiến hành đánh giá độ đo tương tự giữa các chữ ký mờ của hình ảnh qua độ đo mờ Hamming, từ đó đánh giá độ tương tự giữa các hình ảnh. Hơn nữa, nhằm gia tăng tốc độ truy vấn, bài báo đề xuất cấu trúc dữ liệu cây FS-Tree (fuzzy S-Tree) để lưu trữ các chữ ký mờ dựa trên độ đo FHD (fuzzy Hamming distance). Tiếp theo, bài báo xây dựng thuật toán truy vấn hình ảnh trên cây FS-Tree và kết xuất ra các hình ảnh tương tự với hình ảnh cần truy vấn. Sau cùng, bài báo đưa ra mô hình thực nghiệm và đánh giá phương pháp dựa trên dữ liệu hình ảnh mẫu Corel gồm 10,800 hình ảnh.

*Từ khóa:* Truy vấn ảnh, FHD, FS-Tree, Chữ ký mờ

## IMAGE RETRIEVAL SYSTEM USING FUZZY SIGNATURE AND FS-TREE

### ABSTRACT

To query image will find out the similar images in content with the query image. The problem need to build the image retrieval system to find the similar images which ensure the speed and space to query. In order to solve this problem, the paper extracts the color feature regions in each image on the base of Harris-Laplace detector, at that time to build the fuzzy signature structure to describe the feature region in color content of image with the MPEG7 standard. According to the fuzzy signature, the paper evaluates the similar measure between the fuzzy signatures of images through the Hamming fuzzy measure, from there to assess the similarity between the images. Moreover, in order to speed up query image, the paper provides the data structure FS-Tree (fuzzy S-Tree) to store the fuzzy signatures on the base of FHD measure (fuzzy Hamming distance). Next, the paper builds the image retrieval algorithm in FS-Tree and shows the similar images with the query image. Finally, the paper gives the experimental model and assesses the propose method on the base of the Corel's sample image dataset which have 10,800 color images.

*Keywords:* Image Retrieval, FHD, FS-Tree, Fuzzy Signature

## 1. Giới thiệu

Tìm kiếm hình ảnh trong một tập lớn các hình ảnh là một bài toán khó. Một cách giải quyết là gán nhãn các hình ảnh [6, 7] nhưng sẽ tốn nhiều chi phí, tiêu tốn nhiều thời gian và không khả thi cho nhiều ứng dụng khác nhau. Hơn nữa, quá trình xử lý gán nhãn phụ thuộc vào ngữ nghĩa mô tả hình ảnh. Vì vậy hệ truy vấn ảnh dựa trên nội dung được phát triển nhằm rút trích các thuộc tính thị giác để mô tả nội dung của hình ảnh [6, 8]. Một số hệ thống truy vấn ảnh số đã xây dựng như: QBIC, ADL, DBLP, Virage, Alta Vista, SIMPLY city,...

Các công trình về truy vấn hình ảnh dựa trên nội dung như: Hệ truy vấn ảnh dựa trên histogram màu [6], lượng tử hoá và so sánh độ tương tự của hình ảnh dựa trên histogram màu [7], độ đo tương tự hình ảnh dựa trên việc kết hợp màu sắc và cấu trúc hình ảnh [9], truy vấn hình ảnh dựa trên màu sắc [10], truy vấn hình ảnh dựa trên độ tương tự của hình ảnh [8], truy vấn ảnh dựa trên histogram và cấu trúc hình ảnh [11], kỹ thuật truy vấn ảnh VBA (Variable-Bin Allocation) dựa trên chữ ký dạng chuỗi bit nhị phân và cây chữ ký S-Tree [8], lượng tử hoá màu sắc để giảm số chiều không gian màu sắc [12],...

Trong cách tiếp cận của bài báo sẽ tạo ra chữ ký mờ của một hình ảnh, là cách mô tả trừu tượng về phân bố màu sắc của hình ảnh. Nội dung của bài báo sẽ hướng đến việc truy vấn hiệu quả các “*hình ảnh tương tự*” trong một hệ thống dữ liệu lớn về hình ảnh. Trong bài báo này sẽ tiếp cận việc mô tả ngữ nghĩa về mặt nội dung của hình ảnh thông qua chữ ký mờ, đồng thời xây dựng lưu trữ chữ ký này lên cây FS-Tree. Cấu trúc dữ liệu FS-Tree sẽ mô tả mối quan hệ giữa các chữ ký mờ, từ đó mô tả mối quan hệ giữa các nội dung của hình ảnh. Dựa trên việc mô tả mối quan hệ ngữ nghĩa nội dung hình ảnh của cấu trúc dữ liệu FS-Tree, bài báo sẽ tiến hành tìm ra các hình ảnh tương tự theo nội dung trên các cơ sở dữ liệu ảnh Corel. [16]

Bài báo thực hiện việc xây dựng hệ truy vấn ảnh tương tự dựa trên vùng đặc trưng cục bộ RBIR (region-based image retrieval). Trước hết, bài báo sẽ trích xuất các điểm đặc trưng dựa vào phương pháp Harris-Laplace, từ đó tạo ra các vùng đặc trưng cho hình ảnh. Dựa trên các vùng đặc trưng này bài báo sẽ tạo ra các chữ ký mờ và đánh giá độ tương tự của hình ảnh. Nhằm gia tăng tốc độ truy vấn, bài báo xây dựng cây FS-Tree lưu trữ các chữ ký nhị phân để từ đó xây dựng thuật toán truy vấn hình ảnh tương tự trên cây FS-Tree. Bài báo sẽ đóng góp được hai phần chính đó là giảm khối lượng không gian truy vấn và làm tăng tốc độ truy vấn các đối tượng ảnh trên một cơ sở dữ liệu ảnh lớn.

Đóng góp của bài báo trong việc xây dựng chữ ký mờ dựa trên histogram của hình ảnh và xây dựng độ tương tự giữa các hình ảnh dựa trên độ đo mờ Hamming. Qua đó, bài báo đóng góp thuật toán và phương pháp truy vấn ảnh tương tự dựa trên việc xây dựng cấu trúc dữ liệu cây FS-Tree. Mục tiêu của bài báo nhằm giảm không gian và làm tăng tốc độ truy vấn ảnh trên dữ liệu ảnh lớn.

