Các công trình nghiên cứu phát triển và ứng dụng
Công nghệ Thông tin và Truyền thông
https://www.ictmag.vn
Tập 2020, Số 1
Tháng 6, số đặc biệt CITA 2021

MỤC LỤC

Research, Development and Application on
Information and Communication Technology
https://www.ictmag.vn
Volume 2020, No.1
June, CITA 2021 Special Issue

TABLE OF CONTENTS

A Comparative Analysis of Filter-based Feature Selection Methods for Software Fault Prediction

Ha Thi Minh Phuong[1], Le Thi My Hanh[2], Nguyen Thanh Binh[1]
[1] The University of Danang - Vietnam-Korea University of Infomation and Communication Technology
[2] The University of Danang - University of Science and Technology

Tác giả liên hệ: Ha Thi Minh Phuong, Email: htmphuong@vku.udn.vn
Ngày nhận bài: 28/12/2020, ngày sửa chữa: 09/04/2021, ngày duyệt đăng: 19/05/2021
Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.969

Abstract: The rapid growth of data has become a huge challenge for software systems. The quality of fault prediction model depends on the quality of software dataset. High-dimensional data is the major problem that affects the performance of the fault prediction models. In order to deal with dimensionality problem, feature selection is proposed by various researchers. Feature selection method provides an effective solution by eliminating irrelevant and redundant features, reducing computation time and improving the accuracy of the machine learning model. In this study, we focus on research and synthesis of the Filter-based feature selection with several search methods and algorithms. In addition, five filter-based feature selection methods are analyzed using five different classifiers over datasets obtained from National Aeronautics and Space Administration (NASA) repository. The experimental results show that Chi-Square and Information Gain methods had the best influence on the results of predictive models over other filter ranking methods.

Keywords: *Feature selection, filter, wrapper, hybrid, embedded.*

Tên bài: **Phân tích so sánh các kỹ thuật lựa chọn đặc trưng dựa trên phương pháp lọc trong dự đoán lỗi phần mềm**

Tóm tắt: Sự phát triển mạnh mẽ của dữ liệu trong các hệ thống đã trở thành một thách thức lớn cho ngành công nghệ phần mềm. Chất lượng của các mô hình dự đoán lỗi phụ thuộc nhiều vào chất lượng của các tập dữ liệu. Trong đó, dữ liệu đa chiều là vấn đề chính ảnh hưởng đến hiệu quả của các mô hình dự đoán lỗi. Để giải quyết vấn đề dữ liệu đa chiều này, phương pháp lựa chọn đặc trưng đã được các nhà nghiên cứu tập trung khai thác trong những năm gần đây. Phương pháp lựa chọn đặc trưng cung cấp một giải pháp hiệu quả bằng cách loại bỏ các thuộc tính bị nhiễu, không liên quan và dư thừa từ đó góp phần giảm thời gian tính toán và cải thiện độ chính xác của mô hình học máy. Trong nghiên cứu này, tác giả tập trung phân tích và so sánh hiệu quả của các kỹ thuật lựa chọn đặc trưng dựa trên phương pháp lọc. Ngoài ra, chúng tôi tiến hành thực nghiệm để so sánh hiệu quả của năm kỹ thuật lựa chọn đặc trưng dựa trên phương pháp lọc trên các thuật toán phân loại khác nhau. Thực nghiệm được tiến hành trên các bộ dữ liệu thu được từ kho dữ liệu NASA. Các kết quả thực nghiệm cho thấy các phương pháp Chi-Square và Information Gain cho kết quả của các mô hình dự đoán tốt hơn so với các phương pháp lọc khác.

Từ khóa: *Lựa chọn đặc trưng, phương pháp lọc, phương pháp bao bọc, phương pháp lai, phương pháp nhúng.*

I. INTRODUCTION

Recently, applications have generated massive amounts of data such as video, photos, voice and data obtained from social networking and from the cloud computing. Such complex data contains the characteristics of multi-dimensional dimensions, noisy data, redundant or missing attributes that pose challenges for data analysis and decision making. In order to handle this issue, feature selection has been investigated in the data preprocessing phase. Feature selection (FS) is used to select optimal subset that improves the performace of the predictive model. Feature selection minimizes redundant, irrelevant, and interference features. As a results, feature selection may improve efficiency of classifiers and give the most informative features [1]. FS methods are classified into three categories including Filter, Wrapper and Embedded. Filter method is divided into Filter Feature Ranking and Filter Feature Subset Selection [2]. Filter Feature Ranking methods assess and rank attributes in datasets which are independent with learning algorithms that requires less computation time. Some of the statistical measurement methods used in the

Filter include Information gain, Chi-square, Fisher score, Gain Ratio, and Relief. Wrapper uses machine learning techniques to evaluate a subset of features against their respective criteria. Wrapper's performance is dependent on the sorting algorithm. The best subset of the features are selected based on the results of the classification algorithm. Computationally, Wrapper methods require more complex computation than the Filters, due to the iterative learning steps and cross validation. However, these methods are more accurate than Filter. Some algorithms used in Wrapper are Recursive feature elimination [3], Sequential feature selection algorithm [4] and Genetic algorithm. The third approach is the Embedded method which uses combined learning methods and hybrid methods to select optimal features. Feature selection is a crucial data preprocessing step as it solves the high dimensional data [5], improves the quality of data, redudes the computational time and increases the predictive performance of models. In this study, we examine a comparative performance analysis of five filter-based feature methods namely Chi-Square, Fisher Score, Information Gain, Gain Ratio and Relief based on five different classifiers. Five classifiers including K-nearest Neighbors, Decision Tree, Random Forest, Naive Bayes, and Multilayer Perceptron. From our experimental results, Chi-Square and Information Gain recorded the best performance of filter methods across datasets and models. The rest of this paper is structured as follows. Section 2 presents a literature review of feature selection methods. Experimental design and results of some filter-based feature selection methods are presented in section 3 and section 4. Section 5 concludes the comparative analysis and discusses future work.

II. BACKGROUND

Studies have shown that feature selection techniques can improve predictive efficiency and accuracy for machine learning techniques. The feature selection technique plays an important role in minimizing computational complexity, capacity and cost [6].

1. Feature Selection Process

The process of selecting features in a data set consisting of 4 stages: selecting search techniques, determining search strategy, determining evaluation criterion and reaching stopping criteria.

a) Selecting search techniques

Ang et al. [7] state that the first stage in the feature selection process is to find subsets search techniques. Search techniques are classified into forward, backward and random search. The search process starts with an empty set so that new properties are added in each loop called forward search. In contrast to forward search, backward search starts with a data set with full properties and properties are discarded until an optimal subset is reached. Another approach is random search which constructs a subset of properties by adding and removing properties at each loop. After selecting the trait search techniques, the search strategy will be applied at stage 2.

Table I
LIST OF FEATURE SELECTION METHODS

	Feature Selection	Search Method
Filter-based Feature Ranking Methods	Information Gain	Ranker Search
	Attribute Evaluator(IG)	Ranker Search
	Gain Ratio Attribure	Ranker Search
	Evaluator(GR)	Ranker Search
	Chi-Square	Ranker Search
	Fisher Score	Ranker Search
	Relief	Ranker Search
	Correlation	Correlation
Filter-based Subset Selection	Correlation-based Feature Subset Selection (CFS)	Best First Search (BFS)
		Greedy
		Greedy Stepwise Search (GSS)
		Ant Search (AS)
		Bat Search (BAT)
		Firefly Search (FS)
		Genetic Search (GS)
		PSO
		Search (PSOS) Best
	Consistency Feature Subset Selection (CNS)	Best First Search (BFS) Greedy
		Greedy Stepwise Search (GSS)
		Ant Search (AS)
		Bat Search (BAT)
		Firefly Search(FS)
		Genetic Search (GS)
		PSO Search (PSOS)

b) Determine search strategy

The possible search strategies are randomized, exponential and sequential in the literature. Table I lists the different search strategies and their algorithms [2]. A good search strategy requires optimal solutions, local search capabilities and computational efficiency [8]. Based on these search requests, the algorithms are further classified as the optimal and suboptimal selection of the algorithm.

c) Determine evaluation criterion

The most optimal features are selected based on evaluation criteria. Based on the technical evaluation methods, the features [9] are classified into Filter, Wrapper, Embedded and Hybrid.

d) Reach stopping criteria

The selection criteria specify a process for feature selection that stops when the optimal feature subset is reached. Stopping criteria for feature selection are effective with low complexity computation finding subsets of optimal features

and solving overfitting problems. The choice of the stopping standards is influenced by the preexecution stages. Some of the stopping standards include: predefined quantities of properties, predetermining the number of repetitions, percentage progress between 2 consecutive loops, relying on evaluation functions.

e) Validate the final results

To evaluate the results of the feature selection techniques, some evaluation measures are used such as Crossvalidation, Confusion matrix, Jaccard similarity-based measure and Rand Index. Some of the evaluation metrics for classification techniques classification and clustering include Accuracy, Precision, Recall...

2. Filter-based Feature Selection Techniques

The Filter method relies on the unique feature of the data to evaluate and select a subset of properties, using the evaluation criteria extracted from the data set such as distance, information, dependency, consistency. Specifically, the filter method uses typical criteria of the ranking technique and the ranking order method for the selection of variables. Due to simplicity, high performance and easy to generate optimal features, filter methods are used in this study [10].

a) Chi-square

In statistics, Chi-square [11] is applied to check the independence of two events, where if two events A and B are defined as independent if $P(AB) = P(A).P(B)$, which is equivalent to $P(A|B) = P(A)$ and $P(B|A) = P(B)$. In feature selection, the two main events are features and targets. We use the Chi-square value to find out which feature contains a lot of information for the model. We calculate the Chi-square value between each feature and the target. Characteristics that give high value is a good one. Chi-square is calculated as follows:

$$X^2_{(D,t,c)} = \Sigma_{e_t \in 0,1} \Sigma_{e_c \in 0,1} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \quad (1)$$

where e_t, e_c have 2 values: 0 and 1, N is the observed value in D and E is the expected value.

b) Fisher score

Fisher score is one of the most widely used feature selection algorithm for determining a subset of features. The idea in Fisher score is to select each feature independently according to its score under the Fisher criterion. The Fisher Score of the j^{th} feature is computed below [12]:

$$F(x^j) = \frac{\sum_{k=1}^{c} n_k(\mu_k^j - \mu^j)^2}{(\sigma^i)^2} \quad (2)$$

where $(\sigma^i)^2 = \sum_{k=1}^{c} n_k(\sigma_k^i)^2$
μ_k, n_k are the mean vector and size of the k^{th} class respectively in the reduced data space. After computing the Fisher score for each feature, it selects the top-m ranked features with large scores.

c) Information Gain

Information Gain is an entropy-based feature evaluation method. For example, in text classification problem, it is defined as the amount of information provided by the feature item for text category. Information gain is calculated by how much of a term can be used for classification of information, in order to measure the importance of lexical items for the classification. The formula of the information gain is shown below [13]:

$$G(D,t) = -\sum_{i=1}^{m} P(C_i) log P(C_i) + P(t) \sum_{i=1}^{m} P(C_i|t) log P(C_i|t) + P(\bar{t}) \sum_{i=1}^{m} P(C_i|\bar{t}) log P(C_i|\bar{t}) \quad (3)$$

C is a set of document collection, in whitch there is not the feature t. The value of $G(D,t)$ is greater; t is more useful for the classification for C. This t should be selected. If the greater value of $G(D,t)$ is wanted, it should make the value of $P(t)$ and $P(\bar{t})$ smaller.

d) Gain Ratio

Gain ratio is a modification of the information gain that reduces its bias. Gain ratio takes number and size of branches into account when choosing a feature. It corrects the information gain by taking the intrinsic information of a split into account. Intrinsic information is entropy of distribution of instances into branches. Value of feature decreases as intrinsic information gets larger [14].

$$Gainratio(Feature) = \frac{Gain(Feature)}{Intrinsicinfo(Feature)} \quad (4)$$

e) Relief

The main idea of the Relief algorithm by Kira and Render [15] is similar to the basic rules of k-nearest neighbor algorithm. Being the same class is much more likely for the closer distances to a given distance. If a feature is useful, it is expected that the closest distances of the same class is closer to range given throughout this attribute than the closest distances of all the other classes. The weight of a given feature is calculated below:

$$W = \frac{(W - diff(x_{xj}), nearhit_{ij}^2 + diff(x_{ij}, nearmiss_{ij})^2)}{m} \quad (5)$$

where m is the sample size (randomly selected from a subnet of the training set), $diff(x_{xj}, nearhit_{ij}^2)$ is difference between values of attribute within randomly selected j

Table II
THE DESCRIPTION OF DATASETS

Dataset	Lang	No. of features	No. of Modules	No. of Defective Modules	Defect rate
CM1	C	41	505	48	9.5
MW1	C	38	253	27	10.67

distance and $nearhit_{ij}^2$ value of attribute within the closest training sample in the same class, $diff(x_{ij}, nearmiss_{ij})$ is described as value of the closest training sample from different class. For an useful attribute, x_{ij} and $nearmiss_{ij}$ values are expected to be very close to each other. If an attribute is not useful, both differences are expected to take almost the same distribution.

III. EXPERIMENTAL DESIGN

1. Experimental Setup

From previous studies, many researchers proved that feature selection methods can improve the performance of predictive models. The selection of optimal metrics considerably reduces the computational time, complexity and storage [6]. However, only small number of studies [16, 17] have been carried out to evaluate the effectiveness of different feature selection methods and identify which ones are the most useful. Therefore, we examine the comparative performance analysis of 5 filter-based feature ranking methods namely Chi-Square, Fisher Score, Information Gain, Gain Ratio and Relief based on five different classifiers. The classifiers are K-nearest Neighbors, Decision Tree, Random Forest, Naive Bayes and Multilayer Perceptron. The experiments were conducted on two datasets namely CM1 and MW1 from NASA repository. A brief description of the datasets is shown in Table II.

2. Software Defect Datasets

The experimental results are affected by the quality of datasets, hence choosing the right datasets plays an important role. According to Catal and Diri [18], by using datasets from National Aeronautics Space Administration (NASA) Facility Metrics Data Program (MDP) repository[14], the classification results are more reliable. A brief description of the datasets with the number of features and modules is shown in Table II.

3. Performance Evaluation Metrics

In order to assess and rank features in datasets, filter feature ranking method uses the computational metrics of dataset. After grading each feature based on different characteristics such as statistics, probability or instance, features are generated according to their score [18]. In this study, $log_2 N$ [19] is used to filter top-ranked features, where N is the number of metrics in full defect dataset. Table III illustrates the filter feature ranking methods used in this study.

In order to evaluate the effectiveness of filter feature selection methods, performance metrics include Accuracy, Recall, Precision, AUC (area under ROC curve) and F-measure. Table IV presents a confusion matrix for the performance of a model.

- True Positive (TP): indicates that the positive samples are correctly labeled by the classifier.
- True Negative (TN): indicates that to the negative samples are correctly labeled by the classifier.
- False Positive (FP): indicates that the negative samples are incorrectly labeled as positive.
- False Negative (FN): indicates that the positive samples are mislabeled as negative.
- Accuracy is a ratio of correctly predicted observation to the total observations. In the experiment, the value of accuracy is ranged from 0 to 1.
- Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The high precision shows that the predictive model is more accurate.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

F_1 is the weighted average of Precision and Recall. F_1 assesses whether the increase in precision (recall) is greater than the reduction in recall (precision).

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

ROC curve (the Receiver Operating Curve) and AUC (Area Under the ROC Curve) evaluate the performance of software fault prediction models. The nearer the curve reaches the left border of the ROC space and then reaches the top border of the ROC space, the more correct the test. Otherwise, the curve approaches the diagonal line and AUC value is small; the fault prediction model brings low efficiency.

IV. EXPERIMENTAL RESULTS

This section illustrates the experimental results of comparative performance analysis of five above filter-based feature selection methods based on the accuracy metric. The results were compared on two cases (with and without feature selection). Table V and table VI present the comparisons of filter methods on each of the five classifiers respectively. From the results of table V, on the CM1 datasets, Chi-Square method had the highest accuracy value

Table III
FILTER FEATURE RANKING METHODS

Filter feature ranking methods	Search Method	Measured-based	Reference
Gain Ratio	Ranker Search	Probability-based	[20],[21]
Information Gain	Ranker Search	Statistical based	[20],[21]
Chi-square	Ranker Search	Statistical based	[20],[21]
Relief	Ranker Search	Instance based	[20],[21]
Fisher Score	Ranker Search	Statistical based	[21],[22]

Table IV
CONFUSION MATRIX

	Predicted	
Observed	**Faulty**	**Non-Faulty**
Fault Prone	C	41
Not Fault Prone	C	38

on the predictive performance of Multilayer Perceptron and Naive Bayes with 86.86 and 85.34 respectively. Moreover, Decision Tree and Random Forest based on Gain Ratio reached the best accuracy values of 82.89 abd 85.05. This result showed Chi-Square gave the best performance on CM1 datasets compared to other filter methods.

In table VI, Multilayer Perceptron with Information Gain achieved the highest accuracy of 97.83 on the MW1 dataset. Furthermore, the accuracy performance of all classfiers based on FS methods, in this case, Gain Ratio, Information Gain and Relief were better than when no FS methods are applied. Specifically, Naive Bayes using Gain Ratio and Information Gain had the highest accuracy value of 97.51 in comparison to 95.58 value achieved by the model had no FS methods. Same also was observed for K-nearest Neighbors, Decision Tree and Random Forest, the accuracy values are better than models without FS methods. Hence, the selection of these metrics can increase the accuracy of classifiers compared to models without feature selection. It was observed that Chi-Square and Information Gain had the best influence on the prediction models over other filter methods. This clearly shows the predictive model developed with feature selection methods outperformed other models.

It is clearly observed that based on each method, the best subset of features was selected with a certain threshold value. In table V and table VI, we have shown that the selected features were suggested for each methods. In order to generate the number of features by feature selection methods, the metric values are calculated based on $log_2 N$ where N is the number of metrics in the full defect dataset. In this case, the number of selected features is 6 on both CM1 and MW1 datasets. As presented in table V and table VI, the selected features are listed based on specific threshold values which are calculated by the formula given in section III.3. We observed that the filter methods with the lesser of features achieve better performance compared to models which were trained with all the features. This concludes that reducing number of features improve the performance of the prediction models.

V. CONCLUSION

In software fault prediction, the datasets used for machine learning may contain irrelevant, redundancy features or to be noisy. Hence, feature selection methods for training data plays an important role in selecting the optimal features for predictive models to improve the quality of system and achieve a strong prediction model. Our objective is to empirically evaluate the performance of different filter-based ranking methods using different classifiers over two datasets from NASA repository. The filter feature selection methods are Gain Ratio, Information Gain, Relief, Chi-square and Fisher Score. From the experimental results, Chi-Square and Information Gain are potential methods which had high improvement on the predictive performance of classifiers on the CM1 and MW1 respectively. Additionally, further analysis showed list of selected features for each method based on the threshold value. It was conclusively discovered that the performance of feature selection methods varies from the used datasets and the choice of classification algorithm. In the future work, we will study the ensemble learning in order to propose a hybrid approach which is based on feature selection and ensemble techniques.

REFERENCES

[1] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, vol. 207. Springer, 2006.

[2] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, "Performance analysis of feature selection methods in software defect prediction: a search method approach," *Applied Sciences*, vol. 9, no. 13, p. 2764, 2019.

[3] K. Yan and D. Zhang, ""feature selection and analysis on correlated gas sensor data with recursive feature elimination"," *Sensors and Actuators B: Chemical*, vol. 212, pp. 353–363, 2015.

[4] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[5] A. Akintola, A. Balogun, F. Lafenwa-Balogun, and H. Mojeed, "Comparative analysis of selected heterogeneous classifiers for software defects prediction using filter-based feature selection methods," *FUOYE Journal of Engineering and Technology*, vol. 3, 03 2018.

[6] M. Gutkin, R. Shamir, and G. Dror, "Slimpls: A method for feature selection in gene expression-based disease classification," *PloS one*, vol. 4, p. e6416, 02 2009.

Table V
PERFORMANCE ACCURACY VALUES OF FILTER METHODS ON CM1 DATASETS

Method	**No. of selected features**	**Threshold value**	**Selected Features**	**Classifier**				
				K-nearest Neighbors	**Decision Tree**	**Random Forest**	**Naive Bayes**	**Multilayer Perceptron**
Gain Ratio	6	0.65	halstead_effort, halstead_prog_time, halstead_volume, halstead_content, halstead_difficulty, percent_comments	78.95	**82.89**	**85.05**	84.13	85.50
Information Gain	6	0.029	loc_comments, num_unique_operands, halstead_content,call_pairs, num_unique_operators, loc_executable	76.19	74.77	77.22	85.97	86.42
Relief	6	28	multiple_condition_count, halstead_level, cyclomatic_complexity, number_of_lines, loc_blank, halstead_difficulty	61.60	61.80	69.16	83.23	86.11
Chi-square	6	0.029	loc_comments, num_unique_operands, loc_blank, halstead_content, num_operators, num_unique_operators	61.52	54.38	66.02	**85.34**	**86.86**
Fisher Score	6	31	multiple_condition_count, halstead_level, cyclomatic_complexity, number_of_lines, loc_blank, halstead_difficulty	73.45	81.30	82.60	84.73	86.25
Without FS	All features		All features	79.55	82.05	84.10	83.20	85.34

Table VI
PERFORMANCE ACCURACY VALUES OF FILTER METHODS ON MW1 DATASETS

Method	**No. of selected features**	**Threshold value**	**Selected Features**	**Classifier**				
				K-nearest Neighbors	**Decision Tree**	**Random Forest**	**Naive Bayes**	**Multilayer Perceptron**
Gain Ratio	6	0.52	halstead_effort, halstead_prog_time, halstead_content, halstead_volume, halstead_difficulty, halstead_length	95.02	96.19	97.20	**97.51**	97.79
Information Gain	6	0.054	loc_comments, halstead_volume, edge_count, node_count, halstead_error_est, loc_blank	**95.57**	**96.75**	97.40	**97.51**	**97.83**
Relief	6	31	halstead_difficulty, halstead_error_est, essential_complexity, percent_comments, parameter_count, halstead_content	94.46	95.39	**97.65**	96.12	97.79
Chi-square	6	0.055	loc_comments, halstead_volume, num_unique_operands, edge_count, node_count, design_complexity	93.63	96.40	96.88	96.40	97.80
Fisher Score	6	31	condition_count, essential_complexity, branch_count, num_unique_operands, design_density decision_density	95.45	96.43	97.75	96.54	97.79
Without FS	All features		All features	95.43	96.61	97.58	95.58	97.72

[7] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 5, pp. 971–989, 2015.

[8] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[9] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[10] T. M. P. Hà and T. Q. H. Phan, "Nghiên cứu các kỹ thuật lựa chọn đặc trưng trong tập dữ liệu," *Hội thảo Khoa học quốc gia CITA 2020 lần thứ 9*, pp. 204–210, 2020.

[11] X. Jin, A. Xu, R. Bie, and P. Guo, "Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles," in *International Workshop on Data Mining for Biomedical Applications*, pp. 106–115, Springer, 2006.

[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. USA: Wiley-Interscience, 2000.

[13] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," *Proceedings of the 21st Australasian Computer Science Conference ACSC'98*, pp. 181–191, 1998.

[14] J. Han, M. Kamber, and J. Pei, "3 - data preprocessing," in *Data Mining (Third Edition)* (J. Han, M. Kamber, and J. Pei, eds.), The Morgan Kaufmann Series in Data Management Systems, pp. 83–124, Boston: Morgan Kaufmann, third edition ed., 2012.

[15] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992* (D. Sleeman and P. Edwards, eds.), pp. 249–256, San Francisco (CA): Morgan Kaufmann, 1992.

[16] K. Gao, T. Khoshgoftaar, H. Wang, and N. Seliya, "Choosing software metrics for defect prediction: An investigation on feature selection techniques," *Softw., Pract. Exper.*, vol. 41, pp. 579–606, 04 2011.