## 2. Các kiến thức cơ sở

### 2.1. Chữ ký nhị phân

Chữ ký nhị phân là vector bit được tạo thành bằng phép bấm các đối tượng dữ liệu [5], chữ ký sẽ có  $k$  bit 1 và  $(m-k)$  bit 0 trong dãy bit  $[1..m]$ , với  $m$  là chiều dài của chữ ký. Các đối tượng dữ liệu và các đối tượng truy vấn được mã hóa trên cùng một thuật toán mã hóa chữ ký. Khi các bit trong chữ ký đối tượng dữ liệu  $s_i$  hoàn toàn phủ các bit trong chữ ký truy vấn  $s_q$ , thì đối tượng dữ liệu này là một ứng viên thỏa câu truy vấn. Theo tài liệu [5], kết quả truy vấn sẽ có ba trường hợp xảy ra, gồm:

(1) Đối tượng dữ liệu phù hợp với câu truy vấn. Khi đó mọi bit trong  $s_q$  được phủ bởi các bit trong chữ ký  $s_i$  của đối tượng dữ liệu (nghĩa là  $s_q \wedge s_i = s_q$ );

(2) Đối tượng không phù hợp với câu truy vấn (nghĩa là  $s_q \wedge s_i \neq s_q$ );

(3) Chữ ký được đối sánh và cho ra một kết quả phù hợp, nhưng đối tượng dữ liệu không phù hợp với điều kiện tìm kiếm trong câu truy vấn. Để loại ra trường hợp này, các đối tượng phải được kiểm tra sau khi các chữ ký đối tượng được đối sánh phù hợp.

### 2.2. Chữ ký mờ

Chữ ký mờ  $F$  có chiều dài  $m$  là một vector  $(f_1, f_2, \dots, f_m)$ , với  $f_i \in [0, 1]$ ,  $i = 1, \dots, m$  [2].

Phép kết nối chữ ký mờ  $F^i$  và  $F^j$  là một chữ ký mờ: [2]

$$F^i \wedge F^j = (f_1^i \wedge f_1^j, f_2^i \wedge f_2^j, \dots, f_m^i \wedge f_m^j), \text{ với } f_r^i \wedge f_r^j = \min\{f_r^i, f_r^j\}, r = 1, \dots, m$$

Phép kết hợp chữ ký mờ  $F^i$  và  $F^j$  là một chữ ký mờ: [2]

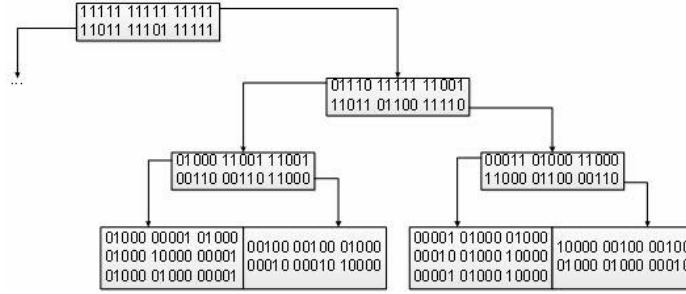
$$F^i \vee F^j = (f_1^i \vee f_1^j, f_2^i \vee f_2^j, \dots, f_m^i \vee f_m^j), \text{ với } f_r^i \vee f_r^j = \max\{f_r^i, f_r^j\}, r = 1, \dots, m$$

### 2.3. Cây chữ ký S-Tree

S-Tree [5, 8] là cây nhiều nhánh cân bằng, mỗi một nút của S-Tree nhiều cặp phần tử  $\langle s, p \rangle$ , với  $s$  là một chữ ký nhị phân,  $p$  là con trỏ tham chiếu đến nút con. Nút gốc của cây chứa ít nhất là hai cặp phần tử và nhiều nhất là  $M$  cặp phần tử  $\langle s, p \rangle$ . Mỗi nút trong của cây chứa ít nhất là  $m$  cặp phần tử  $\langle s, p \rangle$  và nhiều nhất là  $M$  cặp phần tử  $\langle s, p \rangle$ , với  $1 \leq m \leq M/2$ . Mỗi một nút lá của cây S-Tree chứa tập các phần tử  $\langle s, oid \rangle$ , với  $oid$  là định danh của đối tượng,  $s$  là chữ ký của đối tượng tương ứng. Mỗi chữ ký tại một nút cha là tổ hợp tất cả các chữ ký của nút con. Chiều cao tối đa của cây S-Tree có  $n$  chữ ký sẽ là  $h = \lceil \log_m n - 1 \rceil$ .

Quá trình xây dựng cây S-Tree được thực hiện dựa trên thao tác chèn và tách nút. Tại thời điểm bắt đầu, cây S-Tree chỉ chứa một nút lá rỗng, sau đó từng chữ ký sẽ được chèn

vào trong cây S-Tree. Khi nút lá  $v$  trở nên đầy sẽ được tách thành hai nút, đồng thời nút cha  $v_{paren}$  sẽ được tạo ra (nếu chưa tồn tại) và hai chữ ký mới sẽ được đặt vào nút  $v_{paren}$ .



Hình 1. Một ví dụ về cây S-Tree [8]

### 2.4. Độ đo mờ Hamming

Cho hai vector giá trị thực  $n$ -chiều  $x$  và  $y$ , gọi tập mờ về độ khác nhau là  $D_\alpha(x, y)$ , với hàm thuộc là  $\mu_{D_\alpha(x,y)} = 1 - e^{-\alpha(x-y)}$ . Theo tài liệu [4], khoảng cách mờ Hamming FHD giữa  $x$  và  $y$  được ký hiệu là  $FHD_\alpha(x, y)$  là lực lượng mờ của tập mờ  $D_\alpha(x, y)$  và có hàm thuộc ứng với tham số  $\alpha$  là:  $\mu_{FHD(x,y)}(\alpha) : \{0, 1, \dots, n\} \rightarrow [0, 1]$ . Mức độ khác nhau của  $x$  và  $y$  tại thành phần thứ  $k$ , ứng với hằng số điều chỉnh  $\alpha$  sẽ là:  $\mu_{FHD(x,y)}(k; \alpha) = \mu_{Card(D_\alpha(x,y))}(k)$ , với  $k \in \{0, 1, \dots, n\}, n = |Support(D_\alpha(x, y))|$ .