[17] K. Muthukumaran, A. Rallapalli, and N. L. B. Murthy, "Impact of feature selection techniques on bug prediction models," in *Proceedings of the 8th India Software Engineering Conference*, ISEC '15, (New York, NY, USA), p. 120–129, Association for Computing Machinery, 2015.

[18] C. Catal and B. Diri, "A systematic review of software fault prediction studies," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7346–7354, 2009.

[19] H. Wang, T. Khoshgoftaar, and A. Napolitano, "A comparative study of ensemble feature selection techniques for software defect prediction," in *2010 Ninth International Conference on Machine Learning and Applications*, pp. 135–140, 12 2010.

[20] S. S. Rathore and A. Gupta, "A comparative study of feature-ranking and feature-subset selection techniques for improved fault prediction," in *Proceedings of the 7th India Software Engineering Conference*, ISEC '14, (New York, NY, USA), Association for Computing Machinery, 2014.

[21] Z. Xu, J. Liu, Z. Yang, G. An, and X. Jia, "The impact of feature selection on defect prediction performance: An empirical comparison," *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 309–320, 2016.

[22] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, (Arlington, Virginia, USA), p. 266–273, AUAI Press, 2011.

Ha Thi Minh Phuong received Bachelor degree in Information Technology from The University of Danang-University of Science and Technology in 2010. She earned M.Sc. degree in Computer Science at Yuan Ze University, Taiwan in 2013. She is a lecturer at The University of Danang - Vietnam-Korea University of Infomation and Communication Technology. She is currently pursuing the Ph.D. degree in Computer Science at the University of Danang. Her current research focuses are on software testing, deep learning.
Email: htmphuong@vku.udn.vn

Le Thi My Hanh is currently a lecturer of the Information Technology Faculty, University of Science and Technology, Danang, Vietnam. She gained M.Sc. degree in 2004 and the Ph.D. degree in Computer Science at the University of Danang in 2016. Her research interests are about software testing and more generally application of heuristic techniques to problems in software engineering.
Email: ltmhanh@dut.udn.vn

Nguyen Thanh Binh graduated in Information Technology from the University of Danang - University of Science and Technology in 1997. He received PhD. degree in Information Technology at Grenoble Institute of Technology, France in 2004. He has been qualified as Associate Professor since 2013. He is currently working at the University of Danang - Vietnam-Korea University of Information and Communication Technology. His research interests include software engineering and software quality.
Email: ntbinh@vku.udn.vn

An Improvement of Least Square - Twin Support Vector Machine

Nguyen The Cuong, Nguyen Thanh Vi
Faculty of Basic, Telecommunications University, Khanh Hoa, Vietnam
Tác giả liên hệ: Nguyen The Cuong, nckcbnckcb@gmail.com
Ngày nhận bài: 26/03/2021, ngày sửa chữa: 01/06/2021, ngày duyệt đăng: 12/06/2021
Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.970

Abstract: In binary classification problems, two classes of data seem to be different from each other. It is expected to be more complicated due to the number of data points of clusters in each class also be different. Traditional algorithms as Support Vector Machine (SVM), Twin Support Vector Machine (TSVM), or Least Square Twin Support Vector Machine (LSTSVM) cannot sufficiently exploit information about the number of data points in each cluster of the data. Which may be effect to the accuracy of classification problems. In this paper, we propose a new Improvement Least Square - Support Vector Machine (called ILS-SVM) for binary classification problems with a class-vs-cluster strategy. Experimental results show that the ILS-SVM training time is faster than that of TSVM, and the ILS-SVM accuracy is better than LSTSVM and TSVM in most cases.

Keywords: *Support vector machine, twin support vector machine, least square twin support vector machine.*

Tên bài: **Một Cải Tiến Của Máy Véc-tơ Tựa Song Sinh Dùng Bình Phương Tối Thiểu**

Tóm tắt: Trong bài toán phân loại nhị phân, hai lớp dữ liệu thường khác biệt nhau. Phức tạp hơn là số lượng các điểm dữ liệu của mỗi cụm trong mỗi lớp cũng khác nhau. Các thuật toán phân loại truyền thống như Máy Véc-tơ Tựa (SVM), Máy Véc-tơ Tựa Song Sinh (TSVM) hay Máy Véc-tơ Tựa Song Sinh Dùng Bình Phương Tối Thiểu (LSTSVM) không khai thác đầy đủ thông tin về số lượng các điểm trong mỗi cụm. Điều này có thể ảnh hưởng đến độ chính xác của bài toán phân loại. Trong bài này, chúng tôi giới thiệu một thuật toán phân loại nhị phân mới, là cải tiến của LSTSVM (được gọi là ILS-SVM) với chiến lược lớp-vs-cụm. Các kết quả thực nghiệm chỉ ra rằng thời gian huấn luyện của ILS-SVM là nhanh hơn của TSVM, độ chính xác của ILS-SVM là tốt hơn LSTSVM và TSVM trong hầu hết các trường hợp.

Từ khóa: *Máy véc-tơ tựa, máy véc-tơ tựa song sinh, máy véc-tơ tựa song sinh dùng bình phương tối thiểu*

I. INTRODUCTION

In the early years of the 20th century, the Support Vector Machine (SVM) [1, 2] was a popular binary classification algorithm applied to many different fields in practice [3–7]. The SVM seeks a hyperplane separating two classes so that the margin between them is largest. Actual data is often established with different clusters, but SVM does not fully exploit information about the number of data points of the clusters.

Nowadays, with rapid development, datasets are increasing in number and diversifying in structure. This fact requires classification algorithms to guarantee accuracy and improve the speed. Many variants of SVM have been recently proposed to improve the speed and other task of standard SVM [6–10]. Some typical innovations of SVM are Twin Support Vector Machine (TSVM) [11], Structural Twin Support Vector Machine (S-TSVM) [12], and Least Square Twin Support Vector Machine (LSTSVM) [13]. The main idea of TSVM is to seek two hyperplanes such that each hyperplane is closer to one class and far away from the other by solving two Quadratic Programming Problems (QPPs) whose size are smaller than the QPP in SVM. Despite having to solve two QPPs, the speed of TSVM is approximately four times faster than standard SVM. S-TSVM has the same strategy as TSVM. Besides, S-TSVM fully exploits structural information with cluster granularity into learning the model to build a more reasonable classifier. LSTSVM has the same strategy as TSVM and S-TSVM. But, LSTSVM attempts to solve two modified primal problems of TSVM, instead of two dual problems usually solved. LSTSVM shows that the solution of the two modified primal problems reduces to solving just two Systems of Linear Equations (SLEs) as opposed to solving two QPPs in TSVM.

Based on the strategy of S-TSVM and LSTSVM, we propose a new binary classification model: Improvement Least Square - Support Vector Machine (called ILS-SVM) with a class-vs-cluster strategy. Instead of solving two SLEs as in LSTSVM, ILS-SVM will solve $(l+k)$ SLEs, where k and l are the number of clusters in class $\{+\}$ and class $\{-\}$, respectively. This method allows ILS-SVM to effectively improve computation speed and classification accuracy.

The paper is organized as follows. Section 2 briefly introduces the background of SVM, TSVM, and LSTSVM; Section 3 is devoted to a detailed description of ILS-SVM along with the algorithms and discussions; All experimental results are presented in Section 4, together with the comparative evaluation; The conclusion is given in Section 5. All algorithms are settled by version 3.8.3 of Python Programming Language.

II. BACKGROUND

In this section, we first briefly describe the background of SVM, TSVM, and LSTSVM.

1. Problem

Consider a binary classification problem with the dataset, denoted by matrix C, consisting of m points (each point is a row of C) $\mathbf{x}_j^T \in \mathbb{R}^n,\ j = 1, \ldots, m$. We also write $\mathbf{x}_j \in C$ to indicate that $\mathbf{x}_j$ is a row of C. Suppose that $y_j \in \{-1, 1\}$ is the $j-$th data point label. Class $\{+\}$ consists of m_A points denoted by a matrix $A \subset \mathbb{R}^{m_A \times n}$, class $\{-\}$ consists of m_B points denoted by a matrix $B \subset \mathbb{R}^{m_B \times n}$. There are k clusters in class $\{+\}$, whose $i-$th cluster consists of m_{Ai} points and is denoted by matrix $A_i \subset \mathbb{R}^{m_{Ai} \times n}$. Also, there are l clusters in class $\{-\}$, whose $j-$th cluster consists of m_{Bj} points and is denoted by matrix $B_j \subset \mathbb{R}^{m_{Bj} \times n}$.

2. Support Vector Machine

Standard SVM [1] seeks a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ separating class $\{+\}$ and class $\{-\}$ such that the margin $\frac{2}{\|\mathbf{w}\|}$ between two classes is largest. However, Standard SVM is only available when the data is linearly separable. In the case when the data is not linearly separable, Soft SVM [1] is recommended with the more loser constraints:

$$\begin{cases} \min\limits_{\mathbf{w},\ b,\ \boldsymbol{\xi}} & c\mathbf{e}^T\boldsymbol{\xi} + \frac{1}{2}\|\mathbf{w}\|_2^2, \\ \text{s.t.} & D(C\mathbf{w} + \mathbf{e}b) + \boldsymbol{\xi} \geq \mathbf{e},\ \boldsymbol{\xi} \geq \mathbf{0}, \end{cases} \tag{1}$$

where $D \in \mathbb{R}^{m \times m}$ is the diagonal matrix with $D_{jj} = y_j;\ \forall j$, $\boldsymbol{\xi} \in \mathbb{R}^m$ is the vector of slack variables, $c \in \mathbb{R}$ is the penalty coefficient, appropriately selected to adjust the role between terms in the objective function, $\mathbf{e} \in \mathbb{R}^m$ is the vector of ones. A new data point $\mathbf{x}$ will be classified in class $\{+\}$ if $\text{sgn}(f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+b) > 0$ and in class $\{-\}$ if $\text{sgn}(f(\mathbf{x})) < 0$, (see Figure 1.a).

3. Twin Support Vector Machine

Based on the strategy of Multisurface Proximal Support Vector Classification via Generalized Eigenvalues (GEPSVM) [14], the main idea of TSVM [11] for binary classification problem is to seek two hyperplanes:

- $f_+(\mathbf{x})(= \mathbf{w}_+^T\mathbf{x} + b_+) = 0$ is closer to class $\{+\}$ and far away from class $\{-\}$,
- $f_-(\mathbf{x})(= \mathbf{w}_-^T\mathbf{x} + b_-) = 0$ is closer to class $\{-\}$ and far away from class $\{+\}$

by solving two QPPs as follows:

$$\begin{cases} \min\limits_{\mathbf{w}_+, b_+, \boldsymbol{\xi}} & \frac{1}{2}\|A\mathbf{w}_+ + \mathbf{e}_A b_+\|_2^2 + c_1\mathbf{e}_B^T\boldsymbol{\xi}, \\ \text{s.t.} & -(B\mathbf{w}_+ + \mathbf{e}_B b_+) + \boldsymbol{\xi} \geq \mathbf{e}_B,\ \boldsymbol{\xi} \geq \mathbf{0}, \end{cases} \tag{2}$$

$$\begin{cases} \min\limits_{\mathbf{w}_-, b_-, \boldsymbol{\eta}} & \frac{1}{2}\|B\mathbf{w}_- + \mathbf{e}_B b_-\|_2^2 + c_2\mathbf{e}_A^T\boldsymbol{\eta}, \\ \text{s.t.} & (A\mathbf{w}_+ + \mathbf{e}_A b_+) + \boldsymbol{\eta} \geq \mathbf{e}_A,\ \boldsymbol{\eta} \geq \mathbf{0}. \end{cases} \tag{3}$$

Where c_1, c_2 are penalty coefficients to adjust the role between terms in the objective functions, $\mathbf{e}_A \in \mathbb{R}^{m_A}$, $\mathbf{e}_B \in \mathbb{R}^{m_B}$ are vectors of ones, $\boldsymbol{\xi} \in \mathbb{R}^{m_B}$, $\boldsymbol{\eta} \in \mathbb{R}^{m_A}$ are vectors of slack variables. A new data $\mathbf{x}$ is classified into class $\{+\}$ or class $\{-\}$ depending on whether it is closer to the hyperplane $f_+(\mathbf{x}) = 0$ or the hyperplane $f_-(\mathbf{x}) = 0$, (see Figure 1.b). It has been shown in [11] that the training time of TSVM is approximately four times faster than that of SVM.

4. Least Squares Twin Support Vector Machine

LSTSVM [13] modifies the primal problems (2) and (3) of Linear TSVM in a least squares sense, with the inequality constraints replaced by equality constraints as follows:

$$\begin{cases} \min\limits_{\mathbf{w}_+, b_+} & \frac{1}{2}\|A\mathbf{w}_+ + \mathbf{e}_A b_+\|_2^2 + \frac{1}{2}c_1\boldsymbol{\xi}^T\boldsymbol{\xi}, \\ \text{s.t.} & -(B\mathbf{w}_+ + \mathbf{e}_B b_+) + \boldsymbol{\xi} = \mathbf{e}_B, \end{cases} \tag{4}$$

$$\begin{cases} \min\limits_{\mathbf{w}_-, b_-} & \frac{1}{2}\|B\mathbf{w}_- + \mathbf{e}_B b_-\|_2^2 + \frac{1}{2}c_2\boldsymbol{\eta}^T\boldsymbol{\eta}, \\ \text{s.t.} & (A\mathbf{w}_+ + \mathbf{e}_A b_+) + \boldsymbol{\eta} = \mathbf{e}_A. \end{cases} \tag{5}$$

Note also that the QPPs (4) and (5) use the square of $2-$norm of vectors of slack variables $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ with weights $\frac{c_1}{2}$ and $\frac{c_2}{2}$ instead of $1-$norm with weight c_1 and c_2 as used in (2) and (3), which makes the constraints $\boldsymbol{\xi} \geq \mathbf{0}$ and $\boldsymbol{\eta} \geq \mathbf{0}$ redundant [13]. A new data point is assigned into class $\{+\}$ or $\{-\}$ in the same manner as in TSVM, (see Figure 1.c). It has been shown in [13] that the training time of LSTSVM is faster than that of TSVM (see TABLE 1).

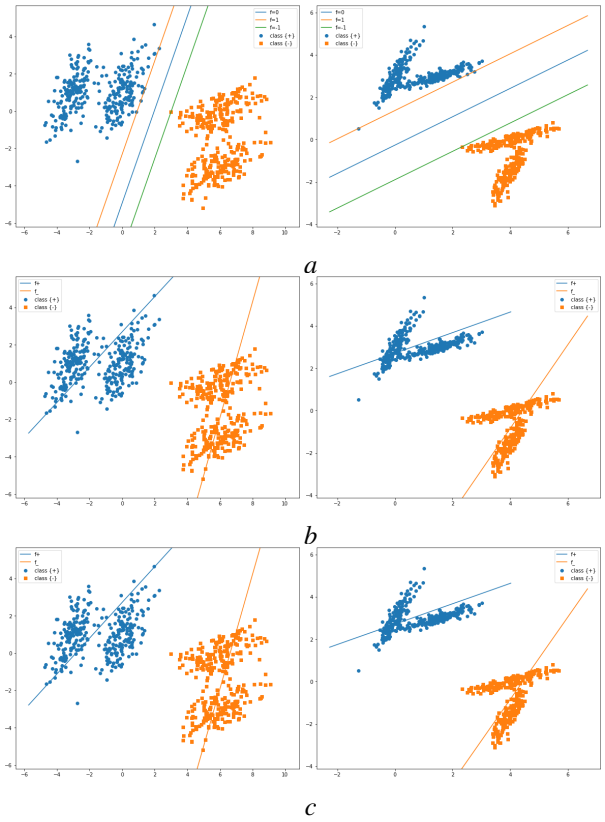


Figure 1. The SVM (a), TSVM (b), and LSTSVM (c) cannot fully exploit information about the number of data-points of clusters in each class.

III. IMPROVEMENT LEAST SQUARE - SUPPORT VECTOR MACHINE

In this section, we describe a new classification algorithm: Improvement Least Square - Support Vector Machine (called ILS-SVM).

Similar to the S-TSVM [12], ILS-SVM also has two steps. The first step is grouping in each class by Ward's linkage clustering method; the second step is the least square model learning. Suppose that, there are k clusters in class $\{+\}$, each cluster consists of m_{Ai} points and is represented by matrix $A_i \subset \mathbb{R}^{m_{Ai}\times n}$, there are l clusters in class $\{-\}$, each cluster consists of m_{Bi} points and is represented by matrix $B_i \subset \mathbb{R}^{m_{Bi}\times n}$. ILS-SVM uses a class-vs-cluster strategy to determine $(l+k)$ hyperplanes such that each of which is closer to one class and far away from cluster of other class. Specifically, the method need to find l hyperplanes such that the $j-$th hyperplane, $f_{j+}(\mathbf{x}) = \mathbf{w}_{j+}^T\mathbf{x} + b_{j+} = 0$, is closer to class A and far away from cluster B_j; Also, It need to find k hyperplanes such that the $i-$th hyperplane, $f_{i-}(\mathbf{x}) = \mathbf{w}_{i-}^T\mathbf{x} + b_{i-} = 0$, is closer to class B and far away from cluster A_i (see Figure 2); Here $\mathbf{w}_{j+}, \mathbf{w}_{i-} \in \mathbb{R}^n, b_{j+}, b_{i-} \in \mathbb{R}$.

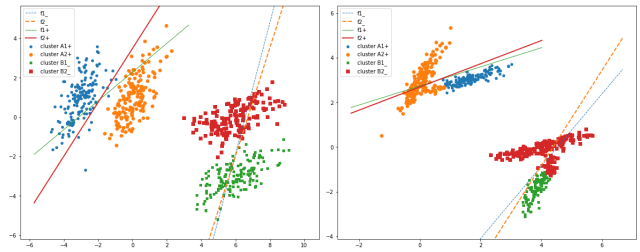


Figure 2. ILS-SVM exploit information about the number of data-points of clusters.

The classifier is now selected as:

$$f(\mathbf{x}) = \underset{+,\ -}{\operatorname{argmin}}(f_+(\mathbf{x}),\ f_-(\mathbf{x})), \tag{6}$$

with

$$f_+(\mathbf{x}) = \sum_{j=1}^{l} \frac{m_{Bj}}{m_B} f_{j+}(\mathbf{x});\ f_-(\mathbf{x}) = \sum_{i=1}^{k} \frac{m_{Ai}}{m_A} f_{i-}(\mathbf{x}). \tag{7}$$

From the definition, we see that $f_+(\mathbf{x})$ is the average, taking into account the weights, distances from $\mathbf{x}$ to the hyperplanes $\{f_{j+}(\mathbf{x}) = 0\}$. The $j-$th hyperplane's weight is proportional to m_{Bj} - the number of data points in the cluster Bj. Similarly, $f_-(\mathbf{x})$ is the weighted average of distances from $\mathbf{x}$ to the hyperplanes $\{f_{i-}(\mathbf{x}) = 0\}$. By virtue of (6), a new data point $\mathbf{x}$ is classified into class $\{+\}$ or $\{-\}$ depending on whether $f_+(\mathbf{x})$ is less than or greater than $f_-(\mathbf{x})$.

1. The linear case

We determine $(l+k)$ hyperplanes in ILS-SVM by solving $(l+k)$ QPPs as follows:

$$\begin{cases} \min\limits_{\mathbf{w}_{j+},b_{j+},\boldsymbol{\xi}_j} & \frac{1}{2}||A\mathbf{w}_{j+}+\mathbf{e}_A b_{j+}||_2^2+\frac{c_1}{2}\boldsymbol{\xi}_j^T\boldsymbol{\xi}_j+\frac{c_2}{2}(\|\mathbf{w}_{j+}\|_2^2+b_{j+}^2), \\ \text{s.t.} & (B_j\mathbf{w}_{j+}+\mathbf{e}_{Bj}b_{j+})+\boldsymbol{\xi}_j = \mathbf{e}_{Bj}, \end{cases} \tag{8}$$

$j = 1, \ldots,\ l$ and

$$\begin{cases} \min\limits_{\mathbf{w}_{i-},b_{i-},\boldsymbol{\eta}_i} & \frac{1}{2}||B\mathbf{w}_{i-}+\mathbf{e}_B b_{i-}||_2^2+\frac{c_3}{2}\boldsymbol{\eta}_i^T\boldsymbol{\eta}_i+\frac{c_4}{2}(\|\mathbf{w}_{i-}\|_2^2+b_{i-}^2), \\ \text{s.t.} & (A_i\mathbf{w}_{i-}+\mathbf{e}_{Ai}b_{i-})+\boldsymbol{\eta}_i = \mathbf{e}_{Ai}, \end{cases} \tag{9}$$

$i = 1, \ldots,\ k$.

Here, $\mathbf{e}_{Ai} \in \mathbb{R}^{m_{Ai}\times 1}$, $\mathbf{e}_{Bj} \in \mathbb{R}^{m_{Bj}\times 1}$, $\mathbf{e}_A \in \mathbb{R}^{m_A\times 1}$, $\mathbf{e}_B \in \mathbb{R}^{m_B\times 1}$ are vectors of ones. $\boldsymbol{\eta}_i \in \mathbb{R}^{m_{Ai}\times 1}$, $\boldsymbol{\xi}_j \in \mathbb{R}^{m_{Bj}\times 1}$ are vectors of slack variables. c_1, c_2, c_3, c_4 are penalty coefficients to adjust the role between terms in the objective functions. In the problem (8), $||A\mathbf{w}_{j+} + \mathbf{e}_A b_{j+}||_2^2$ is the sum of squares of distances from data points in class A to the hyperplane $f_{j+}(\mathbf{x}) = 0$, the inequality constraints are replaced by equality constraints, and is defined by the points of cluster B_j, making training times faster than that of TSVM. $\frac{1}{2}c_1\boldsymbol{\xi}_j^T\boldsymbol{\xi}_j$ is the sum of squares of errors,

$\frac{1}{2}c_2(\|\mathbf{w}_{j+}\|_2^2 + b_{j+}^2)$ is the regularization term. The problem (9) is similarly established for class $\{-\}$ with the constraints is defined by cluster A_i.