### 2.4. Trích xuất vùng đặc trưng của hình ảnh

Để trích xuất đặc trưng thị giác của hình ảnh, bước đầu tiên cần phải chuẩn hóa kích thước hình ảnh, tức là chuyển đổi các hình ảnh đầu vào có kích thước khác nhau trở thành hình ảnh có kích thước  $k \times k$  để từ đó rút trích các đặc trưng màu sắc của hình ảnh. Vì ảnh theo chuẩn JPEG được mô tả trên không gian màu YCbCr, do đó cần sử dụng không gian màu YCbCr để trích xuất thông tin đặc trưng của ảnh. Gọi  $Y, Cb, Cr$  lần lượt là cường độ sáng, thành phần màu Blue, thành phần màu Red. Theo tài liệu [13], phép chuyển đổi từ không gian màu RGB sang không gian màu YCbCr như sau:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.996 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}, R, G, B \in [0, 1]$$

Theo tài liệu [14], [15], phép biến đổi Gaussian theo hệ thống thị giác của con người như sau:

$$L(x, y, \delta_d) = \frac{1}{10} [6.G(x, y, \delta_d) * Y + 2.G(x, y, \delta_d) * Cb + 2.G(x, y, \delta_d) * Cr] \text{ với}$$

$$G(x, y, \delta_d) = \frac{1}{\sqrt{2\pi} \cdot \delta_d} \cdot \exp\left(-\frac{x^2 + y^2}{2 \cdot \delta_d^2}\right)$$

Cường độ đặc trưng  $I_0(x, y)$  cho ảnh màu được tính theo phương trình:

$$I_0(x, y, \delta_l, \delta_d) = \text{Det}(M(x, y, \delta_l, \delta_d)) - \alpha \cdot \text{Tr}^2(M(x, y, \delta_l, \delta_d))$$

Trong đó,  $\text{Det}(\bullet), \text{Tr}(\bullet)$  lần lượt là định thức và vết của ma trận,  $M(x, y, \delta_l, \delta_d)$  là ma trận moment bậc hai, được định nghĩa như sau:

$$M(x, y, \delta_l, \delta_d) = \delta_d^2 \cdot G(\delta_l) * \begin{bmatrix} L_x^2 & L_x L_y \\ L_x L_y & L_y^2 \end{bmatrix}$$

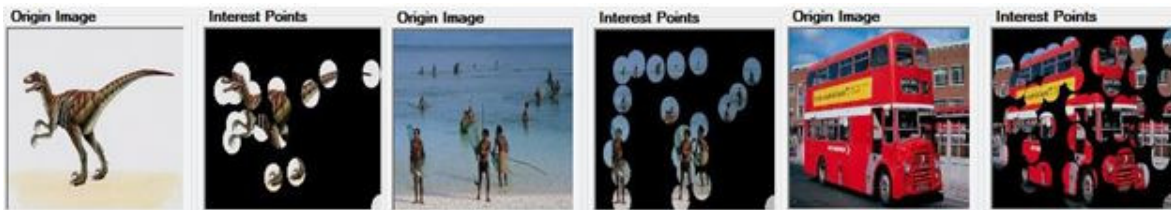
Trong đó,  $\delta_l, \delta_d$  là các giá trị vi phân,  $L_\alpha$  là đạo hàm theo hướng  $\alpha$ . Các điểm đặc trưng của ảnh màu được rút trích theo công thức:

$$I_0(x, y, \delta_l, \delta_d) > I_0(x', y', \delta_l, \delta_d), \text{ với } x', y' \in A$$

$I_0(x, y, \delta_l, \delta_d) \geq \theta$ , với  $A$  là tập các điểm láng giềng của  $(x, y)$  và  $\theta$  là giá trị ngưỡng.

Tập các đường tròn đặc trưng  $O_l = \{o_l^1, o_l^2, \dots, o_l^n\}$  có tâm là các điểm đặc trưng và tập bán kính của đường tròn đặc trưng  $R_l = \{r_l^1, r_l^2, \dots, r_l^n\}$ .

Các giá trị của bán kính đặc trưng được trích xuất theo phương pháp LoG (Laplace-of-Gaussian) và có miền giá trị là  $[0, \min(M, N)/2]$ , với  $M, N$  là chiều cao và chiều rộng của hình ảnh.



Hình 2. Trích xuất vùng đặc trưng trên ảnh theo phương pháp Harris-Laplace

### 3. Xây dựng cấu trúc dữ liệu và thuật toán truy vấn ảnh

#### 3.1. Tạo chữ ký mờ

Mỗi vùng đặc trưng  $o_l^i \in O_l$  của hình ảnh  $I$  sẽ được tính histogram dựa trên dải màu chuẩn  $C$ , thực hiện phương pháp phân cụm dựa trên độ đo Euclide trong không gian màu RGB để phân loại các màu sắc của từng điểm ảnh trên hình ảnh. Gọi  $p$  là một điểm trên ảnh  $I$  và có vector giá trị màu trong không gian RGB là  $V_p = (R_p, G_p, B_p)$ . Gọi  $V_m = (R_m, G_m, B_m)$  là vector màu thuộc tập dải màu chuẩn  $C$ , sao cho:  $V_m = \min\{\|V_p - V_i\|, V_i \in C\}$ . Khi đó, tại điểm  $p$  sẽ được chuẩn hóa theo vector màu  $V_m$ . Theo thực nghiệm, bài báo sẽ sử dụng tập dải màu theo chuẩn MPEG7 để tính histogram cho các ảnh màu trên dữ liệu ảnh Corel.

Bảng 1. Danh mục màu theo chuẩn MPEG7

Index	Color	R	G	B	Index	Color	R	G	B
0	Black	0	0	0	13	Plum	146	109	0
1	Sea Green	0	182	0	14	Teal	146	182	170
2	Light Green	0	255	170	15	Dark Red	182	0	0
3	Olive Green	36	73	0	16	Magenta	182	73	170
4	Aqua	36	146	170	17	Yellow Green	182	182	0
5	Bright Green	36	255	0	18	Flouro Green	182	255	170
6	Blue	73	36	170	19	Red	219	73	0
7	Green	73	146	0	20	Rose	219	146	170
8	Turquoise	73	219	170	21	Yellow	219	255	0
9	Brown	109	36	0	22	Pink	255	36	170
10	Blue Gray	109	109	170	23	Orange	255	146	0
11	Lime	109	219	0	24	White	255	255	255
12	Lavender	146	0	170					

Quá trình tạo chữ ký mờ cho mỗi hình ảnh được thực hiện như sau:

*Bước 1.* Chọn tập dải màu  $C = \{c_1, c_2, \dots, c_n\}$  theo chuẩn MPEG7 làm cơ sở cho việc tính histogram của hình ảnh, gọi  $I$  là ảnh màu cần tính histogram. Lượng tử hoá các màu chiếm ưu thế của ảnh  $I$  sẽ được tập màu  $C_I = \{c'_1, c'_2, \dots, c'_n\}$ , vector histogram của ảnh  $I$  sẽ là  $H_I = \{h'_1, h'_2, \dots, h'_n\}$ .