By substituting the equality constraints into the objective function, QPP (8) becomes:

$$\min_{\mathbf{w}_{j+},b_{j+}} \frac{1}{2}\|A\mathbf{w}_{j+} + \mathbf{e}_A b_{j+}\|_2^2 + \frac{c_1}{2}\|B_j\mathbf{w}_{j+} + \mathbf{e}_{Bj}b_{j+} - \mathbf{e}_{Bj}\|_2^2 + \frac{c_2}{2}(\|\mathbf{w}_{j+}\|_2^2 + b_{j+}^2). \tag{10}$$

Setting the gradient of (10) with respect to $\mathbf{w}_{j+}$ and b_{j+} to zero, we have:

$$A^T(A\mathbf{w}_{j+}+\mathbf{e}_A b_{j+})+c_1B_j^T(B_j\mathbf{w}_{j+}+\mathbf{e}_{Bj}b_{j+}-\mathbf{e}_{Bj})+c_2\mathbf{w}_{j+} = \mathbf{0},$$

$$\mathbf{e}_A^T(A\mathbf{w}_{j+}+\mathbf{e}_A b_{j+})+c_1\mathbf{e}_{Bj}^T(B_j\mathbf{w}_{j+}+\mathbf{e}_{Bj}b_{j+}-\mathbf{e}_{Bj})+c_2 b_{j+} = 0.$$

Defining $H = \begin{bmatrix}A & \mathbf{e}_A\end{bmatrix}$, $G_j = \begin{bmatrix}B_j & \mathbf{e}_{Bj}\end{bmatrix}$, $\mathbf{u}_j = \begin{bmatrix}\mathbf{w}_{j+}\\ b_{j+}\end{bmatrix}$, $j = 1,\ldots,l$, and I being the identity matrix of order $(n+1)$, it follows that

$$H^THu_j + c_1G_j^TG_j\mathbf{u}_j - c_1G_j^T\mathbf{e}_{Bj} + c_2I\mathbf{u}_j = \mathbf{0} \tag{11}$$

$$\Rightarrow \left[\tfrac{1}{c_1}H^TH + G_j^TG_j + \tfrac{c_2}{c_1}I\right]\mathbf{u}_j = G_j^T\mathbf{e}_{Bj} \tag{12}$$

$$\Rightarrow \mathbf{u}_j = \left[\tfrac{1}{c_1}H^TH + G_j^TG_j + \tfrac{c_2}{c_1}I\right]^{-1}G_j^T\mathbf{e}_{Bj}. \tag{13}$$

In exactly similar way, by defining $G = \begin{bmatrix}B & \mathbf{e}_B\end{bmatrix}$, $H_i = \begin{bmatrix}A_i & \mathbf{e}_{Ai}\end{bmatrix}$, $\mathbf{v}_i = \begin{bmatrix}\mathbf{w}_{i-}\\ b_{i-}\end{bmatrix}$, $i = 1,\ldots,k$, we obtain the solutions of problem (9):

$$\mathbf{v}_i = \left[\tfrac{1}{c_3}G^TG + H_i^TH_i + \tfrac{c_4}{c_3}I\right]^{-1}H_i^T\mathbf{e}_{Ai}. \tag{14}$$

Algorithm 1: Linear ILS-SVM

1 Consider the problem 2.1. We generate the linear classifier $f(\mathbf{x})$ as follows:
2 Clustering dataset by using Ward's linkage [12].
3 Define $H_i = \begin{bmatrix}A_i & \mathbf{e}_{Ai}\end{bmatrix}$, $G = \begin{bmatrix}B & \mathbf{e}_B\end{bmatrix}$, $G_j = \begin{bmatrix}B_j & \mathbf{e}_{Bj}\end{bmatrix}$, $H = \begin{bmatrix}A & \mathbf{e}_A\end{bmatrix}$.
4 Determining $(\mathbf{w}_{j+}, b_{j+})$ and $(\mathbf{w}_{i-}, b_{i-})$ via (13), (14).
5 Classifying a new data $\mathbf{x}$ by using (6) and (7).

2. The nonlinear case

Let $\Phi : \mathbb{R}^n \to \mathbb{H}$ be a nonlinear mapping, where $\mathbb{H}$ is a Hilbert space whose dimension is not less than n (maybe infinite-dimensional). Since $\mathbb{S} = \text{span}(\Phi(C^T))$ is a subspace of $\mathbb{H}$ whose dimension does not exceed m, we can consider $\mathbb{S}$ as an Euclidean space and $\Phi : \mathbb{R}^n \to \mathbb{S}$. Suppose that after the clustering step on space $\mathbb{S}$ we obtain k clusters $\Phi(A_1),\ldots,\Phi(A_k)$ in the class $\Phi(A)$, each cluster $\Phi(A_i)$ consists of m_{Ai} data points; and l clusters $\Phi(B_1),\ldots,\Phi(B_l)$ in the class $\Phi(B)$, each cluster $\Phi(B_j)$ consists of m_{Bj} data points. In space $\mathbb{S}$, a hyperplane $\Phi(\mathbf{x}^T)\mathbf{h} + b = 0$ (with $\mathbf{h} \in \mathbb{S}$ being the normal vector) can be rewritten as $\Phi(\mathbf{x}^T)\Phi(C^T)\mathbf{u} + b = 0$ for some vector $\mathbf{u} \in \mathbb{R}^m$. Therefore, by defining $\Phi(\mathbf{x}^T)\Phi(C^T) = K(\mathbf{x}^T, C^T)$, the hyperplane has the form $K(\mathbf{x}^T, C^T)\mathbf{u} + b = 0$, K is a predefined kernel [15].

ILS-SVM determines l hyperplanes such that the $j-$th one: $K(\mathbf{x}^T,\ C^T)\mathbf{u}_{j+}+b_{j+} = 0$ is closer to class $\Phi(A)$ and far away from cluster $\Phi(B_j)$. It also determines k hyperplanes such that the $i-$th one: $K(\mathbf{x}^T,\ C^T)\mathbf{u}_{i-}+b_{i-} = 0$ is closer to class $\Phi(B)$ and far away from cluster $\Phi(A_i)$. Specifically, we have $(l + k)$ QPPs as follows:

$$\begin{cases}\min\limits_{\mathbf{u}_{j+},b_{j+},\boldsymbol{\xi}_j} \frac{1}{2}\|K(A,C^T)\mathbf{u}_{j+}+\mathbf{e}_A b_{j+}\|_2^2+\frac{c_1}{2}\boldsymbol{\xi}_j^T\boldsymbol{\xi}_j+\frac{c_2}{2}(\|\mathbf{u}_{j+}\|_2^2+b_{j+}^2),\\ \text{s.t.} \quad (K(B_j,C^T)\mathbf{u}_{j+} + \mathbf{e}_{Bj}b_{j+}) + \boldsymbol{\xi}_j = \mathbf{e}_{Bj};\end{cases} \tag{15}$$

$\mathbf{u}_{j+} \in \mathbb{R}^m$, $j = 1,\ldots,l$ and

$$\begin{cases}\min\limits_{\mathbf{u}_{i-},b_{i-},\boldsymbol{\eta}_i} \frac{1}{2}\|K(B,C^T)\mathbf{u}_{i-}+\mathbf{e}_B b_{i-}\|_2^2+\frac{c_3}{2}\boldsymbol{\eta}_i^T\boldsymbol{\eta}_i+\frac{c_4}{2}(\|\mathbf{u}_{i-}\|_2^2+b_{i-}^2),\\ \text{s.t.} \quad (K(A_i,C^T)\mathbf{u}_{i-} + \mathbf{e}_{Ai}b_{i-}) + \boldsymbol{\eta}_i = \mathbf{e}_{Ai};\end{cases} \tag{16}$$

$\mathbf{u}_{i-} \in \mathbb{R}^m$, $i = 1,\ldots,k$.

By substituting the constraints into objective function, these QPPs become:

$$\min_{\mathbf{u}_{j+},b_{j+}} \frac{1}{2}\|K(A,C^T)\mathbf{u}_{j+} + \mathbf{e}_A b_{j+}\|_2^2 + \frac{c_1}{2}\|K(B_j,C^T)\mathbf{u}_{j+} + \mathbf{e}_{Bj}b_{j+} - \mathbf{e}_{Bj}\|_2^2 + \frac{c_2}{2}(\|\mathbf{u}_{j+}\|_2^2 + b_{j+}^2), \tag{17}$$

$j = 1,\ldots,l$ and

$$\min_{\mathbf{u}_{i-},b_{i-}} \frac{1}{2}\|K(B,C^T)\mathbf{u}_{i-} + \mathbf{e}_B b_{i-}\|_2^2 + \frac{c_3}{2}\|K(A_i,C^T)\mathbf{u}_{i-} + \mathbf{e}_{Ai}b_{i-} - \mathbf{e}_{Ai}\|_2^2 + \frac{c_4}{2}(\|\mathbf{u}_{i-}\|_2^2 + b_{i-}^2), \tag{18}$$

$i = 1,\ldots,k$.

The solution of QPPs (17) and (18) can be derived to be:

$$\overline{\mathbf{u}_j} = \left[\tfrac{1}{c_1}H^TH + G_j^TG_j + \tfrac{c_2}{c_1}I\right]^{-1}G_j^T\mathbf{e}_{Bj}, \tag{19}$$

$$\overline{\mathbf{v}_i} = \left[\tfrac{1}{c_3}G^TG + H_i^TH_i + \tfrac{c_4}{c_3}I\right]^{-1}H_i^T\mathbf{e}_{Ai}, \tag{20}$$

where, $H_i = \begin{bmatrix}K(A_i,C^T) & \mathbf{e}_{Ai}\end{bmatrix}$, $G = \begin{bmatrix}K(B,C^T) & \mathbf{e}_B\end{bmatrix}$, $G_j = \begin{bmatrix}K(B_j,C^T) & \mathbf{e}_{Bj}\end{bmatrix}$, $H = \begin{bmatrix}K(A,C^T) & \mathbf{e}_A\end{bmatrix}$, $\overline{\mathbf{u}_j} = \begin{bmatrix}\mathbf{u}_{j+}\\ b_{j+}\end{bmatrix}$, $j = \overline{1,l}$, $\overline{\mathbf{v}_i} = \begin{bmatrix}\mathbf{u}_{i-}\\ b_{i-}\end{bmatrix}$, $i = \overline{1,k}$, and I is the identity matrix of order $(m + 1)$.

The classification function now is selected as:

$$f(\mathbf{x}) = \underset{+,\ -}{\text{argmin}}(f_+(\mathbf{x}),\ f_-(\mathbf{x})), \tag{21}$$

with

$$\begin{cases} f_+(\mathbf{x}) = \sum_{j=1}^{l} \frac{m_{Bj}}{m_B}(K(\mathbf{x}^T, C^T)\mathbf{u}_{j+} + b_{j+}); \\ f_-(\mathbf{x}) = \sum_{i=1}^{k} \frac{m_{Ai}}{m_A}(K(\mathbf{x}^T, C^T)\mathbf{u}_{i-} + b_{i-}). \end{cases} \quad (22)$$

Remark: In (19), (20) we need to compute the inverse of square matrices in order $(m + 1)$. This work will become difficult when m is large. So it is necessary to reduce the size of those matrices. This problem can be solved by using the Sherman-Morrison-Woodbury (SMW) formula [16].

Algorithm 2: Nonlinear ILS-SVM

1 Consider the problem 2.1. We generate the nonlinear classifier $f(\mathbf{x})$ as follows:
2 Choosing a kernel function $K(\mathbf{x}^T, C^T)$, typically the Gaussian kernel [15].
3 Clustering the dataset by using Ward's linkage [12].
4 Define $H_i = \left[K(A_i, C^T) \quad \mathbf{e}_{Ai}\right]$, $G = \left[K(B, C^T) \quad \mathbf{e}_B\right]$, $G_j = \left[K(B_j, C^T) \quad \mathbf{e}_{Bj}\right]$, $H = \left[K(A, C^T) \quad \mathbf{e}_A\right]$.
5 Determining $(\mathbf{u}_{1+}, b_{1+}), \ldots, (\mathbf{u}_{l+}, b_{l+})$ and $(\mathbf{u}_{1-}, b_{1-}), \ldots, (\mathbf{u}_{k-}, b_{k-})$ via (19) and (20).
6 Classifying a new data $\mathbf{x}$ by using (21) and (22).

IV. EXPERIMENTS

In this section, we compare the ILS-SVM against LSTSVM [13] and TSVM [11] on various datasets. All algorithms are settled by version 3.8.3 of Python programming language, and run on a Laptop with an AMD Ryzen 5 with 8GB RAM. We use the following libraries: "scipy.cluster.hierarchy" to cluster the data, "cvxopt" to solve the QPP, "matplotlib" to show the figures, "pandas" and "numpy" to process data, 'sklearn' to evaluate and adjust hyperparameters of all models. All settings are uploaded to [17]. We implement these algorithms on 14 UCI datasets [18] which have been experimented in [11], [13]. For each dataset, we randomly select 90% of extracted dataset for training and 10% for testing, and use 10-fold cross validation (CV) to evaluate the accuracy of all algorithms. The results are shown in TABLE 1 (by applying Algorithm 1), and TABLE 2 (by applying Algorithm 2).

TABLE 1 shows the training time and the classification accuracy comparison between ILS-SVM with LSTSVM and TSVM for the linear case on 14 UCI datasets. All hyperparameters such as c_1, c_2, c_3, c_4 of ILS-SVM, c_1, c_2 of LSTSVM and TSVM are belong to the set $\{0.0001, 0.001, 0.1, 1, 10, 100, 1000\}$ and obtained by using grid-search technique. From TABLE 1 we can see that the training time of ILS-SVM is faster than that of TSVM, but slower than LSTSVM. This is because ILS-SVM has to cluster the data and process the problem on each cluster. That is the cost to obtain a more precise classification. Thereby we can see that the classification accuracy of ILS-SVM is better than that of LSTSVM and TSVM in most datasets. The TABLE 2 shows the comparison of training time and classification accuracy for the nonlinear version of ILS-SVM, LSTSVM and TSVM on 4 UCI datasets. The kernel parameters of the Gaussian kernel are obtained through grid-search from the same range, while all penalty parameters of all algorithms are fixed to be one.

Table I
TEST TRAINING TIME (MS) AND CV ACCURACY (%) WITH A LINEAR KERNEL (ALG. 1)

Datasets($m \times n$) ($k \times l$)	Algorithms		
	ILS-SVM	LSTSVM	TSVM
Hepatitis(155×19) (5×3)	3.001 89.9 +/- 6.6	1.001 86.4 +/- 11.7	8.002 84.2 +/- 8.9
Australian(690×14) (4×5)	6.001 87.3 +/- 4.6	1.000 86.3 +/- 3.6	57.014 86.8 +/- 3.4
BUPA-liver(345×6) (2×3)	3.017 67.4 +/- 6.7	1.000 66.1 +/- 7.8	14.003 65.8 +/- 6.0
CMC(844×9) (3×5)	8.989 66.5 +/- 5.8	1.000 64.6 +/- 5.1	143.032 64.8 +/- 5.2
Credit(690×19) (2×3)	12.001 86.5 +/- 2.0	2.000 86.3 +/- 2.7	103.022 86.1 +/- 3.3
Diabetes(768×8) (5×2)	4.981 74.9 +/- 5.0	0.001 74.3 +/- 4.6	24.006 74.7 +/- 5.7
Flare-sol(1066×10) (2×2)	19.985 82.1 +/- 4.5	0.999 81.9 +/- 3.6	244.055 81.8 +/- 4.3
German(1000×20) (3×2)	16.986 71.7 +/- 6.2	1.000 70.8 +/- 5.5	221.048 71.0 +/- 7.2
Heart-stat(270×13) (2×2)	2.000 84.0 +/- 5.5	0.001 84.8 +/- 3.8	10.001 84.0 +/- 3.9
Image(2310×18) (3×2)	6.999 90.2 +/- 1.0	1.000 90.2 +/- 3.0	51.011 91.6 +/- 2.2
Ionosphere(350×34) (3×2)	5.005 89.2 +/- 7.5	0.999 89.6 +/- 6.2	15.002 88.0 +/- 6.9
Spect(265×22) (3×3)	3.987 86.1 +/- 4.7	0.001 83.5 +/- 6.7	9.995 83.1 +/- 6.1
Sonar(208×60) (6×3)	3.993 81.3 +/- 10.3	1.000 74.3 +/- 9.5	9.002 73.8 +/- 9.6
Heart-c(303×13) (2×4)	2.000 85.3 +/- 7.1	0.001 83.8 +/- 6.3	12.003 83.4 +/- 7.7

Table II
TEST TRAINING TIME (S) AND CV ACCURACY (%) WITH AN RBF KERNEL (ALG. 2)

Dataset($m \times n$) ($k \times l$)	Algorithms		
	ILS-SVM	LSTSVM	TSVM
Hepatitis(155×19) (5×3)	0.271 87.0 +/- 9.1	0.119 84.9 +/- 9.3	0.125 83.5 +/- 12.0
heart-c(303×13) (2×4)	0.931 83.1 +/- 7.3	0.460 81.6 +/- 6.0	0.472 82.7 +/- 7.1
Australian(690×14) (4×5)	4.911 87.6 +/- 4.2	2.377 81.8 +/- 3.2	2.446 85.6 +/- 3.8
WPBC(198×35) (2×3)	0.374 81.7 +/- 9.3	0.183 78.1 +/- 6.3	0.194 80.6 +/- 10.2

V. CONCLUSION

This paper has proposed a new Improvement Least Square - Support Vector Machine for classification problems with a class-vs-cluster strategy. This algorithm is performed in two steps: The first step is grouping in each class by Ward's linkage clustering method; The second step is the least square learning model. The classifier is based on the weighted average distances from the data point to the class representative hyperplanes. ILS-SVM has a faster execution time than TSVM, and slower than LSTSVM. But in terms of classification accuracy, ILS-SVM is better than TSVM, and LSTSVM in most cases. For problems with big data and each class contains many clusters, the ILS-SVM algorithm is more effective in data classification. The ILS-SVM algorithm may not really be suitable for multi-classes problems. However, it seems useful in solving the classification problem with imbalanced data. And this is an interesting research direction in the future.

REFERENCES

[1] V. Vapnik, *The Natural Of Statistical Learning Theory*. Springer-Verlag New York, 1995.

[2] G. Fung and O. L. Mangasarian, "Proximal support vector machine," in *KDD '01: Proceedings of the Seventh ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. San Francisco California: Association for Computing Machinery, New York, NY, United States, 2001, pp. 77–86. [Online]. Available: https://dl.acm.org/doi/10.1145/502512.502527

[3] W. Noble, *Support Vector Machine Applications in Computational Biology*. MIT Press, 2004.

[4] M. Adancon and M. Cheriet, "Model selection for the ls-svm. application to handwriting recognition," *Pattern Recognition*, vol. 42, pp. 3264–3270, 2009.

[5] Y. Tian, Y. Shi, and Y. Liu, "Recent advances on support vector machines research," *Technological and Economic Development of Economy*, vol. 18, pp. 5–33, 2012.

[6] D. Tomar and S. Agarwal, "Twin support vector machine: A review from 2007 to 2014," *Egyptian Informatics Journal*, vol. 16, pp. 55–69, 2015.

[7] J. Cervantes, F. Lamont, L. Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[8] X. Pan, Y. Luo, and Y. Xu, "K-nearest neighbor based structural twin support vector machine," *Knowledge-Based Systems*, vol. 88, pp. 34–44, 2015.

[9] X. Xie and S. Sun, "Multitask centroid twin support vector machines," *Neurocomputing*, vol. 149, pp. 1085–1091, 2015.

[10] B. Mei and Y. Xu, "Multi-task least squares twin support vector machine for classification," *Neurocomputing*, vol. 338, pp. 26–33, 2019.

[11] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine intelligence*, vol. 29, pp. 905–910, 2007.

[12] Z. Qi, Y. Tian, and Y. Shi, "Structural twin support vector machine for classification," *Knowledge-Based Systems*, vol. 43, pp. 74–81, 2013.

[13] M. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Systems with Applications*, vol. 36, pp. 7535–7543, 2009.

[14] O. Mangasarian and E. Wild, "Multisurface proximal support vector classification via generalized eigenvalues," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, pp. 69–74, 2006.

[15] B. Schoelkopf and A. Smola, *Learning with Kernel*. MIT Press, 2002.

[16] G. Golub and C. Van Loan, *Matrix Computations*. The John Hopkins University Press, 2013.

[17] N. Cuong, *Python code*. [Online]. Available: https://github.com/makeho8/python/

[18] *UCI Machine Learning Repository*, Center for Machine Learning and Intelligent Systems at the University of California, Irvine. [Online]. Available: http://archive.ics.uci.edu/ml/machine-learning-databases/

Cuong Nguyen The received his B.Sc. degree in Mathematical Education from Hanoi National University of Education, Vietnam in 2009. M.Sc. degree in Mathematical analysis from Hue University College of Sciences, Vietnam in 2014. He is currently working toward his Ph.D. degree in Computer Science at Hue University College of Sciences. He works at Telecommunications University.
Email: nckcbnckcb@gmail.com.

Vi Nguyen Thanh received her B.Sc. degree in Mathematical Education from Da Nang University, Vietnam in 2011. M.Sc. degree in Mathematical analysis from Hue University College of Sciences, Vietnam in 2014. She works at Telecommunications University.
Email: thanhvy.rose@gmail.com

Tìm kiếm ảnh dựa vào Ontology

Nguyễn Thị Uyên Nhi[1,3], Văn Thế Thành[2], Lê Mạnh Thạnh[3]
[1] Khoa Thống kê – Tin học, Trường ĐH Kinh tế – ĐH Đà Nẵng
[2] Phòng quản lý khoa học và đào tạo sau Đại học, Trường ĐH Công nghiệp Thực phẩm TP.HCM
[3] Trường Đại học Khoa học, Đại học Huế

Tác giả liên hệ: Nguyễn Thị Uyên Nhi, uyennhi203@gmail.com
Ngày nhận bài: 15/04/2021, ngày sửa chữa: 01/06/2021, ngày duyệt đăng: 12/06/2021
Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.965

Tóm tắt: Bài toán tìm kiếm ảnh đóng vai trò quan trọng trong mọi lĩnh vực của cuộc sống. Trong bài báo này, một phương pháp tìm kiếm ảnh dựa vào Ontology OBIR (Ontology-based Image Retrieval) được đề xuất nhằm xác định ngữ nghĩa cấp cao của hình ảnh. Ontology bán tự động được đề xuất xây dựng làm cơ sở tri thức cho tìm kiếm ảnh theo tiếp cận ngữ nghĩa, đồng thời câu truy vấn SPARQL được tự động tạo ra từ văn bản đầu vào hoặc từ lớp ngữ nghĩa của ảnh đầu vào, được xác định thông qua công cụ tìm kiếm học máy dựa trên đồ thị cụm láng giềng, để tìm kiếm trên Ontology này. Để minh chứng cho các lý thuyết đã đề xuất, chúng tôi tiến hành thực nghiệm trên bộ ảnh ImageCLEF (20.000 hình ảnh) và Stanford Dogs (20.580 hình ảnh). Kết quả thực nghiệm của phương pháp tìm kiếm ảnh dựa vào Ontology được so sánh với phương pháp tìm kiếm ảnh theo nội dung, đồng thời được so sánh với các công trình khác cùng tập dữ liệu ảnh nhằm chứng minh tính hiệu quả và đúng đắn của các đề xuất trong bài báo.

Từ khóa: *OBIR, Ontology, SPARQL, semantic image.*

Title: **Ontology-based Image Retrieval**

Abstract: Image retrieval problem plays an important role in all areas of life. In this pa-per, Ontology-based Image Retrieval (OBIR) is proposed to determine the high-level semantics of the image. The semi-automatic frame ontology is proposed to be built as a knowledge base for SBIR, at the same time the SPARQL query is automatically generated from the input text or the semantic class of the input image, that identified through the machine learning search engine based on the neighborhood cluster graph, to retrieval on this ontology. Experiments were performed on ImageCLEF and Stanford Dogs to demonstrate the effectiveness of the proposals. Experi-mental results of OBIR are compared with CBIR on the neighborhood cluster graph, and compared with other works on the same image datasets, to prove the effectiveness and correctness of the proposals in the paper.

Keywords: *OBIR, Ontology, SPARQL, smantic image.*

I. GIỚI THIỆU

Ảnh số đóng một vai trò quan trọng trong mọi lĩnh vực của cuộc sống như y khoa, giáo dục, giải trí,... Tìm kiếm ảnh dựa trên nội dung CBIR (Content-based Image Retrieval) [4, 10] là phương pháp phổ biến nhất hiện tại, sử dụng các đặc trưng cấp thấp như màu sắc, kết cấu, hình dạng... và các độ đo khoảng cách để tìm tập ảnh tương tự. Tuy nhiên, luôn tồn tại một "khoảng cách ngữ nghĩa" (semantic gap) [2, 20] giữa các đặc trưng cấp thấp được máy tính trích xuất và ngữ nghĩa cấp cao được truyền đạt của người dùng. Tìm kiếm ảnh theo ngữ nghĩa SBIR (Semantic-based Image Retrieval) [15, 17] là phương pháp xác định ngữ nghĩa cấp cao của hình ảnh, nhằm giải quyết thách thức này.

Có nhiều phương pháp tìm kiếm ảnh theo ngữ nghĩa, phổ biến nhất là dựa vào phương pháp học máy để liên kết đặc trưng cấp thấp với các nhãn lớp ngữ nghĩa, dựa vào Ontology để xác định ngữ nghĩa cấp cao. Trong phương pháp học máy, các phương pháp học sâu được sử dụng rộng rãi do tốc độ nhanh và độ chính xác cao của nó nhằm nhận dạng hình ảnh, phát hiện đối tượng, phân đoạn ngữ nghĩa như: phương pháp sử dụng mạng CNN phân loại khái niệm ngữ nghĩa cho hình ảnh nhằm chú thích các hình ảnh Web không được gắn nhãn [25], phân đoạn ngữ nghĩa với DeepLab v3 và thuật toán phân đoạn siêu pixel-dịch chuyển nhanh [29], phương pháp điều chỉnh không gian tích hợp vào mạng CNN nhằm giảm nhiễu và phân đoạn hình ảnh hiệu quả hơn [9]... Tuy nhiên, các phương pháp học sâu phát hiện đối tượng, phân đoạn ngữ nghĩa với các mô tả đặc trưng hình ảnh ở mức độ thấp, không thể chỉ ra rõ ràng mối quan hệ giữa các đối tượng hay thiếu nhận thức

ngữ cảnh vì nó thiếu các mô tả ngữ nghĩa nâng cao, do đó việc hiểu thông tin ngữ nghĩa cấp cao theo của hình ảnh theo yêu cầu người dùng trở nên khó khăn [7]. Ontology là một mô hình dữ liệu có thể biểu diễn một cách rõ ràng các khái niệm và các mối quan hệ khác nhau giữa các khái niệm. Do các tính năng định hướng máy tính và dựa trên logic, kỹ thuật Ontology đã được áp dụng rộng rãi cho mô hình hóa và phân tích thông tin như tích hợp ngữ nghĩa của cơ sở dữ liệu không đồng nhất, truy xuất và quản lý thông tin hình ảnh [8, 21]. Vì vậy, nhiều nghiên cứu nhằm tìm kiếm ngữ nghĩa ảnh trên Ontology được quan tâm như: khung khái niệm dựa trên Ontology để truy xuất hình ảnh của các hiện vật bảo tàng [21], nhận diện rủi ro trong xây dựng dựa vào ngữ nghĩa hình ảnh trên Ontology [28], sử dụng Ontology của hệ thống tim mạch của con người để cải thiện việc phân loại các hình ảnh mô học [16]...

Trong cách tiếp cận của chúng tôi, một khung Ontology được xây dựng bán tự động cho bài toán tìm kiếm ảnh theo tiếp cận ngữ nghĩa. Tìm kiếm ảnh dựa trên Ontology OBIR (Ontology-based Image Retrival) được chúng tôi thực hiện theo hai cách: (1) từ văn bản đầu vào của người dùng, xác định từ khóa để tìm kiếm trên Ontology và (2) từ hình ảnh đầu vào, thông qua công cụ tìm kiếm dựa trên phương pháp học máy (đồ thị cụm láng giềng) để trích xuất đặc trưng từ hình ảnh, phân lớp hình ảnh, tra cứu từ vựng thị giác (nhằm giải quyết bài toán chuyển từ một ảnh sang vec-tơ gồm các từ vựng thị giác). Sau đó tạo tự động câu lệnh SPARQL để truy vấn trên Ontology. Kết quả tìm kiếm ảnh trên Ontology là một tập ảnh tương tự cùng với ngữ nghĩa của các hình ảnh. Hệ truy vấn hình ảnh sau khi được tích hợp với Ontology sẽ cải thiện kết quả tìm kiếm cũng như độ tin cậy về mặt ngữ nghĩa cho hệ thống. Các đóng góp chính của bài báo bao gồm: (1) Đề xuất xây dựng một khung Ontology (Frame Ontology) bán tự động cho bộ dữ liệu ảnh dựa trên ngôn ngữ OWL; (2) Đề xuất mô hình tìm kiếm ảnh dựa vào Ontology; (3) Xây dựng thực nghiệm và chứng minh tính đúng đắn của phương pháp đề xuất dựa trên bộ dữ liệu ImageCLEF [12] và Stanford Dogs [14].

Phần còn lại của bài báo gồm: Phần II khảo sát một số công trình liên quan đồng thời phân tích các ưu nhược điểm để chứng minh tính hiệu quả của tìm kiếm ảnh dựa trên Ontology; Phần III trình bày phương pháp xây dựng Ontology bán tự động cho dữ liệu ảnh; Phần IV mô tả hệ tìm kiếm ảnh dựa trên Ontology OnSBIR và Phần V là thực nghiệm, đánh giá so sánh; Kết luận và hướng phát triển tương lai được thể hiện trong Phần VI.

II. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Nhiều công trình nghiên cứu về Ontology cho mục đích phân tích ngữ nghĩa, chú thích và tìm kiếm ngữ nghĩa ảnh. Phương pháp tìm kiếm ảnh dựa trên Ontology đã được đề xuất theo hai hướng chính bao gồm: tìm kiếm ảnh trên Ontology dựa trên văn bản đầu vào [5, 17, 23] và tìm kiếm ảnh dựa trên ảnh đầu vào [8, 15] nhằm liên kết các đặc trưng cấp thấp và ngữ nghĩa cấp cao của ảnh trên Ontology.

Các phương pháp tìm kiếm ảnh theo tiếp cận ngữ nghĩa dựa trên Ontology với văn bản đầu vào được đề xuất trong các nghiên cứu: Vijayarajan, V. và cộng sự (2016) [23] xây dựng một Ontology trên tập ảnh ImageCLEF gồm 20.000 hình ảnh với các chú thích đi kèm mỗi ảnh, cho bài toán SBIR. Quá trình tìm kiếm hình ảnh phụ thuộc vào việc phân tích văn phạm của chú thích để tạo thành các từ khóa mô tả nội dung hình ảnh, tuy nhiên chưa thực hiện phân lớp nội dung hình ảnh từ các đặc trưng cấp thấp để tạo các từ khóa nhằm thực hiện tìm kiếm trên Ontology. Ontology được xây dựng với công cụ Protégé, đầu ra là RDF và OWL. Hình ảnh được truy xuất trên Ontology dựa vào truy vấn SPARQL tạo tự động. Tuy nhiên, khi số lượng ảnh lớn, Ontology xây dựng thủ công trên Protégé tốn nhiều thời gian và nhân lực, đồng thời Ontology trong đề xuất chỉ dừng lại mức độ miền phân lớp và phân cấp miền, các định nghĩa, quan hệ chưa được thực hiện. Mohd K. và cộng sự (2017) [17] đã đề xuất mô hình tìm kiếm ảnh theo tiếp cận ngữ nghĩa với Ontology mờ đa phương thức và Dbpedia, nhằm nâng cao hiệu quả. Ontology mờ được xây dựng bằng cách sử dụng các khái niệm mô tả các đối tượng trong ảnh. Các khái niệm, phạm trù và hình ảnh được liên kết với nhau bằng các giá trị mờ trong Ontology. Tuy nhiên, phương pháp này chỉ tìm kiếm trên từ khóa, không tìm kiếm từ ảnh, do đó chưa liên kết được với đặc trưng cấp thấp. Đồng thời, Ontology chỉ xây dựng trên tập ảnh nhỏ với năm phân lớp ngữ nghĩa, chưa đại diện cho các tập ảnh đa dạng trong thực tế. Nhóm nghiên cứu Bouchakwa, M. (2020) [5] đề xuất cải tiến ngữ nghĩa dựa trên thẻ (Tag) ảnh bằng cách: đầu tiên, các ý nghĩa ngữ nghĩa khác nhau được suy ra từ các truy vấn không rõ ràng của người dùng dựa trên Ontology, sau đó truy vấn ban đầu được định dạng lại bằng cách thực thi một tập hợp các quy tắc ngữ nghĩa trên Ontology. Các hình ảnh trên một tập ảnh được gom cụm theo nội dung ngữ nghĩa và lọc, xếp hạng lại các kết quả truy xuất cụm hình ảnh để sắp xếp theo mức độ thích hợp. Để đánh giá hiệu suất của phương pháp đề xuất đề xuất, thực nghiệm được tiến hành trên bộ sưu tập 25.000 hình ảnh được chia sẻ trên Flickr. Các phương pháp tìm kiếm ảnh theo tiếp cận ngữ nghĩa dựa trên Ontology kết hợp với đặc trưng thị giác được đề xuất trong các nghiên cứu: Manzoor, U. và cộng sự (2015) [15] đề xuất phương pháp tìm kiếm ảnh theo tiếp cận ngữ nghĩa dựa trên Ontology miền cụ thể, để thu thập hình ảnh có liên quan đến tìm kiếm của người dùng. Hệ thống thực hiện tìm kiếm theo hai cách: từ hình ảnh đầu vào là sự kết hợp giữa đặc trưng cấp thấp và Ontology, hoặc từ văn bản do người dùng nhập vào, thực hiện tìm kiếm trực tiếp thên

Ontology. Miền động vật có vú gồm 900 ảnh với 20 loại động vật khác nhau được sử dụng để xây dựng Ontology và thực nghiệm chứng minh tính hiệu quả của đề xuất với độ chính xác là 0,60. Tuy nhiên, Ontology chỉ xây dựng được trên miền giá trị, chưa tạo mối quan hệ phân cấp giữa các lớp cũng như khái niệm ngữ nghĩa cho miền. Tập ảnh xây dựng Ontology là nhỏ, tính kế thừa và khả năng mở rộng chưa được thể hiện trong bài báo. Nhóm nghiên cứu Filali J. (2016) [8] trình bày một hệ thống tìm kiếm hình ảnh dựa trên từ vựng thị giác và Ontology. Đề xuất cho mọi hình ảnh tìm kiếm, xây dựng từ vựng thị giác và Ontology dựa trên các chú thích hình ảnh. Quá trình tìm kiếm hình ảnh được thực hiện bằng cách tích hợp cả đặc trưng thị giác và ngữ nghĩa tương tự. Từ vựng thị giác được lấy ra từ các đặc trưng cấp thấp của hình ảnh và ảnh được tìm kiếm dựa vào khoảng cách Euclide. Các Ontology được làm phong phú thêm bởi các khái niệm và mối quan hệ được trích ra từ tài nguyên từ vựng của BabelNet, thực hiện tìm kiếm ảnh trên Ontology dựa vào độ đo ngữ nghĩa. Kết quả tìm kiếm trên đặc trưng cấp thấp và trên Ontology được giao lại để tìm ra tập ảnh tương tự về mặt nội dung cấp thấp và ngữ nghĩa cấp cao. Hệ thống thực hiện xây dựng Ontology từ 20.000 hình ảnh của tập ảnh ImageCLEF, bởi đây là tập ảnh có một danh sách phân lớp ngữ nghĩa lớn, đại diện cho hầu hết các phân lớp ảnh có thể có. Tuy nhiên, thực nghiệm chưa được đánh giá nên tính hiệu quả của hệ thống về độ chính xác chưa kiểm chứng. Gonçalves, F. M. F. và cộng sự (2018) [11] đề xuất một phương pháp tiếp cận ngữ nghĩa kết hợp CBIR, học không giám sát và kỹ thuật Ontology. Đối với mô hình ngữ nghĩa, Ontology miền đại diện được xây dựng cho các đặc tính và cấu trúc của thực vật. Một cách tiếp cận dựa trên đồ thị được đề xuất để kết hợp thông tin ngữ nghĩa và kết quả tìm kiếm nội dung trực quan cấp thấp. Phương pháp đề xuất được thực nghiệm đánh giá trên tập dữ liệu Oxford Flowers với 102 phân lớp để chứng minh tính hiệu quả. Độ chính xác khi tìm kiếm ảnh có Ontology (0,8539) cao hơn so với CBIR và học không giám sát (0,8020) chứng tỏ, sự kết hợp tìm kiếm ảnh từ đặc trưng cấp thấp với ngữ nghĩa cấp cao trên Ontology cho hiệu suất vượt trội. Tuy nhiên, Ontology này xây dựng trên miền nhỏ, chưa thực hiện định nghĩa các khái niệm miền và chưa làm rõ được mối quan hệ giữa các miền giá trị. Mazo, C. và cộng sự (2020) [16] sử dụng Ontology của hệ thống tim mạch của con người để cải thiện việc phân loại các hình ảnh mô học với phương pháp kết hợp SVM. Phương pháp này cải thiện việc phân loại tự động các hình ảnh mô học và có khả năng nhận ra các mô biểu mô, trước đây không được phát hiện bởi bất kỳ phương pháp thị giác máy tính nào khi so sánh đồng thời, bao gồm một đề xuất của CNN có tên là HistoResNet. Zhang, M. và cộng sự (2020) [28] đề xuất một phương pháp nhận dạng tự động kết hợp phát hiện đối tượng và Ontology để xác định rủi ro trong quá trình xây dựng và ngăn ngừa tai nạn xây dựng. Đầu tiên, mạng CNN được dùng để trích xuất thông tin ngữ nghĩa cấp thấp từ các phần tử ngữ cảnh và các thuộc tính quan hệ không gian phần tử từ hình ảnh xuất từ video giám sát. Tiếp đến, Ontology được thiết lập trong phạm vi của một ngữ cảnh và ngôn ngữ logic của Ontology được dùng để chuyển đổi thông tin ngữ nghĩa cấp thấp của hình ảnh thành ngữ nghĩa cấp cao. Thứ ba, các quy tắc rủi ro được chuyển thành các quy tắc Ontology, và các tình huống rủi ro cao có thể phát sinh thực tế được xác định bởi một công cụ suy luận Pellet. Cuối cùng, một cảnh đào hố móng được lấy làm ví dụ, và kết quả thử nghiệm được sử dụng để xác minh tính khả thi và hiệu quả của phương pháp đề xuất. Phương pháp đề xuất có thể được dùng để nâng cao hiệu quả quản lý an toàn xây dựng. Các nghiên cứu về tìm kiếm ảnh theo tiếp cận ngữ nghĩa được khảo sát, phân tích cho thấy tính hiệu quả của phương pháp tìm kiếm ảnh dựa trên Ontology. Hệ tìm kiếm ảnh trên Ontology cho độ chính xác cao hơn so với tìm kiếm thông thường dựa trên nội dung. Tuy nhiên, việc chỉ sử dụng văn bản đầu vào để truy vấn ngữ nghĩa trên Ontology cho kết quả không tốt bằng sự kết hợp giữa ngữ nghĩa trên Ontology và đặc trưng cấp thấp mô tả nội dung ảnh. Với sự kết hợp này, kết quả truy vấn không chỉ tương tự về nội dung thị giác, mà còn tương tự về mặt ngữ nghĩa. Ontology hiệu quả khi xây dựng trên tập dữ liệu lớn, có nhiều phân lớp đại diện cho nhiều đối tượng trong thực tế như bộ ảnh ImageCLEF, Oxford Flowers,... Từ các ưu nhược điểm của các công trình nghiên cứu liên quan, chúng tôi đề xuất phương pháp tìm kiếm ảnh tiếp cận ngữ nghĩa dựa trên Ontology và nội dung cấp thấp của hình ảnh, sử dụng tập ảnh ImageCLEF là tập ảnh mục tiêu để xây dựng khung Ontology. Đồng thời đề xuất phương pháp bổ sung dữ liệu để làm phong phú thêm ngữ nghĩa cho khung Ontology này.

III. PHƯƠNG PHÁP XÂY DỰNG ONTOLOGY CHO DỮ LIỆU ẢNH

Dữ liệu ảnh luôn tăng trưởng không ngừng, có sự biến đổi thường xuyên của miền và mối quan hệ cấp bậc trong mỗi bộ ảnh, do đó xây dựng Ontology cho hình ảnh là một nhiệm vụ khó khăn, đòi hỏi nguồn nhân lực và thời gian lớn. Xây dựng Ontology phải đảm bảo việc bổ sung dữ liệu đúng để làm phong phú ngữ nghĩa hình ảnh. Chúng tôi đề xuất xây dựng một khung ontogy bán tự động có khả năng bổ sung dữ liệu cho bài toán tìm kiếm ảnh theo tiếp cận ngữ nghĩa.

1. Mô hình xây dựng Ontology bán tự động cho tập dữ liệu ảnh

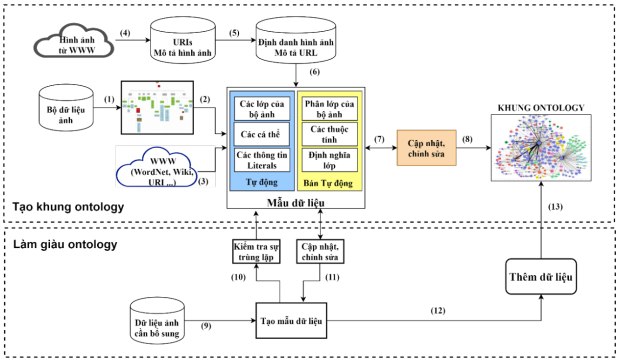
Một mô hình xây dựng khung Ontology (Frame Ontology) bán tự động cho ảnh được đề xuất trong Hình 1, sử

dụng tập ảnh ImageCLEF, là bộ ảnh phong phú về miền (276 lớp) làm nền tảng, kế thừa các hệ thống phân cấp ảnh, có khả năng mở rộng miền và mô tả tốt nhất các quan hệ của ảnh với phân lớp của nó. Khung Ontology bao gồm các thành phần: lớp, phân cấp lớp, các thuộc tính dữ liệu, thuộc tính quan hệ, các thông tin literals cho lớp và hình ảnh, từ điển đồng nghĩa. Các mẫu dữ liệu được tạo ra dựa trên các thành phần này (Hình 2), làm dữ liệu đầu vào cho quá trình xây dựng khung Ontology bán tự động. Mô hình tạo khung Ontology bán tự động bao gồm các bước như sau:

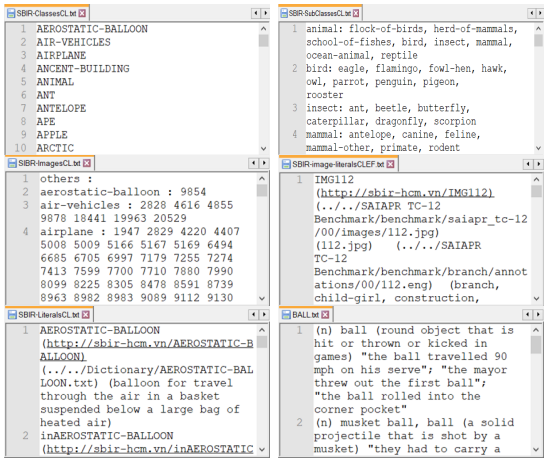
- **Bước 1.** Xác định tập ảnh, kế thừa lớp, phân cấp lớp (1);
- **Bước 2.** Xây dựng mẫu dữ liệu cho các thành phần của khung Ontology, được tạo tập ảnh (2) và từ thông tin WWW (3):
  + Các lớp, các cá thể và thông tin Literals được tạo tự động từ tập ảnh;
  + Phân lớp, các thuộc tính, từ điển được tạo bán tự động từ tập ảnh và WWW;
- **Bước 3.** Ảnh từ WWW, kết xuất các định danh tài nguyên và mô tả (4), các định danh ảnh và URL (5), bổ sung vào các dữ liệu đầu vào (6) cho quá trình tạo Ontology;
- **Bước 4.** Các dữ liệu được tạo từ bước 3 và bước 4 được cập nhật, chỉnh sửa với sự tham gia của chuyên gia (7) và tạo khung Ontology bán tự động (8);

Trong mô hình xây dựng khung Ontology bán tự động, chuyên gia đóng vai trò quan trọng nhằm đảm bảo tính đúng đắn, tin cậy cho Ontology do các dữ liệu được bổ sung thực hiện theo qui trình xử lý dữ liệu, làm sạch dữ liệu, kiểm chứng, kiểm thử dữ liệu,... Bước tham gia chỉnh sửa của chuyên gia được thực hiện thủ công dựa trên các thao tác tự động trước đó như sau:
+ Tạo mẫu các phân lớp của bộ ảnh (SubClass) từ mẫu tập tin lớp (Class), kế thừa từ tập ảnh và từ WWW như cây phân cấp lớp (Hierarchy tree) của ImageNet [13], Ontology (taxonomy) của Dbpedia [6];...
+ Tạo mẫu các định nghĩa lớp hay từ điển đồng nghĩa dựa vào mạng từ WORDNET. Từ WORDNET, tự động lấy về định nghĩa tương ứng của các lớp, sau đó, chuyên gia lọc lại các định nghĩa cho các danh từ và các đồng nghĩa của lớp.
+ Tạo mẫu các thuộc tính quan hệ giữa lớp và các thể hiện (instances) của nó, xác định hình ảnh đại diện phù hợp cho lớp và định nghĩa chính của lớp tùy theo ngữ cảnh của bộ ảnh.

Ontology được tạo bán tự động với sự tham gia chỉnh sửa của chuyên gia có các ưu điểm là giảm chi phí, nhân lực nhưng vẫn đảm bảo tính đúng đắn và tin cậy. Trong khi đó, việc tạo Ontology thủ công trên protégé là không khả



Hình 1. Mô hình xây dựng và bổ sung dữ liệu cho một khung Ontology



Hình 2. Các dữ liệu mẫu cho khung Ontology bán tự động

thi khi số lượng ảnh nhiều, chi phí và nhân lực rất lớn, còn tạo Ontology tự động thì chi phí và nhân lực được giảm đi rất nhiều nhưng không đảm bảo tính chính xác và tin cậy. Các thuật toán để xây dựng Ontology bao gồm tạo các lớp, lớp phân cấp, các cá thể được mô tả như sau:

Thuật toán 1: Tạo lớp cho Ontology (COC).

```
Đầu vào: C = {classi, i = 1..N}, Ontology;
Đầu ra: Ontology;
begin
    foreach class in C do
        Sub = "sbir:" + Class;
        Pre = "rdf:type";
        Obj = "owl:" + "Class";
        Ontology = Ontology ∪ Triple(Sub, Pre, Obj);
    end
    return Ontology;
end
```

2. Bổ sung dữ liệu ảnh cho khung Ontology

Bổ sung dữ liệu cho khung Ontology là bổ sung các mô tả ngữ nghĩa và mở rộng cấu trúc của Ontology, làm cho

Thuật toán 2: Tạo phân cấp lớp cho Ontology (COCS).

```
Đầu vào: superclass, subClass, Ontology;
Đầu ra: Ontology;
begin
    foreach class in C do
        Sub = "sbir:" + subClass;
        Pre = "rdfs:subClassOf ";
        Obj = "owl:" + " superClass ";
        Ontology = Ontology ∪ Triple(Sub, Pre, Obj);
    end
    return Ontology;
end
```

Thuật toán 3: Tạo các cá thể cho Ontology (CIC).

```
Đầu vào: C = {classi, i=1..N}, Individual, Ontology;
Đầu ra: Ontology;
begin
    foreach class in C do
        Sub = "sbir:" + Individual.ElementAt(i);
        Pre = "rdf:type";
        Obj = "owl:NamedIndividual";
        Ontology = Ontology ∪ Triple(Sub, Pre, Obj);
    end
    return Ontology;
end
```

nó chứa nhiều thông tin hơn, giàu ngữ nghĩa hơn. Các lớp, phân cấp lớp, các thuộc tính, quan hệ, các cá thể và các mô tả ngữ nghĩa được bổ sung bán tự động cho khung Ontology ban đầu. Việc bổ sung này đảm bảo các lớp, phân cấp lớp, các thuộc tính, các định nghĩa trong từ điển và các cá thể mới có thể được thêm vào khung Ontology mà không phá vỡ cấu trúc của nó. Mô hình bổ sung dữ liệu cho khung Ontology được đề xuất trong Hình 1. Mô hình này bao gồm hai giai đoạn, giai đoạn tạo khung Ontology (từ bước 1-8) và giai đoạn bổ sung cho khung Ontology. Quá trình bổ sung dữ liệu cho Ontology được thực hiện như sau:

- **Bước 1.** Xác định tập ảnh ảnh để bổ sung cho một khung Ontology đã xây dựng;
- **Bước 2.** Dựa vào mẫu dữ liệu được tạo từ các thành phần của khung Ontology, tạo mẫu dữ liệu để bổ sung bán tự động cho khung Ontology (9);
- **Bước 3.** Kiểm tra tự động sự trùng lặp về lớp, cá thể, thuộc tính... giữa mẫu dữ liệu cần bổ sung vào mẫu dữ liệu của khung Ontology (10).
- **Bước 4.** Cập nhật, chỉnh sửa mẫu dữ liệu với sự tham gia của chuyên gia (11);
- **Bước 5.** Dữ liệu được thêm (12) vào khung Ontology để làm phong phú thêm cho ngữ nghĩa các tập ảnh (13).