*Bước 2.* Thực hiện việc chuẩn hoá histogram của ảnh  $I$  trên dải màu  $C$  sẽ được vector histogram chuẩn hoá  $H = \{h_1, h_2, \dots, h_n\}$ , với mỗi giá trị  $h_i \in [0,1]$  được chuẩn hoá:  $h_i = h'_i / \sum_j h'_j$  nếu  $c_i \in C \cap C_I$ , ngược lại  $h_i = 0$ .

*Bước 3.* Mỗi màu  $c'_j$  sẽ được mô tả thành một chữ ký mờ có chiều dài  $m$  là  $f_1^j f_2^j, \dots, f_m^j$ . Do đó, chữ ký mờ của ảnh  $I$  là:  $f_1^1 f_2^1, \dots, f_m^1 f_1^2 f_2^2, \dots, f_m^2 \dots f_1^n f_2^n, \dots, f_m^n$ , trong đó:

$$f_i^j = \begin{cases} h_i & i = \lceil h_i \times \frac{i}{m} \times 100 \rceil \\ 0 & i \neq \lceil h_i \times \frac{i}{m} \times 100 \rceil \end{cases}$$

Đặt  $F^j = f_1^j f_2^j \dots f_m^j$ , chữ ký mờ của ảnh  $I$  sẽ là:  $FuzzySig = F^1 F^2 \dots F^n$

### 3.2. Độ đo tương tự FHD

Mỗi chữ ký mờ  $FuzzySig_I = F^1 F^2 \dots F^n$  của hình ảnh  $I$  sẽ được chuyển thành vector  $V_I = (v_1, v_2, \dots, v_n)$ , trong đó  $v_i = weight(F^i) = \sum_{k=1}^m w_k^i$ , với  $F^i = f_1^i f_2^i \dots f_m^i$ , và:

$$w_k^i = \begin{cases} 0 & f_k^i = 0 \\ f_k^i + \frac{k}{m} \times 100 & f_k^i \neq 0 \end{cases}$$

Gọi  $J$  là hình ảnh cần tính độ tương tự so với ảnh  $I$ , do đó cần tính khoảng cách mờ Hamming giữa hai vector  $V_I = (v_1^I, v_2^I, \dots, v_n^I)$  và  $V_J = (v_1^J, v_2^J, \dots, v_n^J)$ .

Khoảng cách mờ FHD là tập mờ lực lượng của tập mờ  $D_\alpha(V_I, V_J)$ , nghĩa là:

$$FHD_\alpha(V_I, V_J) = Card(D_\alpha(V_I, V_J)) = \sum_{i=0}^n i / (\mu_{Card(D_\alpha(V_I, V_J))}(i))$$

Trong đó,  $\mu_{Card(D_\alpha(V_I, V_J))}(i) = \mu(i) \wedge (1 - \mu(i+1)) = \min\{\mu(i), (1 - \mu(i+1))\}$ ,  $\mu(i)$  là giá trị lớn nhất thứ  $i$  của hàm thuộc  $\mu_i$  ứng với tập mờ  $D_\alpha(V_I, V_J) = \sum_{i=1}^n i / \mu_i$  và  $\mu(0) = 1, \mu(n+1) = 0$ . Khi đó, mức độ khác nhau của  $V_I$  và  $V_J$  trên  $k$  thành phần sẽ là:  $\mu_{FHD_\alpha(V_I, V_J)}(k, \alpha) = \mu_{Card(D_\alpha(V_I, V_J))}(k)$ , với  $k \in \{0, 1, \dots, n\}, n = |Support(D_\alpha(x, y))|$ .

### 3.3. Tạo cây FS-Tree

Nhằm giảm không gian và tăng tốc độ truy vấn, bài báo tiến hành xây dựng cây chữ ký FS-Tree lưu trữ các chữ ký mờ của hình ảnh. Mỗi một nút trong cây FS-Tree sẽ lưu trữ tập các phần tử  $\{\langle FuzzySig, next \rangle\}$ , với  $FuzzySig$  là chữ ký mờ và  $next$  là con trở tham chiếu đến nút con. Các nút lá sẽ lưu trữ các phần tử  $\{\langle FuzzySig, Oid \rangle\}$ , với  $FuzzySig$  là chữ ký mờ của mỗi hình ảnh và  $Oid$  là định danh của hình ảnh tương ứng. Quá trình tạo cây FS-Tree được thực hiện dựa trên thao tác chèn và tách nút trong cây [5, 8]. Thuật toán tạo cây chữ ký FS-Tree lưu trữ chữ ký mờ được thực hiện như sau:

**Input:** tập các chữ ký FS =  $\{\langle FuzzySig_i, Oid_i \rangle \mid i = 1, \dots, n\}$   
**Output:** cây chữ ký FS-Tree  
**Algorithm1.** Gen-FSTree( $S, Root$ )  
**Begin**  
  Bước 1.  $v = Root$ ;  
  **If**  $FS = \emptyset$  **then** STOP;  
  **Else** Chọn  $\langle FuzzySig, Oid \rangle \in S$  và  $S = S \setminus \langle FuzzySig, Oid \rangle$ ;  
  Qua bước 2;  
  Bước 2. **If**  $v$  là nút lá **then**  
    **begin**  
       $v = v \oplus \langle FuzzySig, Oid \rangle$ ;  
      UnionSignature( $v$ );  
      **If**  $v.count > M$  **then** SplitNode( $v$ );  
      Quay lại bước 1;  
    **end**  
  **Else**  
    **begin**  
      FHD( $SIG0 \rightarrow FuzzySig, Fuzzysig$ )  
      =  $\min\{FDH(SIGi \rightarrow FuzzySig, FuzzySig) \mid SIGi \in v\}$ ;  
       $v = SIG0 \rightarrow next$ ;  
      Quay lại bước 2;  
    **end**  
**End.**

Thuật toán *Algorithm1* sẽ lần lượt đưa các chữ ký *FuzzySig* từ tập chữ ký *FS* vào trong cây FS-Tree. Với mỗi chữ ký *FuzzySig* sẽ được chèn vào nút lá phù hợp, nếu nút lá đầy thì quá trình tách nút sẽ được thực hiện và cây FS-Tree sẽ tăng trưởng chiều cao theo hướng gốc của cây. Tại mỗi nút trong của cây FS-Tree, sẽ ưu tiên đi theo hướng có độ tương tự FHD nhiều hơn, quá trình này sẽ được duyệt cho đến khi tìm ra được nút lá phù hợp.