Ứng dụng tạo khung Ontology và bổ sung dữ liệu SBIR-Ontology được xây dựng trên mô hình đã đề xuất, trên nền tảng dotNET Framework 4.8, ngôn ngữ lập trình C#, lưu trữ bằng Notation 3 syntax (N3). Khung Ontology được tạo từ tập dữ liệu ảnh ImageCLEF (20.000 ảnh) và bổ sung thêm với tập ảnh Stanford Dogs (20.580 ảnh), lưu trữ tại tập tin SBIR-Ontology.n3 (Hình 3).

Hình 3. Ontology theo định dạng N3

Ví dụ của quá trình bổ sung dữ liệu được thể hiện trong Hình 4, cho thấy phân cấp trước và sau khi bổ sung vào khung Ontology. Ban đầu (Hình 4a) phân lớp của khung Ontology có lớp DOG và lớp này không có các lớp con, sau đó, khi bổ sung dữ liệu từ tập ảnh các loại chó của Stanford Dogs, thì phân lớp này tạo thêm các con (Hình 4b). Việc bổ sung này đảm bảo các lớp, phân cấp lớp, các thuộc tính, các định nghĩa trong từ điển và các cá thể mới có thể được thêm vào khung Ontology mà không phá vỡ cấu trúc ban đầu.

(a) (b)

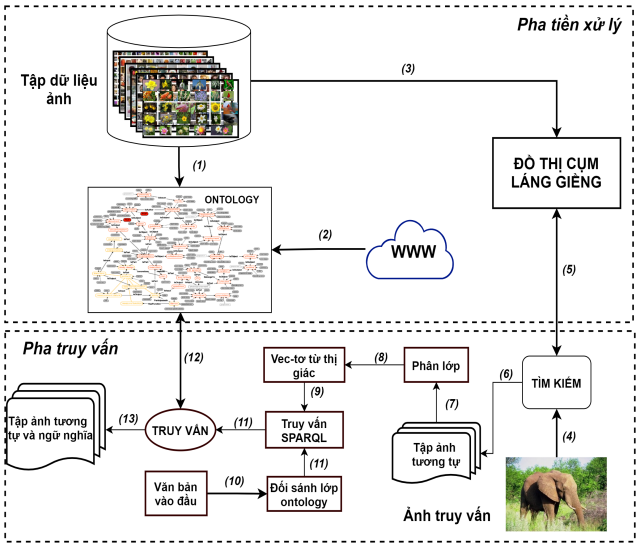
Hình 4. Trực quan phân cấp lớp trước (a) và sau (b) khi bổ sung dữ liệu vào Ontology

IV. HỆ TÌM KIẾM ẢNH DỰA TRÊN ONTOLOGY

1. Mô hình tìm kiếm ảnh

Một hệ tìm kiếm theo ngữ nghĩa dựa trên Ontology, onSBIR, được xây dựng bao gồm 2 pha được mô tả trong Hình 5 như sau:

- **Pha tiền xử lý:**
  (1) Các hình ảnh trong tập ảnh được tổ chức lưu trữ trên cấu trúc đồ thị cụm láng giềng;
  (2) Xây dựng, làm giàu một Ontology bán tự động dựa trên ngôn ngữ bộ ba RDF từ tập ảnh và từ WWW;
- **Pha tìm kiếm:**
  (1) Với một ảnh đầu vào, hệ thống thực hiện tìm kiếm trên đồ thị cụm láng giềng lấy ra tập các ảnh tương tự theo nội dung được sắp xếp theo độ đo, phân lớp tập ảnh này với thuật toán k-NN để tìm tập từ vựng thị giác; Với đầu vào là một đoạn văn bản, hệ thống thực hiện tìm các từ khóa phù hợp với các lớp trong Ontology;
  (2) Tự động tạo câu truy vấn SPARQL để tìm kiếm ảnh trên Ontology;
  (3) Kết xuất tập ảnh tương tự theo ngữ nghĩa trên Ontology.



Hình 5. Mô hình hệ tìm kiếm theo tiếp cận ngữ nghĩa onSBIR

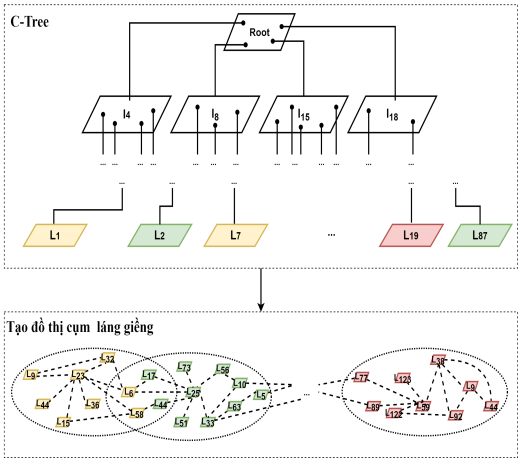
2. Các thành phần của hệ truy vấn onSBIR

a) Trích xuất đặc trưng cấp thấp

Độ chính xác của tìm kiếm ảnh phụ thuộc vào sự lựa chọn các đặc trưng cấp thấp, do đó, để phát huy các đặc trưng trong hệ thống tìm kiếm ảnh, chúng tôi sử dụng các đặc trưng cấp thấp về màu sắc, kết cấu, hình dạng để tạo thành bộ mô tả đặc trưng kết hợp màu sắc được trích xuất dựa trên bộ mô tả màu chủ đạo DCD của MPEG-7, đặc trưng kết cấu được trích xuất dựa vào độ tương phản, phép lọc tần số cao, phép lọc Sobel, phép lọc Gaussian và phương pháp LoG, đặc trưng hình dạng dựa trên phương pháp Laplacian... Một vec-tơ đặc trưng cấp thấp gồm 81 chiều được trích xuất cho bài toán tìm kiếm ảnh trong bài báo này.

b) Đồ thị cụm láng giềng

Đồ thị cụm láng giềng được cải tiến từ cấu trúc cây phân cụm cân bằng C-Tree [18, 19] đã được chúng tôi xây dựng. C-Tree có thể lưu trữ được dữ liệu lớn, hiệu quả cho bài toán tìm kiếm ảnh với thời gian tìm kiếm nhanh, độ chính xác khá cao. Tuy nhiên, nhược điểm chính của C-Tree là mỗi lần tách nút, các phần tử tương tự nhau có thể bị tách thành các nút khác nhau, trong trường hợp xấu nhất, các nút này có thể nằm trên các nhánh riêng biệt. Do đó, quá trình tìm kiếm ảnh trên cây C-Tree có thể sẽ bỏ sót các phần tử tương tự đã bị chuyển nhánh. Điều này làm ảnh hưởng đến hiệu suất tìm kiếm trên C-Tree. Vì vậy, đồ thị gom cụm láng giềng (Hình 6), là sự kết hợp giữa đồ thị gom cụm và cây C-Tree, được đề xuất như sau: mỗi nút lá trên cây C-Tree trong quá trình huấn luyện sẽ tìm được các cụm láng giềng lân cận theo ba mức: mức một nếu khoảng cách giữa hai tâm nút bé hơn θ, mức hai nếu lớp đại diện của hai nút trùng nhau và mức ba nếu lớp đại diện của hai nút có quan hệ phân lớp cha-con. Các mức láng giềng được đánh dấu để tạo thành một đồ thị cụm liên quan với nhau về độ đo và quan hệ cấp bậc (cha-con). Đồ thị gom cụm láng giềng cho phép thời gian tìm kiếm nhanh hơn và chi phí bộ nhớ ít hơn so với đồ thị gom cụm truyền thống.



Hình 6. Đồ thị gom cụm láng giềng

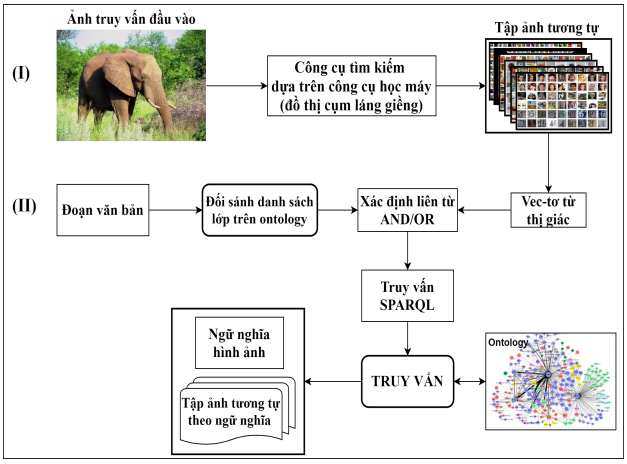
c) Tìm kiếm ảnh dựa trên Ontology

Tìm kiếm ảnh theo tiếp cận ngữ nghĩa dựa trên Ontology được thực hiện theo hai cách: tìm kiếm dựa trên ảnh đầu vào và tìm kiếm dựa đoạn văn bản, mô tả trong Hình 7.

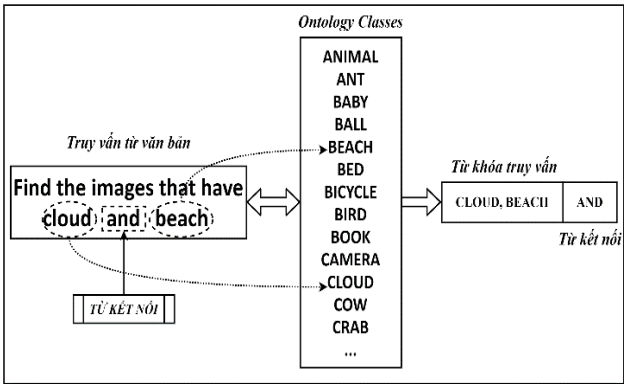
(I) Tìm kiếm từ ảnh đầu vào: ảnh đầu vào có thể chứa một đối tượng hoặc nhiều đối tượng, dựa vào đồ thị

cụm láng giềng để tìm tập ảnh tương tự. Sau đó thực hiện thuật toán phân lớp k-NN để tìm tập từ vựng thị giác, là vec-tơ chứa một hay nhiều lớp ngữ nghĩa của ảnh và tự động tạo câu lệnh SPARQL (AND hoặc OR), từ đó truy vấn trên Ontology để tìm tập các ảnh tương tự va ngữ nghĩa của nó;

(II) Tìm kiếm từ văn bản đầu vào: Từ một đoạn văn bản đầu vào, thực hiện đối sánh danh sách các lớp Ontology để tìm từ khóa truy vấn và từ kết nối AND hoặc OR trong ngữ nghĩa đoạn văn bản, từ đó tự động tạo câu lệnh SPARQL để thực hiện tìm kiếm trên Ontology, truy xuất tập các ảnh tương tự và ngữ nghĩa của tập ảnh. Hình 8 là một ví dụ về phân tích đoạn văn bản nhằm tìm các từ khóa cho câu truy vấn SPARQL.



Hình 7. Các cách tìm kiếm ảnh trên Ontology



Hình 8. Một ví dụ về phân tích văn bản tìm kiếm

Thuật toán phân tích văn bản text để tìm các từ khóa cho truy vấn SPRQL được thực hiện như sau:

Thuật toán tự động tạo câu truy vấn SPRQL được thực hiện như sau:

Thuật toán 4: Trích xuất từ khóa tìm kiếm từ text (GLCT).

```
1  Đầu vào: Text và danh sách lớp LClass
2  Đầu ra: danh sách từ khóa ListClass
3  begin
4      text=txtQuery.Text;
5      Classes=text.Split;
6      strClasses=LClass.Split;
7      foreach (cla ∈ Class) do
8          If (cla ∈ strClass) then
9          ListClass.Add(cla);
10         CSP(ListClass);
11         end
12     end
13 end
```

Thuật toán 5: Tự động tạo câu truy vấn SPARQL (CSP)

```
1  Đầu vào: tập từ vựng thị giác W
2  Đầu ra: câu lệnh SPARQL
3  begin
4      SPARQL= ∅;
5      n = W.count;
6      SELECT DISTINCT ?Im
7      WHERE {";
8      For (i=0..n) do
9      SPARQL+="<subject > : W(i)+ "rdf:type" +
10     <object >: +?Img''+"UNION/AND";
11     end
12     SPARQL+ =}";
13     Return SPARQL;
14 end
```

V. THỰC NGHIỆM VÀ ĐÁNH GIÁ

1. Môi trường thực nghiệm

Hệ truy vấn OnSBIR được xây dựng dựa trên nền tảng dotNET Framework 4.8, ngôn ngữ lập trình C#. Cấu hình máy tính của thực nghiệm: Intel(R) CoreTM i7-8750H, CPU 2,70GHz, RAM 8GB. Tập dữ liệu sử dụng trong thực nghiệm là ImageCLEF và StanfordDogs, được mô tả trong Bảng I.

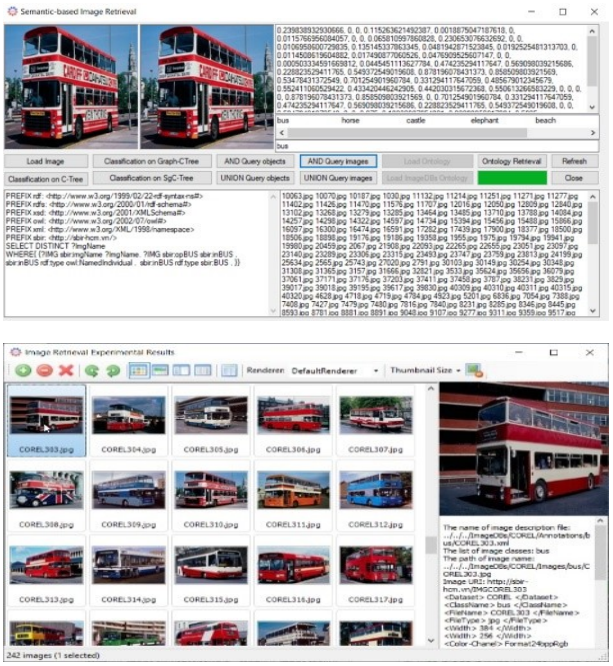
Tập ảnh ImageCLEF bao gồm 20.000 ảnh đa đối tượng, mỗi ảnh có nhiều đối tượng với nhiều chủ đề khác nhau. Tập ảnh này có 276 lớp khác nhau và được trộn lẫn với nhau trong 39 thư mục. Tập ảnh Stanford Dogs gồm 20.580 ảnh, thuộc hệ thống ImageNet, do nhóm nghiên cứu A. Khosla xây dựng vào năm 2011. Đây là một bộ ảnh chỉ có một đối tượng, bao gồm 120 loại chó trên thế giới.

Bảng I
CÁC TẬP DỮ LIỆU ẢNH ĐƯỢC THỰC NGHIỆM

STT	Tên tập ảnh	Số lượng ảnh	Số thư mục ảnh	Số chủ đề ảnh	Kích thước
1	ImageCLEF	20.000	39	276	1,6 GB
2	Stanford Dogs	20.580	120	120	778 MB

2. Ứng dụng

Với một ảnh đầu vào, đặc trưng ảnh được trích xuất để tìm tập ảnh tương tự dựa trên đồ thị cụm láng giềng, phân lớp ảnh để tìm tập từ vựng thị giác, sau đó tự động tạo câu lệnh SPARQL để tìm danh sách các ảnh có cùng ngữ nghĩa với ảnh đầu vào, được thể hiện trong Hình 9. Đồng thời, khái niệm ngữ nghĩa của phân lớp được trích xuất từ tập từ điển đồng nghĩa của Ontology (Hình 10).



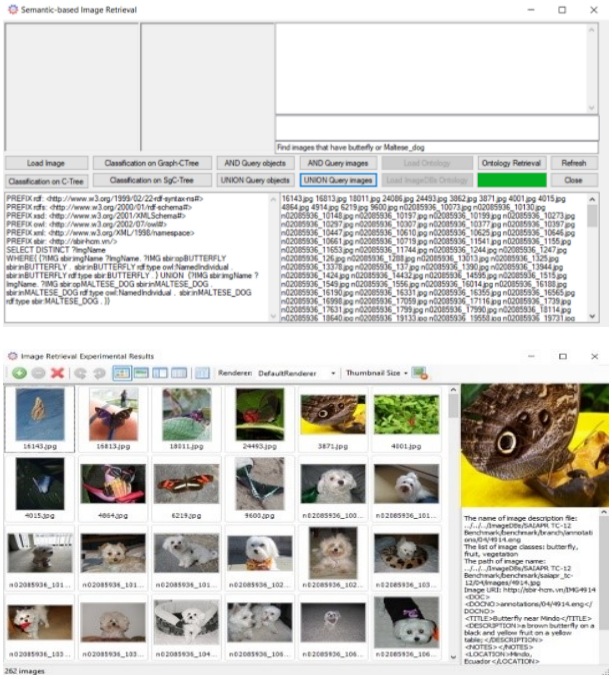
Hình 9. Tìm kiếm ảnh trên Ontology từ ảnh đầu vào

Với đầu vào là một đoạn văn bản, hệ thống đối sánh với các lớp có trong Ontology và từ vựng kết nối (AND/OR) để tìm các từ khóa và tạo câu truy vấn SPARQL tự động để tìm kiếm trên Ontology tập các hình ảnh tương tự về ngữ nghĩa (Hình 11).

3. Đánh giá kết quả

Để đánh giá hiệu quả tìm kiếm ảnh trên Ontology của hệ tìm kiếm onSBIR, các giá trị như độ chính xác (precision), độ phủ (recall), độ dung hòa (F-measure) và thời gian tìm

Hình 10. Khái niệm ngữ nghĩa cho phân lớp



Hình 11. Tìm kiếm ảnh trên Ontology từ văn bản đầu vào

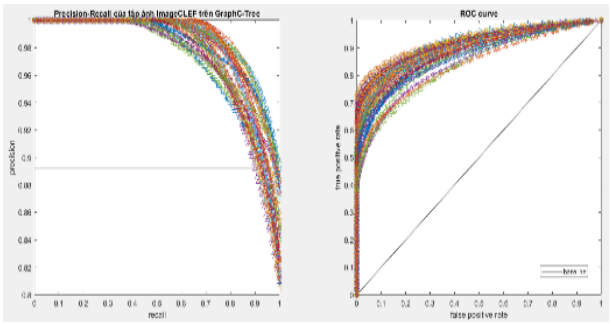
kiếm được tính toán. Bảng II là hiệu suất tìm kiếm trên Ontology của các tập ảnh ImageCLEF và Stanford Dogs. Kết quả tìm kiếm ảnh dựa trên Ontology được so sánh với phương pháp tìm kiếm ảnh dựa vào đồ thị cụm láng giềng (Bảng III). Bảng III cho thấy độ chính xác, độ phủ của hệ thống tìm kiếm ảnh dựa trên Ontology tốt hơn so với hệ thống tìm kiếm ảnh theo nội dung, tuy nhiên thời gian tìm kiếm lâu hơn, do OnSBIR kết hợp tìm kiếm trên cả đồ thị cụm láng giềng và Ontology. Điều này chứng tỏ, phương pháp tìm kiếm ảnh trên Ontology nâng cao độ chính xác và độ phủ cho tìm kiếm.

Bảng II
HIỆU SUẤT TÌM KIẾM TRÊN ONTOLOGY

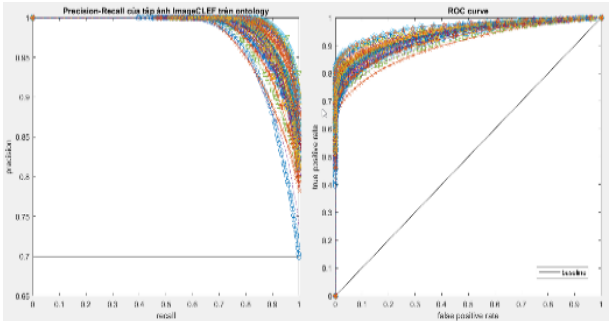
Tập ảnh	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
ImageCLEF	0,932574	0,916225	0,926373	248,5511
Stanford Dogs	0,87395	0,865369	0,869612	284,3384

Bảng III
SO SÁNH HIỆU SUẤT TÌM KIẾM ẢNH

Tập ảnh	Phương pháp	Độ chính xác trung bình	Độ phủ trung bình	Thời gian truy vấn trung bình (ms)
ImageCLEF	OnSBIR	**0,932574**	**0,916225**	248,5511
	Đồ thị cụm láng giềng	0,839814	0,780735	**239,9458**
Stanford Dogs	OnSBIR	**0,87395**	**0,865369**	284,3384
	Đồ thị cụm láng giềng	0,826416	0,513825	**261,8991**



(a)



(b)

Hình 12. Hiệu suất trên đồ thị cụm láng giềng (a) và Ontology (b) của tập ảnh ImageCLEF

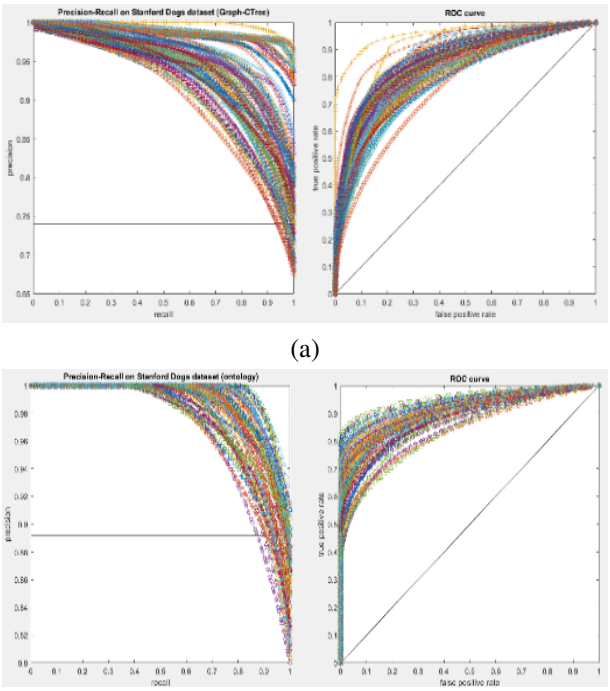
Ngoài ra, các đồ thị Precision-Recall và ROC dựa trên Ontology và dựa trên đồ thị cụm láng giềng (Hình 12, Hình 13) được biểu thị. Mỗi đường cong trên đồ thị Precision – Recall là một thư mục ảnh. Diện tích dưới đường cong càng lớn, độ chính xác truy vấn càng cao. Đồng thời, đường cong ROC cho biết tỷ lệ kết quả truy vấn đúng và sai, diện tích dưới đường cong ROC càng lớn, tỷ lệ truy vấn đúng càng cao. Kết quả thực nghiệm được thể hiện trên các đồ thị cho thấy, diện tích dưới đường cong đồ thị Precision-Recall và ROC của OnSBIR đều lớn hơn so với kết quả của đồ thị cụm láng giềng, chứng tỏ độ chính xác và kết quả truy vấn đúng đều tốt hơn. Từ các kết luận này, có thể thấy tính đúng đắn của các phương pháp tìm kiếm ảnh dựa trên Ontology, nhằm nâng cao hiệu quả tìm kiếm ảnh.

Bảng IV
SO SÁNH ĐỘ CHÍNH XÁC TÌM KIẾM ẢNH GIỮA CÁC PHƯƠNG PHÁP TRÊN TẬP ẢNH IMAGECLEF

Phương pháp	Năm	Độ chính xác trung bình
Mô hình Ontology cho ảnh O-V-A[23]	2016	0,46
Đồ thị mẫu cho Ontology hình ảnh[1]	2017	0,3513
Phương pháp HDLA (hybrid deep learning architecture)[3]	2018	0,797
Phương pháp SDCH (Semantic Deep Cross-modal Hashing)[27]	2019	0,803
Phương pháp CPAH (Consistency Preserving Adversarial Hashing)[26]	2020	0,8324
OnSBIR		**0,932574**

Chúng tôi thực hiện so sánh kết quả tìm kiếm ảnh trên Ontology (onSBIR) với các phương pháp khác trên cùng tập ảnh ImageCLEF. Bảng IV cho thấy, độ chính xác tìm kiếm ảnh dựa trên Ontology được đề xuất trong bài báo vượt trội hơn các phương pháp khác trên tập ảnh ImageCLEF.