Ứng với mỗi chữ ký cần chèn sẽ duyệt qua đường đi có chiều cao  $h = \lceil \log_m n - 1 \rceil$ , với  $m$  là số chữ ký tối thiểu của một nút trong cây S-Tree. Gọi  $k$  là chiều dài của mỗi chữ ký, mỗi một nút trong của cây sẽ có tối đa là  $M$  chữ ký, vì vậy quá trình duyệt cây để tìm ra nút lá phù hợp sẽ có chi phí tối đa là  $k \times M \times \lceil \log_m n - 1 \rceil$ . Tuy nhiên, khi tìm ra nút lá phù hợp nhưng đã bị đầy, cần phải thực hiện quá trình tách nút, việc tách nút dựa trên cơ sở phép toán  $\alpha$ -seed,  $\beta$ -seed được thực hiện như sau:

**Input:** Nút  $v$   
**Output:** Cây FS-Tree sau khi thực hiện phép tách nút  
**Algorithm2.** SplitNode( $v$ )  
**Begin**  
 Tạo nút  $v_\alpha$  và  $v_\beta$  lần lượt chứa chữ ký  $\alpha$ -seed và  $\beta$ -seed  
 $v = v \setminus \{ \alpha\text{-seed}, \beta\text{-seed} \}$   
**For** ( $SIG_i \in v$ )  
**If** ( $FHD(SIG_i \rightarrow FuzzySig, \alpha\text{-seed}) < FHD(SIG_i \rightarrow Fuzzysig, \beta\text{-seed})$ ) **then**  
 $v_\alpha = v_\alpha \oplus SIG_i$ ;  
**Else**  $v_\beta = v_\beta \oplus SIG_i$ ;  
 $s_\alpha = \vee SIG_i^\alpha$ , với  $SIG_i^\alpha \in v_\alpha$ ;  $s_\beta = \vee SIG_i^\beta$ , với  $SIG_i^\beta \in v_\beta$ ;  
**If** ( $v_{parent} \neq \text{null}$ ) **then**  $v_{parent} = v_{parent} \oplus s_\alpha$ ;  $v_{parent} = v_{parent} \oplus s_\beta$ ;  
**If** ( $v_{parent}.count > M$ ) **then** SplitNode( $v_{parent}$ );  
**If** ( $v_{parent} = \text{null}$ ) **then** Root =  $\{ s_\alpha, s_\beta \}$ ;  
**End.**  
**Procedure** UnionSignature( $v$ )  
**Begin**  
 $s = \vee SIG_i$ , với  $SIG_i \in v$ ;  
**If** ( $v_{parent} \neq \text{null}$ ) **then**  
**begin**  
 $SIG_v = \{ SIG_i \mid SIG_i \rightarrow next = v, SIG_i \in v_{parent} \}$ ;  $v_{parent} \rightarrow (SIG_v \rightarrow FuzzySig) = s$ ;  
 UnionSignature( $v_{parent}$ );  
**end**  
**End.**

### 3.4. Thuật toán truy vấn ảnh trên cây FS-Tree

Sau khi lưu trữ chữ ký và định danh của hình ảnh trên cây chữ ký FS-Tree, quá trình truy vấn sẽ đưa ra các chữ ký của hình ảnh dựa trên việc duyệt cây FS-Tree với độ đo tương tự FHD. Sau khi tìm ra các chữ ký hình ảnh tương tự, dựa vào định danh của các



hình ảnh sẽ tìm ra cụ thể các hình ảnh tương tự với hình ảnh truy vấn. Do đó, bài toán cần thực hiện là tìm ra chữ ký của hình ảnh và định danh của hình ảnh tương ứng, quá trình truy vấn này được thực hiện như sau:

```

Input: chữ ký truy vấn FuzzySig và FS-Tree
Output: Tập chữ ký mờ và tập Oid tham chiếu đến hình ảnh tương ứng
Algorithm3. Search-Image-Sig(FuzzySig, FS-Tree)
Begin
  v = root; SIGOUT =  $\emptyset$ ; Stack =  $\emptyset$ ; Push(Stack, v);
  while(not Empty(Stack)) do
  begin
    v = Pop(Stack);
    If(v is not Leaf) then
    begin
      For(SIGi  $\in$  v and SIGi $\rightarrow$ FuzzySig  $\wedge$  FuzzySig = FuzzySig) do
        FHD(SIG0 $\rightarrow$ Fuzzysig, FuzzySig)
          = min{FHD(SIGi $\rightarrow$ FuzzySig, FuzzySig) | SIGi  $\in$  v};
        Push(Stack, SIG0  $\rightarrow$  next);
      end
    Else
      SIGOUT = SIGOUT  $\cup$  {<SIGi  $\rightarrow$  FuzzySig, Oidi> | SIGi  $\in$  v};
    end
  return SIGOUT;
End.

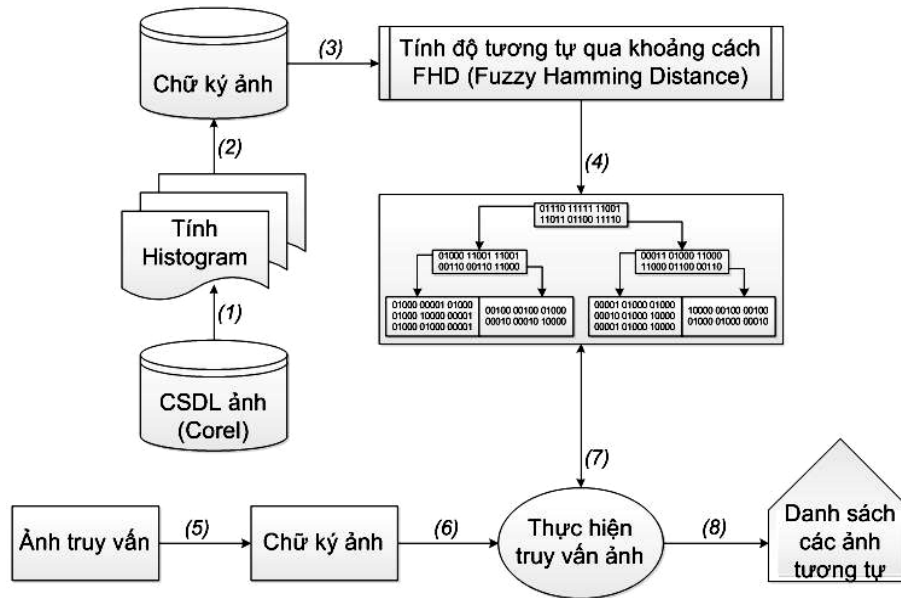
```

Vì FS-Tree là cây nhiều nhánh cân bằng, hơn nữa tại mỗi nút của cây sẽ được duyệt theo hướng tiếp theo có độ tương tự tốt nhất, do đó sẽ tốn chi phí tối đa duyệt cây là  $h = \lceil \log_m n - 1 \rceil$ . Quá trình tìm kiếm trên cây được thực hiện tương tự quá trình duyệt cây, do đó chi phí của quá trình truy vấn trên cây FS-Tree cũng sẽ là  $k \times M \times \lceil \log_m n - 1 \rceil$ , với  $k$  là chiều dài của mỗi chữ ký,  $m$  là số chữ ký tối thiểu,  $M$  là số chữ ký tối đa của một nút trong cây FS-Tree.