Chúng tôi so sánh với các phương pháp có sử dụng Ontology: Vijayarajan V. và cộng sự (2016) [23] chỉ thực hiện tìm kiếm trên Ontology dựa trên văn bản/từ khóa, do đó, hiệu suất tìm kiếm chưa cao; Nhóm nghiên cứu Allani O. (2017) [1] thực hiện kết hợp giữa đặc trưng cấp thấp trên đồ thị và ngữ nghĩa cấp cao trên Ontology, tuy



Hình 13. Hiệu suất trên đồ thị cụm láng giềng (a) và Ontology (b) của tập ảnh Stanford Dogs

nhiên, việc xây dựng Ontology hoàn toàn tự động, không có sự chỉnh sửa, cập nhật của chuyên gia, do đó, hiệu suất tìm kiếm là không cao. Ngoài ra, các phương pháp tìm kiếm ngữ nghĩa ảnh dựa vào các kỹ thuật học máy hiện đại cũng được so sánh với phương pháp của chúng tôi: Phương pháp HDLA (Hybrid Deep Learning Architecture) mô hình hóa các mối tương quan bậc cao giữa các từ trực quan nhằm giảm khoảng cách ngữ nghĩa trong tìm kiếm ảnh [3]; Phương pháp SDCH (Semantic Deep Cross-modal Hashing) sử dụng mạng CNN để trích xuất đặc trưng và hàm băm sâu để lấy ngữ nghĩa ảnh [27]; Phương pháp CPAH (Consistency Preserving Adversarial Hashing) nhằm khai thác tính nhất quán về ngữ nghĩa cho tìm kiếm ảnh, các đặc trưng được trích xuất dựa trên mạng CNN[26]. Từ các kết quả so sánh chứng tỏ, phương pháp tìm kiếm ảnh dựa trên Ontology được đề xuất trong bài báo là đúng đắn và hiệu quả.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, một khung Ontology bán tự động được xây dựng và bổ sung để làm phong phú thêm ngữ nghĩa. Bài toán tìm kiếm trên Ontology được thực hiện theo hai cách: tìm kiếm dựa trên ảnh đầu vào và tìm kiếm từ văn bản. Câu truy vấn SPARQL được tạo ra tự động để tìm kiếm trên Ontology. Hệ tìm kiếm ảnh theo ngữ nghĩa dựa trên Ontology (OnSBIR) được xây dựng. Thực nghiệm được triển khai trên tập ảnh ImageCLEF và Stanford Dogs cho độ chính xác cao, lần lượt là 0,932574 và 0,87395. So sánh với phương pháp tìm kiếm ảnh theo nội dung dựa trên đồ thị cụm láng giềng cho thấy hiệu suất tìm kiếm ảnh dựa trên Ontology là vượt trội. Đồng thời, chúng tôi tiến hành so sánh phương pháp đề xuất với các phương pháp khác trên cùng tập ảnh ImageCLEF cũng cho độ chính xác tốt hơn. Điều này chứng tỏ, các đề xuất của chúng tôi trong bài báo là đúng đắn và hiệu quả. Tuy nhiên, phương pháp đề xuất của chúng tôi, Ontology chưa thực hiện mối quan hệ ngữ nghĩa trong ảnh, vì vậy, trong định hướng tương lai, chúng tôi bổ sung nhiều tập ảnh hơn, làm giàu ngữ nghĩa Ontology với các mối quan hệ trong ảnh, xác định vị trí đối tượng ảnh.

LỜI CẢM ƠN

Chúng tôi xin trân trọng cám ơn Khoa Công nghệ thông tin – Đại học Khoa học- Đại học Huế, nhóm nghiên cứu SBIR-HCM đã góp ý chuyên môn cho nghiên cứu này. Chúng tôi xin trân trọng cảm ơn Trường Đại học Kinh tế – Đại học Đà Nẵng đã tạo điều kiện về cơ sở vất chất giúp chúng tôi hoàn thành bài nghiên cứu này.

TÀI LIỆU THAM KHẢO

[1] Allani O., Zghal H. B., Mellouli N., Akdag H., "Pattern graph-based image retrieval system combining semantic and visual features," *Multimedia Tools and Applications*, vol. 76(19), pp. 20287-20316, 2017.

[2] Alzubaidi M. A., "A new strategy for bridging the semantic gap in image retrieval," *International Journal of Computational Science and Engineering*, vol. 14(1), pp. 27-43, 2017.

[3] Arun K. S., Govindan V. K., "A hybrid deep learning architecture for latent topic-based image retrieval," *Data Science and Engineering*, vol. 3(2), pp. 166-195, 2018.

[4] Bella M. I. T., Vasuki A., "An efficient image retrieval framework using fused information feature," *Computers & Electrical Engineering*, vol. 75, pp. 46-60, 2019.

[5] Bouchakwa M., Ayadi Y., Amous I., "Multi-level diversification approach of semantic-based image retrieval results," *Progress in Artificial Intelligence*, vol. 9(1), pp. 1-30, 2020.

[6] Dbpedia ontology, https://dbpedia.org/ontology/, last accessed 2021/03/20.

[7] Fang W., Ding L., Zhong B., Love P. E., & Luo, H., "Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach," *Advanced Engineering Informatics*, vol. 37, pp. 139-149, 2018.

[8] Filali J., Zghal H., & Martinet J., "Towards Visual Vocabulary and Ontology-based Image Retrieval System," *In International Conference on Agents and Artificial Intelligence*, vol. 2, pp. 560-565, 2016.

[9] Jia F., Liu J., Tai X. C., "A regularized convolutional neural network for semantic image segmentation," *Analysis and Applications*, vol. 19(01), pp. 147-165, 2021.

[10] Garg M., Dhiman G., "A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants," *Neural Computing and Applications*, pp. 1-18, 2020.

[11] Gonçalves F. M. F., Guilherme I. R., Pedronette D. C. G., "Semantic guided interactive image retrieval for plant identification," *Expert Systems with Applications*, vol. 91, pp.12-26, 2018.
[12] ImageCLEF Homepage, https://www.imageclef.org/, last accessed 2021/03/21.
[13] ImageNET Hierarchy, https://observablehq.com/@mbostock/imagenet-hierarchy, last accessed 2021/03/20.
[14] Khosla A., Jayadevaprakash N., Yao B., Li F. F., "Novel dataset for fine-grained image categorization: Stanford dogs," *In: Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, no. 1, 2011.
[15] Manzoor U., Balubaid M. A., Zafar B., Umar H., Khan M. S., "Semantic image retrieval: An ontology based approach," *International Journal of Advanced Research in Artificial Intelligence*, vol. 4(4), pp. 1-8, 2015.
[16] Mazo C., Alegre E., Trujillo M., "Using an ontology of the human cardiovascular system to improve the classification of histological images," *Scientific Reports*, vol. 10(1), pp. 1-14, 2020.
[17] Mohd K., Yanti I.A., Mohd N., Shahrul A., Bloechle M., "Semantic text-based image retrieval with multi-modality ontology and DBpedia," *The Electronic Library*, vol. 35, no. 6, pp. 1191-1214, 2017.
[18] Nhi, N. T. U., Van T. T., Le T. M., "A self-balanced clustering tree for Semantic-based image retrieval," *Journal of Computer Science and Cybernetics*, vol. 36(1), pp. 49-67, 2020.
[19] Nhi N. T. U., Van T. T., Le T. M., "Semantic-Based Image Retrieval Using Balanced Clustering Tree," *In WorldCIST*, vol.2, pp. 416-427, 2021.
[20] Shati N. M., Khalid Ibrahim N., Hasan T. M., "A review of image retrieval based on ontology model," *Journal of Al-Qadisiyah for computer science and mathematics*, vol. 12(1), pp.10, 2020.
[21] Sharma M. K., Siddiqui T. J., "An ontology based framework for retrieval of museum artifacts." *Procedia Comput. Sci.*, vol. 84, pp. 169–176, 2016.
[22] Singh V., Gupta R., "Novel framework of semantic based image reterival by convoluted features with non-linear mapping in cyberspace," *International Journal of Recent Technology and Engineering*, pp. 2277-3878, 2019.
[23] Vijayarajan V., Dinakaran M., Tejaswin P., Lohani M., "A generic framework for ontology-based information retrieval and image retrieval in web data," *Human-centric Computing and Information Sciences*, vol 6(1), pp. 18-27, 2016.
[24] WORDNET Homepage, https://wordnet.princeton.edu/, last accessed 2021/03/17
[25] Xu H., Huang C., Wang D., "Enhancing semantic image retrieval with limited labeled examples via deep learning," *Knowledge-Based Systems*, vol. 163, pp. 252-266, 2019.
[26] Xie D., Deng C., Li C., Liu X., Tao D., "Multi-task consistency-preserving adversarial hashing for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 29, pp. 3626-3637, 2020.
[27] Yan C., Bai X., Wang S., Zhou J., Hancock E. R., "Cross-modal hashing with semantic deep embedding," *Neurocomputing*, vol. 337, pp. 58-66, 2019.
[28] Zhang M., Zhu M., Zhao X., "Recognition of High-Risk Scenarios in Building Construction Based on Image Semantics," *Journal of Computing in Civil Engineering*, vol. 34(4), 2020.
[29] Zhang S., Ma Z., Zhang G., Lei T., Zhang R., Cui Y., "Semantic image segmentation with deep convolutional neural networks and quick shift," *Symmetry*, vol. 12(3), pp. 427-439, 2020.

SƠ LƯỢC VỀ TÁC GIẢ

Nguyễn Thị Uyên Nhi

Sinh năm 1985, nhận bằng cử nhân và thạc sĩ chuyên ngành Khoa học máy tính và kỹ thuật tính toán, tại trường Đại học tổng hợp kỹ thuật Volgagrad, Liên bang Nga, lần lượt vào các năm 2008, 2010. Hiện đang là NCS ngành Khoa học máy tính tại Trường đại học Khoa học, Đại học Huế. Lĩnh vực nghiên cứu: xử lý ảnh, tìm kiếm ảnh, cơ sở dữ liệu mờ.
Email: nhintu@due.edu.vn

Văn Thế Thành

Sinh năm 1979, tốt nghiệp chuyên ngành Toán tin tại Đại học Khoa học Tự nhiên - Đại học Quốc gia TP.HCM vào năm 2001, nhận bằng Thạc sĩ Khoa học Máy tính tại Đại học Quốc gia TP.HCM vào năm 2008. Năm 2016, nhận bằng Tiến sĩ Khoa học Máy tính tại Đại học Khoa học, Đại học Huế. Lĩnh vực nghiên cứu: xử lý ảnh, khai thác dữ liệu ảnh và tìm kiếm ảnh.
Email: thanhvt@hufi.edu.vn

Lê Mạnh Thạnh

Sinh năm 1953, nhận bằng Tiến sĩ ngành khoa học máy tính tại Đại học Budapest (ELTE), Hungary vào năm 1994. Ông trở thành Phó giáo sư tại trường Đại học Khoa học, Đại học Huế, Việt Nam, vào năm 2004. Lĩnh vực nghiên cứu: cơ sở dữ liệu, cơ sở tri thức và lập trình logic.
Email: lmthanh@hueuni.edu.vn

Giải pháp cảnh báo tránh va chạm dựa trên dữ liệu môi trường và đặc điểm điều khiển phương tiện

Quách Hải Thọ[1], Huỳnh Công Pháp[2], Phạm Anh Phương[3]
[1] Trường Đại học Nghệ thuật, Đại học Huế
[2] Trường Đại học Công nghệ Thông tin và Truyền thông Việt - Hàn, Đại học Đà Nẵng
[3] Khoa Tin học, Trường Đại học Sư phạm, Đại học Đà Nẵng

Tác giả liên hệ: Quách Hải Thọ, qhaitho@hueuni.edu.vn
Ngày nhận bài: 13/04/2021, ngày sửa chữa: 01/06/2021, ngày duyệt đăng: 12/06/2021
Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.963

Tóm tắt: Trong quá trình chuyển động của phương tiện khi tham giao thông, các yếu tố cần được xét đến là các tính năng an toàn để tạo nên sự thoải mái cho người ngồi trên xe. Trong bài báo này, với việc phân tích mối quan hệ giữa thời gian tránh va chạm kết hợp với dữ liệu thông tin môi trường, cùng với các thông số dựa trên đặc điểm của người điều khiển phương tiện, từ đó đề xuất giải pháp mô hình cảnh báo tránh va chạm cho phương tiện. Đặc điểm của mô hình cảnh báo tránh va chạm này có thể thích ứng với nhiều điều kiện điều khiển phương tiện khác nhau và đưa ra cảnh báo thích hợp. Kết quả mô phỏng được thực hiện trong môi trường mô phỏng Matlap để chứng minh giải pháp đề xuất hoạt động hiệu quả với những điểm tối ưu như giảm thiểu rủi ro va chạm và cải thiện mức độ an toàn khi điều khiển phương tiện.

Từ khóa: *Thời gian va chạm, lập quy hoạch đường đi, lập kế hoạch chuyển động, hệ thống giao thông thông minh..*

Title: **Solutions on Collision Avoidance based on Environmental Data and Vehicle Control Characteristics**

Abstract: In the process of movement of vehicles when participating in traffic, the factors that need to be taken into account are safety features to create comfort for passen-gers and drivers. In this article, with the analysis of the relationship between time to collision combined with environmental information data, along with parameters based on the characteristics of the vehicle driver, thereby proposing a collision warning model solution for the vehicle. The characteristics of this collision warn-ing model can adapt to a variety of vehicle control conditions and give an appro-priate warning threshold. Simulation results are carried out in a Matlap simulation environment to prove that the proposed solution works efficiently with optimal points such as minimizing the risk of collisions and improving the safety level when driving vehicles.

Keywords: *Time to collision, path planning, motion planning, intelligent transportation systems.*

I. GIỚI THIỆU

Trong những năm gần đây, với nhiều kết quả nghiên cứu về bài toán cảnh báo va chạm cho phương tiện tham gia giao thông đã có thể được phân thành 2 hướng, gồm thuật toán xác định về khoảng cách an toàn [5] và thuật toán xác định về thời gian an toàn [8]. Trong đó, thuật toán xác định về thời gian an toàn sử dụng giá trị thời gian va chạm, sẽ thực hiện việc so sánh thời gian va chạm giữa 2 đối tượng với ngưỡng thời gian an toàn, để từ đó đưa ra quyết định quá trình chuyển động của xe đang ở trạng thái an toàn hay không. Nhóm thuật toán xác định về khoảng cách an toàn sẽ đề cập đến khoảng cách tránh va chạm tối thiểu giữa xe chủ và chướng ngại vật, đây cũng là khoảng cách mà xe chủ cần phải duy trì để tránh va chạm với chướng ngại vật trong điều kiện hoạt động hiện tại của xe[10].

Do hoạt động trong môi trường phức tạp, với các yếu tố ảnh hưởng đến quá trình quy hoạch chuyển động của phương tiện như: hệ thống đường giao thông, điều kiện thời tiết khác nhau, những hạn chế của hệ thống cảm biến, các tình huống bất thường do các đối tượng tham gia giao thông khác... [1–3, 9], đã tạo nên sự phức tạp thay đổi theo thời gian và các tình huống không thể dự đoán trước. Do đó, các thuật toán chỉ thuần túy về xác định khoảng cách an toàn và khoản thời gian an toàn chưa đủ linh hoạt để

thích ứng với đa số các trường hợp xảy ra [7, 12].

Một vài nghiên cứu trước đây cũng đã chỉ ra rằng, với nhiều đối tượng điều khiển phương tiện khác nhau thì khả năng phản ứng trước tình huống nguy hiểm cũng khác nhau. Như trong nghiên cứu [4], H. Zhou và các cộng sự đã chứng minh những tác động không hiệu quả của hệ thống cảnh báo khi thực hiện tín hiệu cảnh báo quá sớm đã gây ra những can thiệp xấu cho đối tượng là người điều khiển trẻ tuổi, vì với nhóm đối tượng này thì khả năng phản ứng với những hình huống nguy hiểm nhanh hơn các nhóm đối tượng lớn tuổi. Ngược lại, với nhóm đối tượng lớn tuổi thì khả năng phản ứng chậm, nếu tín hiệu cảnh báo phát ra với khoảng cách để tạo tín hiệu cảnh báo như nhóm đối tượng trẻ tuổi thì quá trình điều khiển chuyển động của nhóm đối tượng này càng trở nên nguy hiểm. Tương tự như vậy, trong các điều kiện thời tiết khác nhau, như ban ngày và ban đêm, hoặc điều kiện thời tiết có tầm nhìn khác nhau thì thời điểm đưa ra tín hiệu cảnh báo cũng không thể giống nhau.

Như vậy, khi đang điều khiển phương tiện, nếu đối tượng điều khiển phương tiện phát hiện ra vấn đề nguy hiểm cần phải thực hiện quá trình điều khiển tránh nguy hiểm thì không cần phải thu nhận tín hiệu từ hệ thống cảnh báo. Ngược lại, nếu đối tượng điều khiển không tìm thấy tình huống nguy hiểm kịp thời, thì thời gian cảnh báo tránh va chạm để đảm bảo an toàn do hệ thống cảnh báo nguy hiểm đưa ra là thời gian tối thiểu để đối tượng điều khiển phương tiện thực hiện các biện pháp nhằm đảm bảo an toàn, khoảng thời gian này bao gồm thời gian phản ứng của đối tượng điều khiển và thời gian cần thiết để điều khiển phương tiện [12], quá trình điều khiển phương tiện có thể giảm tốc độ bằng thao tác phanh hoặc thực hiện thao tác thay đổi quỹ đạo chuyển động bằng thao tác điều khiển lái.

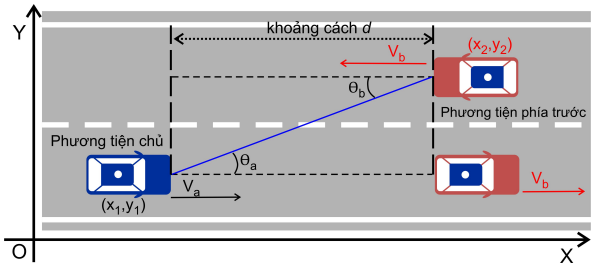
Trong bài báo này, dựa trên giải pháp cảnh báo tránh va chạm xác định khoảng thời gian an toàn, sẽ đề xuất một giải pháp cảnh báo tránh va chạm mới, trong đó kết hợp thêm yếu tố của dữ liệu môi trường xung quanh đã thu nhận được, cùng với đặc điểm điều khiển chuyển động của xe. So với các giải pháp trước đây, giải pháp đề xuất này có khả năng thích ứng và linh hoạt trong các điều kiện phức tạp của môi trường hoạt động, điều này có thể cải thiện được hiệu quả quy hoạch chuyển động và nâng cao tính an toàn của phương tiện. Kết quả mô phỏng được tiến hành trong môi trường Matlap để kiểm chứng giải pháp đưa ra đạt được hiệu quả hoạt động, với quá trình cảnh báo tránh va chạm trong quy hoạch chuyển động của phương tiện.

Phần tiếp theo của bài báo sẽ giới thiệu một số tình huống để làm nền tảng cho việc giải quyết bài toán cảnh báo tránh va chạm đặt ra, từ đó đề xuất mô hình giải pháp để thực hiện thao tác cảnh báo tránh va chạm cho phương tiện chủ dựa trên các yếu tố về thời gian, thông tin môi trường và điều kiện hoạt động của xe. Tiếp theo là phần thực nghiệm và kết luận với một số đề xuất các hướng nghiên cứu tiếp theo cho bài toán nâng cao tính năng an toàn của phương tiện.

II. XÂY DỰNG MÔ HÌNH CẢNH BÁO TRÁNH VA CHẠM

1. Phân tích bài toán

Hệ thống cảnh báo tránh va chạm đang là một trong những tiêu chuẩn được đưa vào danh sách đánh giá an toàn cho phương tiện. Những hệ thống cảnh báo tránh va chạm sử dụng các thiết bị cảm biến để thu nhận dữ liệu môi trường xung quanh, những dữ liệu này được phân tích để phát hiện ra các trường hợp nguy hiểm mà từ đó kích hoạt hệ thống phanh khẩn cấp hoặc đưa ra tín hiệu cảnh báo khẩn cấp nhằm có quyết định điều chỉnh hướng chuyển động phù hợp. Để giải quyết bài toán cảnh báo tránh va chạm, chúng ta sẽ phân tích tình huống có thể xảy ra va chạm trong quá trình hoạt động của phương tiện, gồm: tình huống va chạm trực diện hay va chạm phía trước và tình huống va chạm từ phía sau.



Hình 1. Minh họa tình huống tham gia giao thông

Với V_a là vận tốc của phương tiện chủ, V_b là vận tốc của phương tiện phía trước, θ_a là góc lệch hướng của phương tiện chủ và θ_b là góc lệch hướng của phương tiện phía trước, giá trị θ_a và θ_b được tính từ vị trí trung tâm của xe đến đường trung tâm giữa 2 xe (hình 1), W_a là chiều rộng của phương tiện chủ và W_b là chiều rộng của phương tiện phía trước, k là giá trị trung bình chiều rộng của 2 xe được tính bằng $k = (W_a + W_b)/2$. Vị trí của các phương tiện được xác định trong hệ trục tọa độ Descarter (X,Y), với (x_1, y_1) là vị trí của phương tiện chủ, (x_2, y_2) là vị trí của phương tiện phía trước, thì giá trị d là khoảng cách giữa 2 phương tiện được tính như sau:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Nếu 2 xe di chuyển ngược chiều, thì tình huống xảy ra va chạm trực diện khi:

$$k \leq |d * \sin\theta_a| \tag{2}$$

Và khoảng thời gian 2 xe sẽ xảy ra va chạm:

$$t = 4,3 * \frac{(d * |\cos\theta_a| - 5)}{V_a + V_b} \tag{3}$$

Khoảng thời gian xảy va chạm sẽ được tính toán khi $|\theta_a| \leq 90^0$ và $|\theta_b| > 90^0$. Trường hợp ngược lại khi $|\theta_a| \geq 90^0$ và $|\theta_b| \leq 90^0$ thì không tính giá trị này, vì không xảy ra va chạm. Nếu 2 xe di chuyển cùng chiều, thì trường hợp xảy ra va chạm từ phía sau khi vận tốc của phương tiện chủ lớn hơn vận tốc của phương tiện phía trước $V_a > V_b$. Khoảng thời gian 2 xe sẽ xảy ra va chạm:

$$t = \frac{d}{V_a + V_b} \tag{4}$$

Qua phân tích các tình huống nêu trên, có thể thấy rằng phương pháp tính toán thời gian xảy ra va chạm là như nhau cho dù là va chạm trực diện hay và chạm từ phía sau. Do đó, để giải quyết bài toán trong giải pháp này, chúng ta sẽ sử dụng đến giá trị thời gian va chạm nhằm nâng cao khả năng thích ứng của giải pháp trong môi trường phức tạp của quá trình quy hoạch chuyển động cho phương tiện khi tham gia giao thông.

2. Xây dựng thuật toán

Ý tưởng chính của giải pháp này là chọn ngưỡng khoảng cách hợp lý để thực hiện cảnh báo. Việc chọn ngưỡng khoảng cách để thực hiện cảnh báo là giải pháp được đánh giá đảm bảo hiệu quả an toàn, vì nếu khoảng cách cảnh báo quá dài thì sẽ gây tác động xấu và nhiễu thông tin cho hoạt động điều khiển phương tiện, nếu khoảng cách cảnh báo quá ngắn thì hiệu quả cảnh báo tránh va chạm không tối ưu và không thể tránh được nguy hiểm. Vì vậy, giải pháp đề xuất trong nghiên cứu này được lựa chọn là ngưỡng cảnh báo phù hợp, điều này không chỉ tạo ra tác động tốt và không nhiễu thông tin cho hoạt động điều khiển, mà còn giúp nâng cao hiệu quả an toàn khi điều khiển phương tiện. Hơn nữa, trong các điều kiện môi trường khác nhau, các giải pháp tránh va chạm nhằm đảm bảo an toàn không thể áp dụng cho nhiều đối tượng điều khiển phương tiện khác nhau. Vì thế, việc kết hợp thêm các thông số tác động của môi trường vào quá trình đưa ra giải pháp xây dựng mô hình cảnh báo tránh va chạm này là điều được chúng tôi xác định khá quan trọng và sẽ mang lại hiệu quả tối ưu.

Giải pháp này về cơ bản sẽ xác định theo thời gian phản ứng t của người điều khiển phương tiện. Dựa trên số liệu đã khảo sát trong nghiên cứu của tác giả M. A. Hoque và cộng sự [7], chúng tôi xác định 05 yếu tố có tác động tích cực đến quá trình điều khiển phương tiện và sẽ tính toán trọng số cho các yếu tố này trong quá trình thiết lập mô hình cảnh báo tránh va chạm của giải pháp đưa ra. Cụ thể, các yếu tố này bao gồm: Độ tuổi của người điều khiển X_1 với trọng số ω_1, thời gian kinh nghiệm điều khiển phương tiện X_2 với trọng số ω_2, chỉ số sức khỏe X_3 với trọng số ω_3, chỉ số tinh thần X_4 với trọng số ω_4 và chỉ số thị lực X_5 với trọng số ω_5.

Với các chỉ số lần lược có các giá trị như sau: $X_3 = (1,2,3,4,5)$; $X_4 = (1,2,3,4,5)$ và ma trận quyết định của từng yếu tố ảnh hưởng cùng thời gian phản ứng T như sau:

$$T = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_m \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \dots & \dots & \dots & \dots \\ X_{m1} & X_{m2} & \dots & X_{mn} \end{bmatrix} \tag{5}$$

và

$$P_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \tag{6}$$

$$E_j = -K \sum_{i=1}^{m} P_{ij} \ln(P_{ij}) \tag{7}$$

Trong đó: P_{ij} là tác động của thuộc tính thứ i vào yếu tố x_{ij}, $E_j \in [0,1]$ là tổng tác động của quá trình vào chỉ số X_j và hằng số $K = \frac{1}{\ln(m)}$.

Từ (5), (6) và (7) có thể thấy rằng vai trò tác động của mỗi lược đồ theo một thuộc tính nhất định có xu hướng giống nhau và giá trị E tiến tới 1. Trong trường hợp, khi tất cả yếu tố tác động đều bằng nhau thì vai trò thuộc tính mục tiêu trong quyết định có thể không xét đến, nghĩa là trọng số của chỉ số đang xét sẽ bằng 0. Từ đó, có thể thấy rằng giá trị thuộc tính được xác định bởi sự khác biệt của tất cả các lược đồ. Do đó, mức độ tác động nhất quán d_j của từng yếu tố trong thuộc tính thứ j được xác định như sau:

$$d_j = 1 - E_j \tag{8}$$

Khi đó, hệ số trọng số ω_j và khoảng thời gian đáp ứng τ_{dr} của hệ thống được tính như sau:

$$\omega_j = \frac{d_j}{\sum_{j=1}^{n} d_j} \tag{9}$$

$$Y_i = \sum_{i=1}^{5} X_i \omega_i \tag{10}$$

$$\tau_{dr} = 1.2\sqrt{\frac{75}{Y_i}} \tag{11}$$

Thời gian đáp ứng của hệ thống là thời gian thực hiện của hệ thống phanh, bao gồm: thời gian đáp ứng của hệ thống phanh t_1, thời gian tác động của phanh t_2 và thời

gian phanh liên tục t_3 [6]. Trong đó, thời gian phanh liên tục t_3 được giả định là giá trị không đổi để làm giảm tốc độ và được tính như sau:

$$t_3 = \frac{v}{3,6 * g * a * \mu} - \frac{t_2}{2} \quad (12)$$

Với v là vận tốc của xe chủ, $g = 9.8m/s^2$ là gia tốc trọng trường trái đất, μ là hệ số trượt (là hệ số bám giữa bánh xe và mặt đường) và a là chỉ số môi trường. Trong đó, các giá trị của chỉ số môi trường được đưa ra như sau: $a = 1$ khi mặt đường khô, $a = 0,5$ khi mặt đường có nước, $a = 0,1$ khi mặt đường cát và $a = 0,3$ khi mặt đường đồi núi.

Khoảng thời gian phanh liên tục sẽ thay đổi theo vận tốc của xe và hệ số trượt sẽ tương quan tỉ lệ nghịch với quãng đường phanh, do đó khoảng thời gian sẽ xảy ra va chạm TTC_{av} được tính như sau:

$$TTC_{av}(s) = \frac{D}{V_\tau} \quad (13)$$

Trong đó: D là khoảng cách giữa 2 xe và V_τ là vận tốc tương đối giữa 2 xe. Như vậy, ngưỡng thời gian an toàn TTC_{cs} để tránh va chạm được tính là khoảng thời gian tối thiểu từ khi hệ thống phát ra tín hiệu cảnh báo nguy hiểm đến thời điểm mà người điều khiển phương tiện thực hiện thao tác tránh nguy hiểm, được tính như sau:

$$TTC_{cs}(s) = t_1 + t_2 + t_3 + \tau_{dr} \quad (14)$$

Từ (10), (11), (12) và (14), ngưỡng thời gian an toàn TTC_{cs} được tính như sau:

$$TTC_{cs}(s) = t_1 + \frac{t_2}{2} + \frac{v}{3,6 * g * a * \mu} + 1.2\sqrt{\frac{75}{\sum_{i=1}^{5} X_i \omega_i}} \quad (15)$$

Trong đó: X_i được xác định tùy vào giá trị thực tế của người điều khiển phương tiện. Vì dữ liệu thu nhận được từ môi trường gồm tốc độ và khoảng cách giữa các xe là tương đối, nên chúng ta cần chuyển đổi ngưỡng thời gian an toàn TTC_{cs} thành khoảng cách xe tương ứng và chuẩn hóa các biến số. Điều này có thể làm giảm độ phức tạp về thời gian và không gian của thuật toán.

Từ đó, giá trị khoảng cách cảnh báo D_w và khoảng cách phanh xe D_b được tính như sau:

$$D_w = v_\tau(t_1 + \frac{t_2}{2} + \frac{v}{3,6 * g * a * \mu} + 1.2\sqrt{\frac{75}{\sum_{i=1}^{5} X_i \omega_i}}) \quad (16)$$

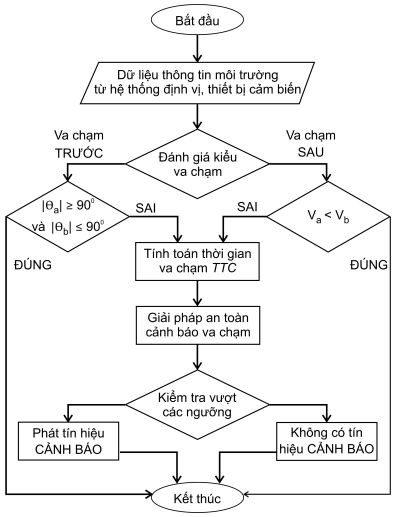
$$D_b = v_\tau(\frac{t_2}{2} + \frac{v}{3,6 * g * a * \mu}) \quad (17)$$

Với V_τ là vận tốc tương đối giữa 2 xe, t_1 là thời gian đáp ứng của hệ thống phanh, t_2 là thời gian tác động của phanh và t_3 là thời gian phanh liên tục, $g = 9,8m/s^2$ là gia tốc trọng trường trái đất, μ là hệ số trượt, $a \in [1,0; 5,0; 1,0; 3]$ là chỉ số môi trường, X_i và ω_i là chỉ số tác động và trọng số tương ứng. Như vậy, với việc chuyển đổi ngưỡng thời gian an toàn để tránh va chạm thành giá trị khoảng cách ngưỡng cảnh báo tương ứng và đưa ra tín hiệu cảnh báo cần thiết trong quá trình điều khiển chuyển động để tránh va chạm. Giải pháp cảnh báo tránh va chạm cho phương tiện tham gia giao thông được chúng tôi đưa ra trong nghiên cứu này sẽ được thực thi với các dữ liệu đầu vào là tốc độ thực tế, đặc điểm môi trường và đặc điểm người điều khiển phương tiện, với các bước cơ bản được trình bày trong thuật toán 1.

Thuật toán 1: Cảnh báo tránh va chạm.

Dữ liệu vào: tập các chỉ số.
Dữ liệu ra: Ngưỡng khoảng cách cảnh báo, khoảng cách phanh xe.
begin
  Xác định trọng số của từng chỉ số( )→ ω_i;
  Tính khoảng thời gian đáp ứng của hệ thống→ τ_{dr};
  Tổng hợp thời gian phanh của từng giai đoạn $(D, V_\tau) \rightarrow TTC_{av}$;
  Tính ngưỡng khoảng cách cảnh báo $(., v_\tau) \rightarrow D_w$;
  Tính ngưỡng khoảng cách phanh xe $(., v_\tau) \rightarrow D_b$;
  Đưa ra quyết định phát tín hiệu cảnh báo ( );
end

Biểu diễn bằng lưu đồ thuật toán:



Hình 2. Lưu đồ thuật toán cảnh báo tránh va chạm

III. ĐÁNH GIÁ VÀ KẾT LUẬN

Để kiểm tra và đánh giá giải pháp đề xuất, chúng tôi tiến hành mô phỏng thực nghiệm các quy trình trong môi trường Matlap. Đồng thời, nhằm đảm bảo tính khách quan và độ tin cậy khi đánh giá, chúng tôi tiến hành mô phỏng với các kịch bản khác nhau, gồm kịch bản xảy ra va chạm trước và kịch bản xảy ra va chạm phía sau. Trong đó các tính toán sử

dụng hệ đo lường SI với khoảng thời gian cập nhật của hệ thống $\tau = 0,2s$, các bộ thu nhận tín chuyển dữ liệu đến bộ điều khiển I/O với chu kỳ thời gian mô phỏng là $0,05s$. Quá trình mô phỏng để xác định giá trị khoảng cách cảnh báo an toàn được tiến hành với 03 tình huống: Tình huống thực hiện phanh khẩn cấp với phương tiện phía trước, tình huống vận tốc của xe phía trước nhỏ hơn xe chủ và tình huống cả 2 xe đều cùng vận tốc. Tham số trong mô phỏng này với vận tốc và gia tốc của các xe được quy định là bằng nhau, do đó giá trị vận tốc tương đối v_τ của các xe có 1 giá trị trong môi trường hoạt động và thời gian đáp ứng của hệ thống phanh $t_1 = 0,5s$, thời gian tác động của phanh $t_2 = 0,4s$, thời gian phanh liên tục $t_3 = 3s$, hệ số trượt $\mu = 0,7$, chỉ số đối tượng điều khiển phương tiện tạo tác động cho hệ thống được tiến hành 02 người, với các chỉ số như sau: $[OBJ]_a = (X_1 = 30, X_2 = 5, X_3 = 7, X_4 = 1,8, X_5 = 5)$ và $[OBJ]_b = (X_1 = 58, X_2 = 3, X_3 = 4, X_4 = 1,1, X_5 = 20)$.

Kết quả mô phỏng tiếp tục được đánh giá so sánh với các giải pháp áp dụng thuật toán khác gồm: thuật toán Mazda[13] và thuật toán Berkeley[13], quá trình đánh giá được tiến hành bằng cách so sánh khoảng cách thực hiện phanh an toàn với các vận tốc tương đối khác nhau.

Trong đó, thuật toán Mazda là thuật toán dựa trên phân tích động học để xác định khoảng cách giới hạn phanh. Tuy nhiên, khoảng cách được tính toán là khoảng cách để đảm bảo an toàn cho thao tác phanh, nên khoảng cách này được sử dụng làm khoảng cách giới hạn cảnh báo [11]. Khoảng cách cảnh báo D_w được tính như sau:

$$D_{wmazda} = \frac{1}{2}\left(\frac{v^2}{a_1} - \frac{(v - v_0)^2}{a_2}\right) + vt_1 + v_0t_2 + d_0 \quad (18)$$

Với d_0 là khoảng cách ban đầu, v là vận tốc của xe phía sau, v_0 là vận tốc tương đối giữa các xe, t_1 là thời gian trễ của hệ thống, t_2 là thời gian phản ứng của người điều khiển, a_1 là vận độ giảm cực đại do thao tác phanh của xe phía sau và a_2 là vận độ giảm cực đại do thao tác phanh của xe phía trước.

Và thuật toán Berkeley là là thuật toán được cải tiến dựa trên thuật toán Mazda nhằm tạo ra khoảng cách cảnh báo an toàn hơn, phương trình khoảng cách cảnh báo $D_{wberkeley}$ được xây dựng như sau:

$$D_{wmazda} = \frac{1}{2}\left(\frac{v^2}{a} - \frac{(v - v_0)^2}{a}\right) + v(t_1 + t_2) + d_0 \quad (19)$$

Với d_0 là khoảng cách ban đầu, v là vận tốc của xe phía sau, v_0 là vận tốc tương đối giữa các xe, t_1 là thời gian trễ của hệ thống, t_2 là thời gian phản ứng của người điều khiển, a là vận độ giảm cực đại do thao tác phanh của xe.

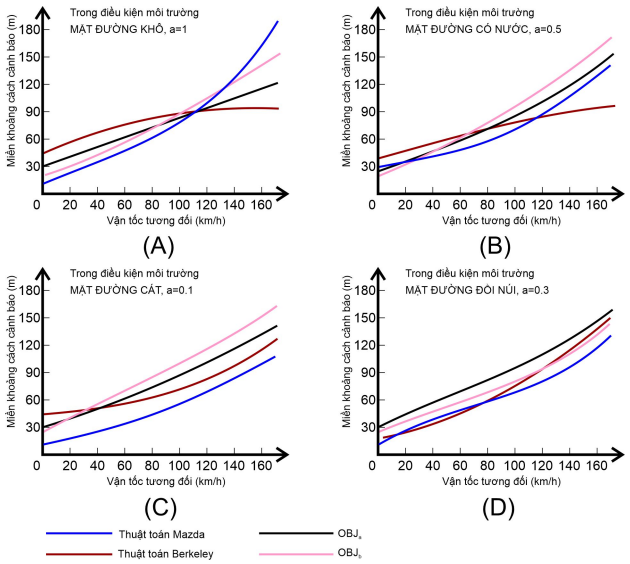
$$\xi = \frac{d - d_\xi}{d_\xi - d_{br}} \quad (20)$$

$$d_{br} = v_0(t_1 + t_2) + \frac{1}{2}a(t_1 + t_2)^2 \quad (21)$$

Với d là khoảng cách hiện tại giữa 2 xe, d_{br} là khoảng cách giới hạn phanh.

1. Kết quả mô phỏng cảnh báo va chạm phía sau

Quá trình thực hiện mô phỏng trong các điều kiện môi trường khác nhau, với 04 chỉ số môi trường được thiết lập cho dữ liệu đầu vào gồm $a \in [1,0,5,0,1,0,3]$ tương ứng với môi trường mặt đường khô, mặt đường có nước, mặt đường cát và mặt đường đồi núi.

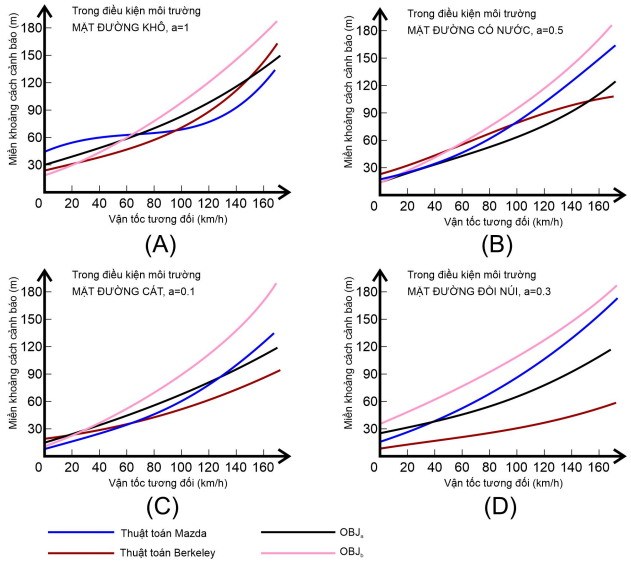


Hình 3. Kết quả mô phỏng cảnh báo va chạm sau

Kết quả số liệu mô phỏng cho thấy với tốc độ di chuyển khác nhau thì trong điều kiện môi trường tốt (mặt đường khô) khoảng cách cảnh báo của OBJ_a (có chỉ số độ tuổi ít) ngắn hơn so với OBJ_b (có chỉ số độ tuổi nhiều), nghĩa là người trẻ tuổi sẽ có phản xạ nhanh hơn khi thay đổi tốc độ so với người lớn tuổi trong điều kiện bình thường và người có chỉ số tuổi lớn thì thao tác thận trọng hơn. Nhưng trong điều kiện mặt đường có nước, mặt đường cát thì khoảng cách cảnh báo của OBJ_a tăng lên rất nhiều và cũng tương đương với khoảng cách cảnh báo của OBJ_b, nghĩa là với chỉ số kinh nghiệm điều khiển còn ít thì khoảng cách dự báo cần thay đổi cho phù hợp với điều kiện thời tiết không thuận lợi. Như vậy, qua kết quả mô phỏng chúng ta có thể thấy được tác động của điều kiện môi trường làm thay đổi quá trình phát tín hiệu cảnh báo tránh va chạm, và tùy thuộc tập chỉ số đầu vào của đối tượng điều khiển phương tiện mà khoảng cách cảnh báo cũng thay đổi tương ứng.

2. Kết quả mô phỏng cảnh báo va chạm trực diện

Quá trình thực hiện mô phỏng trong các điều kiện môi trường khác nhau, với 04 chỉ số môi trường được thiết lập cho dữ liệu đầu vào gồm $a \in [1,0,5,0,1,0,3]$.



Hình 4. Kết quả mô phỏng cảnh báo va chạm trực diện

Kết quả mô phỏng cho thấy, khoảng cách cảnh báo tránh va chạm của giải pháp áp dụng thuật toán Berkeley ít thay đổi theo vận tốc tương đối. Ngoài ra, khi vận tốc tương đối cao thì khoảng cách cảnh báo ngắn nên không kịp thời và khi tốc độ tương đối thấp dễ gây ra cảnh báo sai, từ đó dẫn đến điều kiện an toàn trong cảnh báo nguy hiểm không đảm bảo. Trong mô phỏng tránh va chạm trực diện này thì khoảng cách cảnh báo khi áp dụng giải pháp đề xuất với các chỉ số đầu vào của OBJ_a, OBJ_b thì khoảng cách cảnh báo tránh va chạm nhạy cảm hơn với sự thay đổi tốc độ tương đối, nên có thể đảm bảo an toàn cho các phương tiện với các tốc độ tương đối khác nhau. Trong điều kiện các chỉ số môi trường hoạt động khác nhau, với vận tốc tương đối chậm thì khoảng cách dự báo của giải pháp đề xuất gần bằng với khoảng cách dự báo được thực hiện bởi thuật toán Berkeley, nhưng khi vận tốc tương đối tăng thì khoảng cách dự báo giữa 2 giải pháp cũng tăng tỷ lệ thuận và mức độ gia tăng khoảng cách cảnh báo của giải pháp đề xuất tăng nhanh hơn so với giải pháp áp dụng thuật toán Berkeley.

Còn với giải pháp áp dụng thuật toán Mazda có thời gian cảnh báo tương đối chậm trong các điều kiện môi trường khác nhau, khoảng cách cảnh báo ngắn và chậm, điều này vẫn chưa chiếm được ưu điểm so với giải pháp được đề xuất trong nghiên cứu này.

3. Kết luận

Như vậy với giải pháp kết hợp giữa thời gian tránh va chạm và các chỉ số liên quan về môi trường hoạt động, chỉ số liên quan đến đối tượng điều khiển phương tiện, giải pháp đề xuất đã giải quyết bài toán cảnh báo tránh va chạm dưới góc độ xét các mối quan hệ giữa vị trí các đối tượng, các yếu tố tác động của môi trường như hệ số trượt, tầm nhìn để tạo ra các khoảng thời gian cảnh báo khác nhau, với kết quả mô phỏng ban đầu, đã cho thấy khả năng thích nghi của giải pháp đề xuất có thể áp dụng với nhiều chỉ số đầu vào tùy thuộc vào môi trường hoạt động, đã giảm tỷ lệ cảnh báo sai và cảnh báo thiếu so với các giải pháp áp dụng thuật toán Berkeley và Mazda. Thuật toán đề xuất đã chứng minh tính hợp lý, khả thi và có thể cải thiện hiện quả an toàn giao thông thông qua thực nghiệm.

Ý tưởng chính của giải pháp kỹ thuật này sẽ hỗ trợ cho việc thiết kế xe một điểm dừng an toàn, cho dù việc điều khiển xe hiện tại là như thế nào. Trong tương lai, để tăng độ tin cậy của giải pháp này thì những thiết lập đã được thực nghiệm bằng mô phỏng sẽ được chuyển sang môi trường thực với xe thực nghiệm được trang bị đầy đủ các cảm biến, đồng thời khi thực nghiệm trên thực tế sẽ bổ sung một số yếu tố phân tích tính ổn định của hệ thống sao cho hành vi tham gia giao thông của các đối tượng được dự báo chính xác hơn. Việc triển khai rộng rãi giải pháp này cho các xe bán tự hành trong các hệ thống điều khiển phương tiện sẽ có thể giảm thiểu được số lượng lớn các thiệt hại cũng như tạo ra một kế hoạch chuyển động an toàn cho tương lai.

TÀI LIỆU THAM KHẢO

[1] A. Thakur and R. Malekian (2019),“Internet of vehicles communication technologies for traffic management and road safety applications” *Wireless Personal Communications*, vol. 109, no. 1, pp. 31–49, DOI:10.1007/s11277-019-06548-y.

[2] E. Awad, S. Dsouza, R. Kim et al. (2018), “The moral machine experiment”, *Nature*, vol. 563, no. 7729, pp. 59–64, DOI: 10.1038/s41586-018-0637-6.

[3] H. Peng, L. Le Liang, X. Shen, et al. (2019), “Vehicular communications: a network layer perspective”, *IEEE Transactions On Vehicular Technology*, vol. 68, no. 2, pp. 1064–1078, DOI: 10.1109/TVT.2017.2750903.

[4] H. Zhou, N. Cheng, J. Wang et al (2019), “Toward dynamic link utilization for efficient ve-hicular edge content distribution”, *IEEE Transactions On Vehicular Technology*,vol. 68, no. 9, pp. 8301–8313, DOI: 10.1109/TVT.2019.2921444.

[5] L. Yang, J. Wang, C. Gao et al (2018), “A crisis information propagation model based on a competitive relation”, *Journal of Ambient Intelligence and Humanized Computing*,, vol. 10, no. 8, pp. 2999–3009, DOI: 10.1007/s12652-018-0744-0.

[6] L. Jin, B. Arem, S. Yang (2019), “Performance analysis and boost for a MAC protocol in vehicular networks”, *IEEE Transactions on Vehicular Technology*,vol. 68, no. 9, pp. 8721–8728, DOI: 10.1109/TVT.2019.2927228.

[7] M. A. Hoque, X. Hong, M. S. Ahmed (2019), “Parallel closed-loop connected vehicle simu-lator for large-scale transportation network management: challenges, issues, and solution ap-proaches”, *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 62–77, DOI: 10.1109/MITS.2018.2879163.

[8] N. Jiang, F. Tian, J. Li et al. (2019), “MAN: mutual attention neural networks model for as-pect-level sentiment classification in SIot”, *IEEE Internet of Things Journal*,vol. 15, no. 3, pp. 1054–1065, DOI: 10.1109/JIOT.2020.2963927.