## 4. Ứng dụng thực nghiệm

### 4.1. Mô hình thực nghiệm

Quá trình xây dựng ứng dụng thực nghiệm gồm hai pha, pha thứ nhất sẽ thực hiện quá trình tiền xử lý nhằm chuyển đổi dữ liệu hình ảnh trở thành dạng chữ ký mờ và đưa vào cây chữ ký FS-Tree dựa trên độ đo tương tự FHD. Pha thứ hai sẽ thực hiện quá trình truy vấn, ứng với một hình ảnh cần truy vấn sẽ được chuyển đổi thành chữ ký mờ và sẽ được thực hiện truy vấn trên cây chữ ký FS-Tree dựa trên độ đo tương tự FHD. Sau khi tìm ra các chữ ký của hình ảnh tương tự, sẽ truy xuất hình ảnh cụ thể và sắp xếp theo thứ tự ưu tiên của độ đo tương tự FHD.



Hình 3. Mô hình hệ truy vấn ảnh dựa trên độ đo FHD và cây FS-Tree

**Pha 1:** Thực hiện tiền xử lý

*Bước 1.* Lượng tử hoá hình ảnh trong dữ liệu ảnh và chuyển thành histogram.

*Bước 2.* Chuyển đổi các histogram của hình ảnh thành chữ ký mờ.

*Bước 3.* Lần lượt tính khoảng cách FHD của các chữ ký mờ và chèn các chữ ký mờ vào cây FS-Tree.

**Pha 2:** Thực hiện truy vấn

*Bước 1.* Với mỗi hình ảnh truy vấn, sẽ được tính histogram và chuyển thành chữ ký mờ.

*Bước 2.* Thực hiện truy vấn chữ ký nhị phân trên cây FS-Tree gồm các chữ ký hình ảnh tương tự tại nút lá của cây qua độ đo FHD.

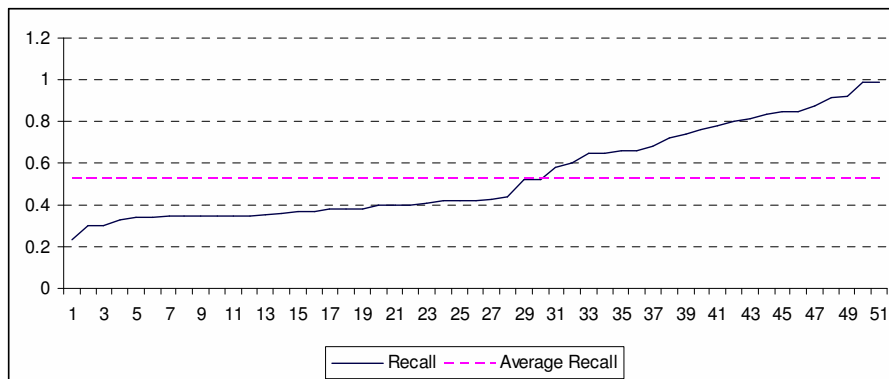
*Bước 3.* Sau khi có các hình ảnh tương tự, tiến hành sắp xếp theo độ tương tự từ cao đến thấp và đưa ra danh sách các hình ảnh trên cơ sở độ tương tự FHD.

**4.2. Kết quả thực nghiệm**

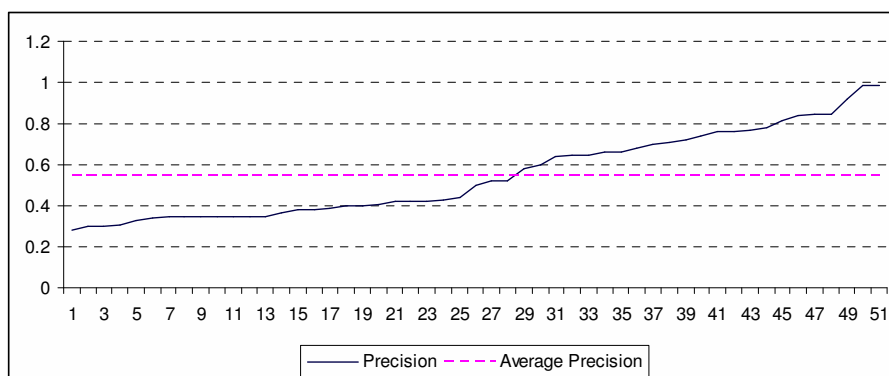
Quá trình thực nghiệm sẽ truy vấn trên dữ liệu mẫu Corel [16] gồm có 10,800 hình ảnh chia thành 80 chủ đề khác nhau. Với mỗi hình ảnh truy vấn sẽ được trích lọc trên dữ liệu ảnh Corel và tìm ra các hình ảnh có độ tương tự nhiều nhất với hình ảnh truy vấn, từ đó đối sánh với danh mục chủ đề hình ảnh nhằm đánh giá độ chính xác của phương pháp.

**Bảng 2.** Một số kết quả đánh giá mẫu về quá trình truy vấn ảnh trên dữ liệu Corel có 10,800 ảnh

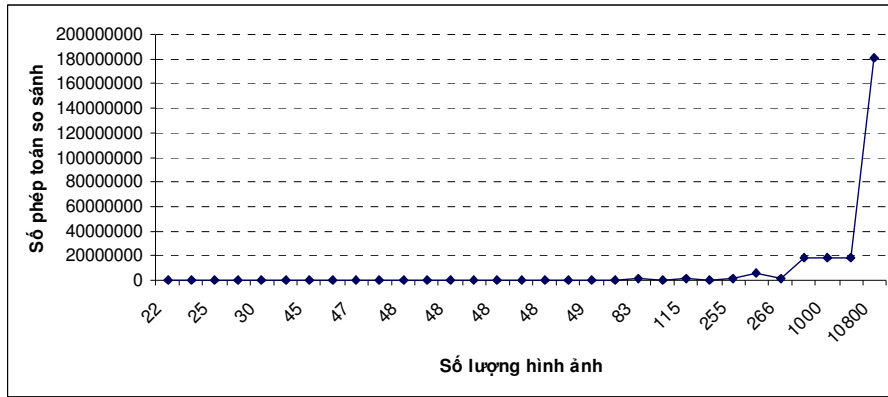
ID	ID Image	Recall	Precision	ID	ID Image	Recall	Precision	ID	ID Image	Recall	Precision
1	644000.jpg	0.990	0.990	18	303024.jpg	0.402	0.390	35	275007.jpg	0.420	0.420
2	569062.jpg	0.440	0.440	19	312001.jpg	0.430	0.430	36	275002.jpg	0.850	0.850
3	135032.jpg	0.580	0.580	20	318003.jpg	0.380	0.380	37	280014.jpg	0.920	0.920
4	113014.jpg	0.340	0.340	21	345002.jpg	0.833	0.500	38	387015.jpg	0.303	0.300
5	212001.jpg	0.365	0.840	22	377003.jpg	0.357	0.350	39	470083.jpg	0.383	0.310
6	280000.jpg	0.355	0.710	23	283000.jpg	0.850	0.850	40	487005.jpg	0.917	0.770
7	476034.jpg	0.330	0.330	24	435000.jpg	0.810	0.810	41	569015.jpg	0.520	0.520
8	40000.jpg	0.740	0.740	25	569012.jpg	0.800	0.640	42	569095.jpg	0.650	0.650
9	84000.jpg	0.350	0.350	26	435011.jpg	0.660	0.660	43	856089.jpg	0.233	0.700
10	114000.jpg	0.876	0.780	27	644099.jpg	0.990	0.990	44	409061.jpg	0.350	0.350
11	124000.jpg	0.370	0.370	28	225000.jpg	0.350	0.350	45	135072.jpg	0.680	0.680
12	131000.jpg	0.400	0.400	29	186000.jpg	0.410	0.410	46	136045.jpg	0.778	0.280
13	150000.jpg	0.350	0.350	30	343000.jpg	0.420	0.420	47	75050.jpg	0.760	0.760
14	167001.jpg	0.338	0.760	31	350000.jpg	0.600	0.600	48	113026.jpg	0.520	0.520
15	208000.jpg	0.400	0.400	32	473000.jpg	0.350	0.350	49	84049.jpg	0.420	0.420
16	221001.jpg	0.350	0.350	33	546000.jpg	0.650	0.650	50	221009.jpg	0.660	0.660
17	247000.jpg	0.380	0.380	34	817000.jpg	0.300	0.300	51	113090.jpg	0.720	0.720



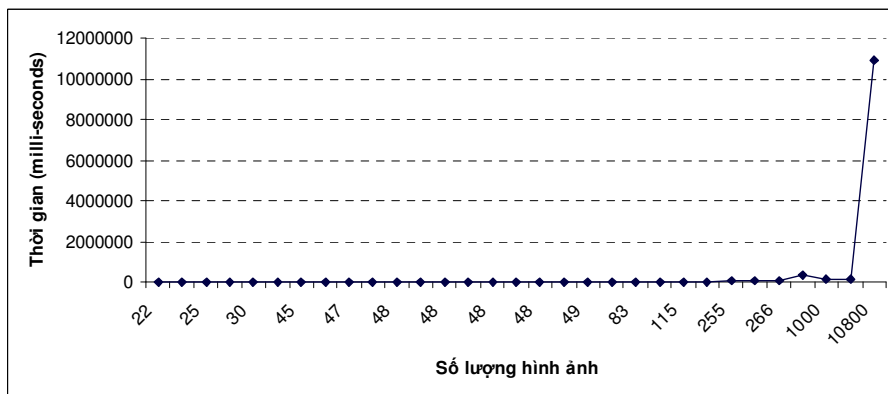
**Hình 4.** Độ chính xác truy vấn hình ảnh theo thực nghiệm



**Hình 5.** Khả năng truy vấn thành công hình ảnh theo thực nghiệm

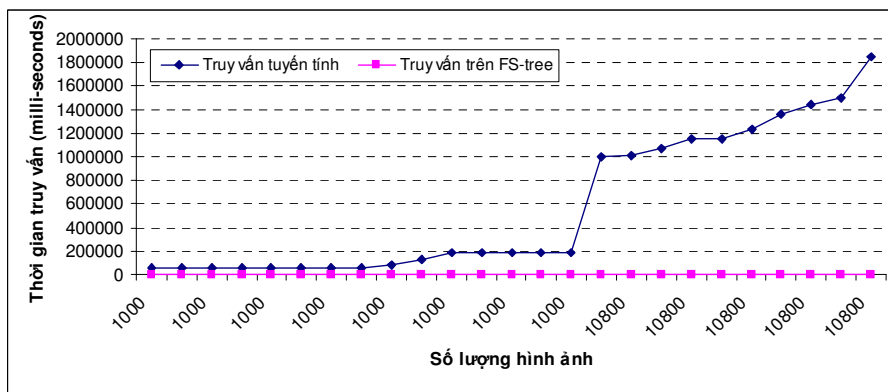


Hình 6. Số phép toán so sánh khi tạo cây FS-Tree

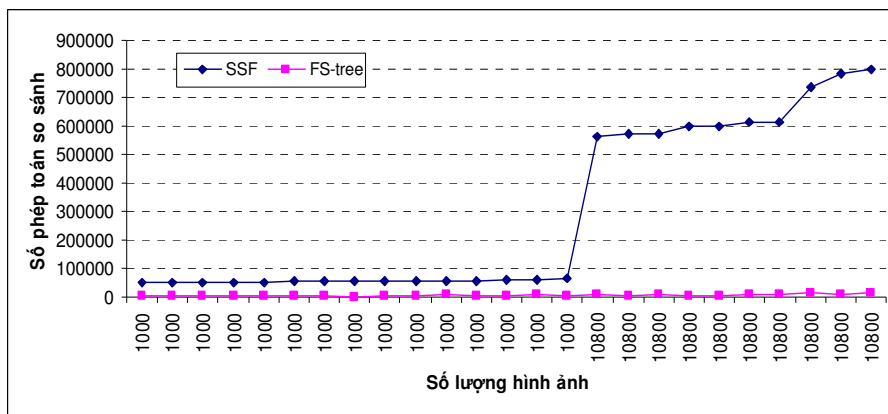


Hình 7. Thời gian tạo cây FS-Tree (tính theo milli giây)

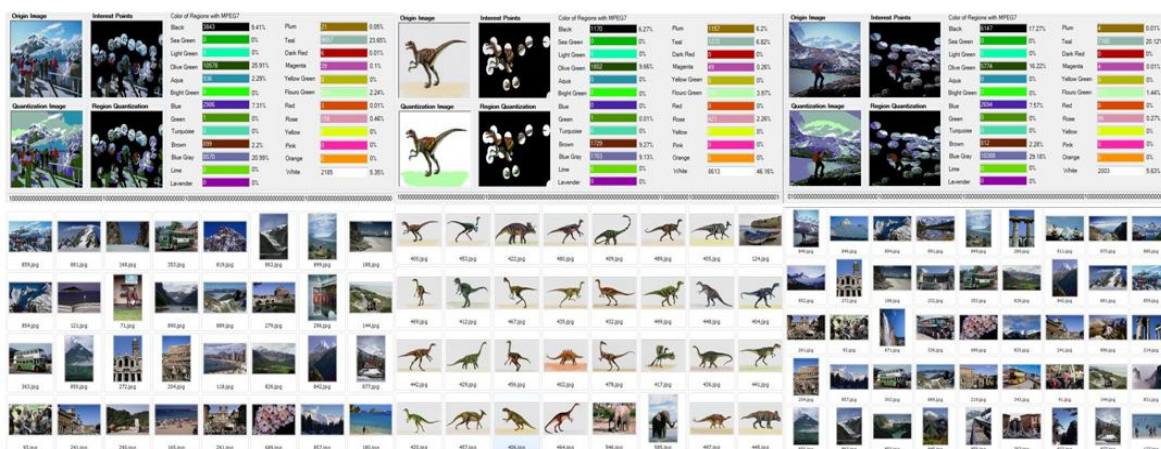
Các chữ ký nhị phân được đưa vào hai dạng cấu trúc truy vấn gồm cấu trúc tập tin chữ ký SSF (sequential signature file) và cây FS-Tree. Hình 8 và Hình 9 mô tả số liệu thực nghiệm về quá trình truy vấn hình ảnh tương tự trên dữ liệu ảnh Corel.



Hình 8. Thời gian (tính theo mili giây) truy vấn ảnh trên cây FS-Tree



Hình 9. Số phép toán so sánh khi thực hiện truy vấn ảnh



Hình 10. Một số kết quả truy vấn ảnh trên dữ liệu ảnh Corel

### 5. Kết luận

Trong bài báo đã đưa ra phương pháp đánh giá độ tương tự giữa hai hình ảnh dựa trên chữ ký nhị phân, đồng thời mô phỏng ứng dụng thực nghiệm trên dữ liệu ảnh phân loại của Corel. Theo thực nghiệm cho thấy sự hiệu quả của phương pháp đã đưa ra và hệ thống này thực thi tốt trên các hệ thống Online và Offline, đồng thời đã cải thiện đáng kể về tốc độ truy vấn hình ảnh rất nhiều lần so với phương pháp đối sánh trực tiếp cũng như phương pháp SSF (signature sequence file). Tuy nhiên, việc sử dụng các đặc trưng về màu sắc sẽ cho kết quả chưa chính xác theo ý nghĩa về nội dung hình ảnh. Do đó, hướng phát triển tiếp theo của bài báo sẽ thực hiện trích xuất đối tượng trên các hình ảnh, từ đó sẽ tiến hành xây dựng chữ ký nhị phân để mô tả các đối tượng cũng như mô tả nội dung cho hình ảnh, đồng thời tạo ra một cấu trúc dữ liệu mô tả mối quan hệ dựa trên độ tương tự về nội dung của các chữ ký với mỗi hình ảnh.

#### Lời cảm ơn

Chúng tôi xin cảm ơn Trung tâm Công nghệ Thông tin, trường Đại học Công nghiệp Thực phẩm Tp.HCM là nơi bảo trợ cho chúng tôi thực hiện nghiên cứu này.

## TÀI LIỆU THAM KHẢO

1. Heba Aboulmagd, Neamat EI-Gayar, Hoda Onsi (2008), “A new approach in content-based image retrieval using fuzzy”, *Springer Science, Telecommun Syst*, 40, 55-66
2. Vaclav Snasel, Zdenek Horak, Milos Kudelka, Ajith Abraham (2011), “Fuzzy Signatures Organized Using S-Tree”, *Proceedings of the IEEE International Conference on Digital Object Identifier*, 633-637
3. K. Konstantinidis, A. Gasteratos, I. Andreadis (2005), “Image retrieval based on fuzzy color histogram processing”, *Optics Communications*, 248(4-6), 375-386
4. Mircea M. Ionescu, Anca L. Ralescu (2005), “Image Clustering for a Fuzzy Hamming Distance Based CBIR System”, *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, 102-108
5. Yangjun Chen and Yibin Chen (2006), “On the Signature Tree Construction and Analysis”, *IEEE Trans. Knowl. Data Eng.*, 18(9), 1207-1224
6. Neetu Sharma. S, Paresh Rawat. S, Jaikaran Singh. S., (2011) “Efficient CBIR Using Color Histogram Processing”, *Signal & Image Processing: An Inter. Jour.*, 2(1), 94-112
7. Fazal Malik, Baharum Bin Baharudin (2012), “Feature Analysis of Quantized Histogram Color Features for Content-Based Image Retrieval Based on Laplacian Filter”, *International Conference on System Engineering and Modeling*, 34, 44-49
8. M. A. Nascimento, E. Tousidou, V. Chitkara, Y. Manolopoulos (2002), “Image indexing and retrieval using signature trees”, *Data & Knowledge Eng.*, 43(1), 57-77
9. Rahul Mehta, Nishchol Mishra, Sanjeev Sharma (2011), “Color - Texture based Image Retrieval System”, *International Journal of Computer Applications*, 24(5), 24-29
10. Gunja Varshney, Uma Soni (2011), “Color-Based Image Retrieval in Image Database System”, *International Journal of Soft Computing and Engineering*, 1(5), 31-35
11. Ch. Kavitha, M. Babu Rao, B.Prabhakara Rao, A.Govardhan (2011), “Image Retrieval based on Local Histogram and Texture Features”, *International Journal of Computer Science and Information Technology*, 2(2), 741-746
12. S-C Ch, C-K Y (2001), “A fast and novel technique for color quantization using reduction of color space dimensionality”, *Pattern Recognizer Letter*, 22(8), 845-956
13. G. H. Liu, J. Y. Yang (2013), “Content-based Image Retrieval Using Color Difference Histogram”, *Pattern Recognition*, 46, 188-198
14. X. Y. Wang, J. F. Wu, H. Y. Yang (2010), “Robust Image Retrieval Based on Color Histogram of Local Feature Regions”, *Springer Science, Multi. Tools Appl*, 49, 323-345
15. X.-Y. Wang et al., (2013), “Robust Color Image Retrieval Using Visual Interest Point Feature of Significant Bit-Planes”, *Digital Signal Processing*, 23(4), 1136-1153
16. Corel Corp, <http://www.corel.com>.