[9] R. Molina-Masegosa, J. Gozalvez (2017), "LTE-V for sidelink 5G V2X vehicular commu-nications: a new 5G technology for short-range vehicle-to-everything communi-cations", *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, DOI: 10.1109/MVT.2017.2752798.

[10] S. Sharma, B. Kaushik (2019), "A survey on internet of vehicles: applications, security is-sues and solutions" , *Vehicular Communications*,vol. 20, no. 5, pp. 725–729, DOI: 10.1016/j.vehcom.2019.100182.

[11] S. Jeong, O. Simeone, J. Kang (2018), "Mobile edge computing via a UAV mounted cloud let: optimization of bit allocation and path planning", *IEEE Transactions On Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, DOI: 10.1109/TVT.2017.2706308.

[12] X. Pei (2012), "Safe distance model and obstacle detection algorithms for A collision warn-ing and collision avoidance system", *Automotive Safety and Energy*, vol. 3, no. 1, pp. 26–33, DOI: 10.3969/j.issn.1674-8484.2012.01.004.

[13] Ararat, Oncu and Kural, Emre and Guvenc, Bilin Aksum."Development of a collision warning system for adaptive cruise control vehicles using a comparison analysis of recent algorithms", *2006 IEEE Intelligent Vehicles Symposium*, pp. 194-199, 2006

SƠ LƯỢC VỀ TÁC GIẢ

Quách Hải Thọ

Sinh năm 1978, tốt nghiệp cử nhân ngành tin học (2000) và thạc sĩ chuyên ngành Khoa học máy tính (2012) tại trường Đại học Khoa học, Đại học Huế. Hiện đang làm NCS ngành Hệ thống thông tin tại trường Đại học Sư phạm, Đại học Đà nẵng. Đơn vị công tác: Phòng Đào tạo, BĐCL& CTSV, trường Đại học Nghệ thuật, Đại học Huế. Lĩnh vực nghiên cứu: Đồ họa máy tính, học máy.
Email: qhaitho@hueuni.edu.vn
Tel: 0913439816

Huỳnh Công Pháp

Sinh năm 1977, đã nhận bằng Tiến sĩ (2010) ngành Công nghệ thông tin tại Đại học Bách Khoa Grenoble–CH Pháp & Viện CNTT Quốc gia NII-Tokyo-Nhật. Hiện là Hiệu trưởng trường Đại học Công nghệ thông tin và Truyền thông Việt - Hàn, Đại học Đà Nẵng. Lĩnh vực nghiên cứu: Mạng máy tính, Lập trình mạng, Web ngữ nghĩa.
Email: hcphap@vku.udu.vn

Phạm Anh Phương

Sinh năm 1974, đã tốt nghiệp đại học (1996) ngành Toán - Tin học tại trường Đại học Sư phạm, Đại học Huế, tốt nghiệp Thạc sĩ (2001) chuyên ngành Khoa học Máy tính tại Đại học Bách Khoa Hà Nội và nhận bằng Tiến sĩ (2010) chuyên ngành - Bảo đảm toán học cho máy tính và hệ thống tính toán tại Viện Công nghệ Thông tin, Hà Nội. Hiện là Phó Trưởng Khoa - Khoa Tin học, Trường ĐH Sư Phạm - Đại học Đà Nẵng. Lĩnh vực nghiên cứu: Kỹ thuật lập trình; Đồ hoạ máy tính, Xử lý ảnh, lý thuyết nhận dạng.
Email: paphuong@ued.udn.vn

Tiếp cận Heuristic giải bài toán Clique lớn nhất trên đơn đồ thị vô hướng không có trọng số

Phan Thành Huấn[1], Huỳnh Thị Châu Ái[2], Châu Lê Sa Lin[3]

[1] Đại học Quốc gia Tp. Hồ Chí Minh

[2] Khoa Kỹ thuật Công nghệ, Trường Đại học Văn Hiến

[3] Khoa Công nghệ Thông tin - truyền thông, Trường Cao đẳng Kinh tế - Kỹ thuật Cần Thơ

Tác giả liên hệ: Phan Thành huấn, huanphan@hcmussh.edu.vn

Ngày nhận bài: 15/04/2021, ngày sửa chữa: 01/06/2021, ngày duyệt đăng: 12/06/2021

Định danh DOI: 10.32913/mic-ict-research.v2021.n1.964

Tóm tắt: Bài toán Clique lớn nhất (Maximum Clique Problem) là bài toán tìm tập con lớn nhất của tập đỉnh trong đơn đồ thị vô hướng, sao cho hai đỉnh phân biệt trong nó luôn kề nhau. Đây là bài toán nổi tiếng thuộc lớp NP-complete, được ứng dụng nhiều trong các lĩnh vực khai thác dữ liệu, phân tích mạng, truy xuất thông tin, y học, giáo dục và nhiều lĩnh vực khác liên quan đến mạng lưới toàn cầu. Có nhiều cách tiếp cận giải bài toán Clique lớn nhất như quy hoạch động, nhánh-cận, heuristic hay meta-heuristic – cho lời giải chính xác hay xấp xỉ. Trong bài báo này, nhóm tác giả phân tích hai thuật giải tiếp cận heuristic gần đây và đề xuất các heuristic tăng độ chính xác của lời giải cho bài toán Clique lớn nhất. Phần thực nghiệm, nhóm tác giả so sánh chất lượng lời giải của thuật giải đề xuất trên 10 bộ dữ liệu từ DIMACS.

Từ khóa: *clique lớn nhất, tiếp cận Heuristic, NP-complete.*

Title: **Heuristic Approach for Solving the Maximum Clique Problem on Simple Undirected and Unweighted Graph**

Abstract: The Maximum Clique Problem is the problem of finding the largest subset of vertices in a simple undirected graph, such that two distinct vertices in it are always adjacent. This is a well-known NP-complete problem, widely applied in the fields of data mining, network analysis, information retrieval, medicine, education and many other fields related to World Wide Web. There are many approaches to solving the Maximum Clique problem such as dynamic programming, branch and cut, heuristic or metaheuristic – for exact or approximate solutions. This article, the authors analyze two recent heuristic approaches and propose heuristics to increase the accuracy of the solution for the Maximum Clique problem. The experimental section, the authors compare the solution quality of the proposed algorithm on 10 datasets from the DIMACS.

Keywords: *maximum clique, Heuristic, NP-complete.*

I. GIỚI THIỆU

Bài toán clique lớn nhất (Maximum Clique Problem – MCP) là một bài toán kinh điển của lý thuyết đồ thị thuộc lớp NP-complete và có nhiều ứng dụng lĩnh vực khoa học máy tính, sinh học, tài chính,.. được giải quyết thông qua bài toán tìm kiếm các Clique trong đồ thị. Trong ứng dụng thực tế, các bài toán thường được mô hình hoá bằng đồ thị rất lớn và cần phải có bộ nhớ lưu trữ đủ lớn cho quá trình thực hiện các thuật toán. Nhóm tác giả Đàm Thanh Phương và đồng sự [16] đã thảo luận những thách thức tính toán từ lý thuyết đến thực hành của bốn ứng dụng cụ thể - cho thấy khó khăn trong thiết kế các thuật toán phù hợp mang hiệu quả cao.

Trong vài năm trở lại đây, có nhiều công trình nghiên cứu về MCP và các ứng dụng không liên quan đến mạng lưới toàn cầu như: xác định các cấu trúc protein tương đồng [2] – sinh học; theo dõi đa mục tiêu [9] – thị giác máy tính; phân cụm điện não đồ [12] – y học; xếp hạng các trường đại học [13] – giáo dục;... Nhóm tác giả dựa vào công trình khảo sát [10], có thể phân loại các nghiên cứu giải bài toán MCP theo hai nhóm chiến lược tiếp cận:

Chiến lược tìm kiếm lời giải chính xác: Thuật toán quy hoạch động [1], thuật toán nhánh-cận [2, 4, 8]. Chiến lược này chỉ thích hợp thực hiện trên các đồ thị có kích thước nhỏ và cho thời gian thực thi là cao. Trong kỷ nguyên bùng nổ dữ liệu của các lĩnh vực – Đây thực sự là thách thức lớn

trong lĩnh vực nghiên cứu lý thuyết tối ưu tổ hợp. Chiến lược này còn là cơ sở cho đánh giá độ chính xác của các giải thuật cho lời giải gần đúng.

Chiến lược tìm kiếm lời giải gần đúng: Đây là giải pháp lựa chọn cho các bài toán có thời gian thực thi là cao và có thể chấp nhận lời giải gần đúng; chiến lược này thường dựa vào kinh nghiệm để đưa ra các heuristic cho từng bài toán hoặc dữ liệu cụ thể – không chắc hiệu quả cho tất cả; gọi là các giải thuật heuristic. Giải thuật heuristic làm giảm đáng kể thời gian thực thi – ưu điểm nổi bật; một số giải thuật heuristic giải bài toán MCP: [3, 5, 14, 18, 19], v.v.. Ngoài ra, chiến lược này còn sử dụng kết hợp nhiều ưu điểm của từng giải thuật heuristic – gọi là meta-heuristic. Gần đây cũng có nhiều nghiên cứu sử dụng các giải thuật meta-heuristic giải bài toán MCP – giải thuật bầy ong [11], giải thuật tối ưu đàn kiến NACO [6], KLS_improve [17], v.v..

Năm 2019, nhóm tác giả Phan Tấn Quốc và Huỳnh Thị Châu Ái đã đề xuất heuristic cải tiến thuật giải MAXCLIQUEHEU1 [15] và MAXCLIQUEHEU2 [5] cho lời giải chất lượng chưa tốt ở một số bộ dữ liệu từ hệ thống DIMACS (10 bộ dữ liệu ở Bảng 2). Vì vậy, nhóm tác giả chọn nghiên cứu đề xuất các heuristic hiệu quả cho việc gia tăng chất lượng của lời giải.

Trong nghiên cứu này, nhóm tác giả tập trung phân tích hai giải thuật heuristic [18] và đề xuất các heuristic hiệu quả làm gia tăng chất lượng lời giải cho bài toán MCP dựa trên các tính toán thống kê phù hợp. Trong Phần 2, trình bày các khái niệm cơ bản về bài toán Clique lớn nhất và phân tích một số thuật giải gần đây. Phần 3, trình bày thuật giải đề xuất MCP-HEU* xác định nhanh Clique lớn nhất của đơn đồ thị vô hướng không có trọng số. Kết quả thực nghiệm được trình bày trong Phần 4 và kết luận ở Phần 5.

II. KHẢO SÁT MỘT SỐ HEURISTIC GIẢI BÀI TOÁN CLIQUE LỚN NHẤT

Trong phần này, nhóm tác giả trình bày các khái niệm liên quan bài toán Clique lớn nhất và đồng thời phân tích 2 giải thuật gần đây của nhóm tác giả Phan Tấn Quốc và Huỳnh Thị Châu Ái [18, 19].

1. Bài toán Clique lớn nhất

Cho G = (V, E) là đơn đồ thị vô hướng không có trọng số, có V là tập chứa các đỉnh của đồ thị, E là tập chứa các cạnh trong đồ thị. Tập các đỉnh liền kề với đỉnh vi, ký hiệu adj(v_i); bậc của đỉnh v_i là lực lượng của adj(v_i), ký hiệu deg(v_i) và deg(v_i) = |adj(v_i)|.

Định nghĩa 1: (Complete Graph) Đồ thị đầy đủ n đỉnh là đơn đồ thị vô hướng mà giữa 2 đỉnh bất kỳ trên đồ thị luôn có cạnh nối, ký hiệu K_n.

Định nghĩa 2: (Clique) $\forall K_{n\geq 1} \subseteq$ G = (V, E), các tập đỉnh của đồ thị $K_{n>1}$ được gọi là các Clique của đồ thị G. Ký hiệu, C_G là một Clique của đồ thị G.

Tính chất 1: Số lượng đỉnh (kích thước) của một Clique C của đồ thị G, ký hiệu là $|C|$. Nếu $v_i \in |C|$ thì $|C| < deg_G(v_i) + 1$.

Định nghĩa 3: (Maximal Clique - Clique cực đại) C_G là một Clique của đồ thị G được gọi là một Clique cực đại, nếu không tồn tại $C'_G \supset C_G$ và $|C'_G| \geq |C_G|$. Ký hiệu, $C_{Maximal}$ là Clique cực đại của đồ thị G.

Định nghĩa 4: (Maximum Clique - Clique lớn nhất) C_G là một Clique của đồ thị G được gọi là một Clique lớn nhất, nếu không tồn tại $C'_G \supset C_G$ và $|C_G| \geq |C'_G|$. Ký hiệu, $C_{Maximum}$ là Clique lớn nhất của đồ thị G.

Định nghĩa 5: (Clique number - Chỉ số Clique) là số lượng đỉnh của $C_{Maximum}$ Clique lớn nhất của đồ thị G. Chỉ số Clique của đồ thị G, ký hiệu là ω(G).

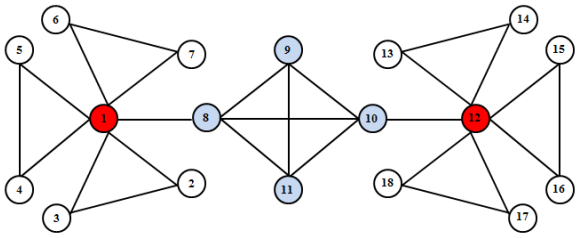
Tính chất 2: (dựa vào định nghĩa 3 và 4) Một Clique lớn nhất là một Clique cực đại, nhưng một Clique cực đại chưa chắc đã là một Clique lớn nhất.

Tính chất 3: (dựa vào định nghĩa 4) Đồ thị có thể có nhiều Clique có kích thước bằng Clique lớn nhất.

Bài toán tìm Clique lớn nhất

Cho G = (V, E) là đơn đồ thị vô hướng không có trọng số. Bài toán tìm Clique lớn nhất (Maximum Clique Problem - MCP) là bài toán tìm một Clique lớn nhất (theo định nghĩa 4) trong đồ thị đã cho.

Cho G là đơn đồ thị vô hướng không có trọng số, có 18 đỉnh và 26 cạnh như ở Hình 1.



Hình 1. Đơn đồ thị vô hướng G cho ví dụ

Bảng I, cho thấy đồ thị G có số lượng phần tử của tập Clique, Clique cực đại và Clique lớn nhất lần lượt là 37, 9 và 1 – số lượng các Clique rất lớn so với Clique cực đại và Clique lớn nhất. Đồng thời cho thấy Clique lớn nhất của đồ thị là **{8, 9, 10, 11}** và chỉ số Clique $\omega(G) = 4$.

2. Một số giải thuật giải bài toán Clique lớn nhất

Năm 2019, nhóm tác giả Phan Tấn Quốc cùng đồng sự đã cải tiến thuật toán MAXCLIQUEHEU1 [15] của nhóm tác giả Vũ Đình Hòa và MAXCLIQEHEU2 [5] của

Bảng I
TẬP CLIQUE, CLIQUE CỰC ĐẠI VÀ CLIQUE LỚN NHẤT TRÊN ĐỒ THI G.

Tập Clique (#C=37)	**Tập Clique cực đại (#$C_{Maximal}$=9)**	**Tập Clique lớn nhất (#$C_{Maximum}$=1)**
{1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {1, 7}, **{1, 8}**, {2, 3}, {4, 5}, {6, 7}, {8, 9}, {8, 10}, {8, 11}, {9, 10}, {9, 11},{10, 11}, **{10, 12}**, {12, 13}, {12, 14}, {12, 15}, {12, 16}, {12, 17}, {12, 18}, {13, 14}, {15, 16}, {17, 18}, **{1, 2, 3}**, **{1, 4, 5}**, **{1, 6, 7}**, {8, 9, 10}, {8, 9, 11}, {8, 10, 11}, {9, 10, 11}, **{12, 13, 14}**, **{12, 15, 16}**, **{12, 17, 18}**, **{8, 9, 10, 11}**	{1, 8}, {10, 12},{1, 2, 3}, {1, 4, 5}, {1, 6, 7}, {12, 13, 14}, {12, 15, 16}, {12, 17, 18}, **{8, 9, 10, 11}**	{8, 9, 10, 11}

Algorithm 1: HEU1 - xác định kích thước Clique lớn nhất của đồ thị.

```
1  Inputs: Đồ thị G = (V, E).
2  Outputs: Kích thước Clique lớn nhất - ω_G.
3  begin
4      ω_Best = 0;
5      C_Maximum = {∅};
6      while điều kiện dừng chưa thỏa do
7          U = {∅};
8          while V ≠ {∅} do
9              S = {k đỉnh thuộc V có bậc lớn nhất};
10             Chọn ngẫu nhiên đỉnh v ∈ S ;
11             U = U ∪ {v};
12             V = V − {v};
13             Xóa các đỉnh không kề với đỉnh v ra khỏi V;
14             Cập nhật bậc các đỉnh trong V
15         end
16         if |U| ≥ ω_Best then
17             ω_Best = |U|;
18             Cập nhật C_Maximum = U;
19         end
20     end
21     return ω_Best;
22 end
```

Algorithm 2: HEU2 - xác định kích thước Clique lớn nhất của đồ thị.

```
1  Inputs: Đồ thị G = (V, E).
2  Outputs: Kích thước Clique lớn nhất - ω_G.
3  begin
4      ω_Best = 0;
5      C_Maximum = {∅};
6      while điều kiện dừng chưa thỏa do
7          Chọn ngẫu nhiên đỉnh v ∈ S;
8          if deg(v_i) ≥ max then
9              U = {∅};
10             foreach v_j ∈ adj(v_i) do
11                 if deg(v_j) ≥ max then
12                     U = U ∪ {v_j};
13                 end
14                 while V ≠ {∅} do
15                     S = {k đỉnh V có bậc lớn nhất từ
                           V};
16                     Chọn ngẫu nhiên đỉnh v ∈ S ;
17                     U = U - {v};
18                     Chỉ giữ lại đỉnh u ∈ U: (u kề với
                           đỉnh v và deg(u) ≥ max);
19                     max = max + 1;
20                 end
21             end
22             if |U| ≥ max then
23                 max = |U|;
24             end
25         end
26         if max ≥ ω_Best then
27             ω_Best = max;
28             Cập nhật C_Maximum = U;
29         end
30     end
31     return ω_Best;
32 end
```

nhóm tác giả Pattabiraman cùng đồng sự; thuật toán cải tiến lần lượt có tên gọi là MAXCLIQUEHEU1_improve và MAXCLIQUEHEU2_improve [18] – để thuận tiện trong việc phân tích, chúng tôi sử dụng tên tắt cho 2 thuật toán cải tiến trên là HEU1 và HEU2. Các giải thuật cải tiến HEU1, HEU2 được trình bày:

Phần mã giả thuật giải HEU1 cải tiến từ MAXCLIQUE-HEU1 [15] - Thuật giải 1

Trong giải thuật HEU1, nhóm tác giả đã bổ sung dòng 9 và 10 – chọn k đỉnh thuộc V có bậc lớn nhất (dòng 9), sử dụng chiến lược chọn ngẫu nhiên (dòng 10) và giải thuật được lặp lại nhiều lần (điều kiện dừng – dòng 6) cho việc chọn lời giải tốt hơn.

Ưu điểm: Dễ hiểu, đơn giản khi sử dụng chiến lược tham lam – "đỉnh có bậc lớn nhất thì có nhiều khả năng là đỉnh thuộc Clique lớn nhất";

Nhược điểm: Chọn tập S gồm k đỉnh có bậc lớn nhất – cần xác định k là bao nhiêu cho phù hợp từng bộ dữ liệu, điều này cho ta thấy chất lượng của lời giải phụ thuộc vào tập đỉnh có bậc lớn nhất được lựa chọn.

Phần mã giả thuật giải HEU2 cải tiến từ MAXCLIQUE-HEU2 [5] - Thuật giải 2

Tương tự, ở giải thuật HEU2, nhóm tác giả đã bổ sung dòng 16 – chọn k đỉnh thuộc U có bậc lớn nhất và chiến lược chọn ngẫu nhiên (dòng 16) và giải thuật được lặp lại nhiều lần (điều kiện dừng – dòng 6) cho việc chọn lời giải tốt hơn. Phần ưu và nhược điểm cũng tương tự như ở thuật

giải HEU1.

Trong cả 2 giải thuật HEU1 và HEU2, nhóm tác giả đã lựa chọn k = 10% đỉnh có bậc lớn nhất cùng chiến lược chọn ngẫu nhiên và số lần lặp lại là 30 lần (điều kiện dừng) – chất lượng của lời giải phụ thuộc vào k.

Phần thực nghiệm của cả 2 giải thuật HEU1 và HEU2 trên 37 bộ dữ liệu từ hệ thống thách thức DIMACS cho chất lượng của lời giải tốt hơn cả 2 giải thuật MAXCLIQUE-HEU1 và MAXCLIQUEHEU2. Tuy nhiên, chất lượng lời giải từ HEU1 và HEU2 chỉ đạt từ 69,00% đến 100,00% và từ 72,00% đến 100,00% so với kết quả tối ưu (ω_{Best}).

Bảng IV, các đỉnh của đơn đồ thị G vô hướng không có trọng số được sắp xếp giảm dần theo bậc.

Bảng II, mô tả kết quả thực nghiệm trên 10 bộ dữ liệu từ hệ thống DIMACS (10 bộ dữ liệu cho lời giải có chất lượng thấp trong 37 bộ dữ liệu) - gồm các thông tin sau: Tên bộ dữ liệu, $|V|$ là số đỉnh, $|E|$ là số cạnh, deg_{min} là bậc cực tiểu, deg_{max} là bậc cực đại, median là giá trị trung vị của bậc các đỉnh, iqr là độ trải giữa của bậc các đỉnh (Q3-Q1), ω_{Best} là chỉ số Clique tốt nhất hiện nay [7], ω_{Heu1} và ω_{Heu2}.

Minh họa thuật giải HEU1 theo đồ thị G ở Hình 1.

Bảng III, bậc của các đỉnh trên đơn đồ thị G vô hướng không có trọng số theo thứ tự các đỉnh.

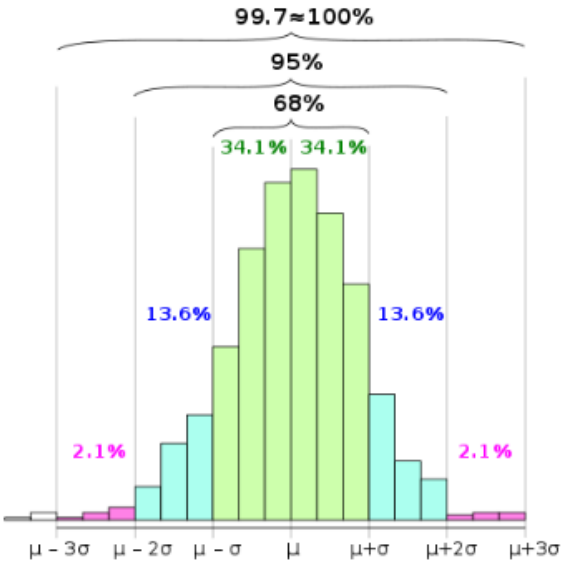
Từ Bảng III, thuật giải HEU1 và HEU2 chọn k = 10% đỉnh có bậc cao nhất trên đồ thị G. Khi đó, tập S = {1, 12}: giả sử chọn ngẫu nhiên đỉnh 1 – kết quả trả về chỉ số Clique ω_{Best}= 3; dựa vào Bảng I, ta thấy với ω_{Best} = 3 chưa phải là lời giải tốt (ω(G) = 4). Từ minh họa và phân tích trên, nhóm tác giả chỉ ra rằng khi chọn k = 10% đỉnh có bậc cao nhất và chọn ngẫu nhiên – đây chỉ là các lựa chọn mang tính chủ quan của nhóm tác giả [18]. Qua đó, cho thấy để giải bài toán Clique lớn nhất cho lời giải chất lượng tốt không chỉ dựa vào yếu tố ngẫu nhiên và cảm tính. Rất cần thiết cho việc đề xuất heuristic phù hợp hơn cho bài toán tìm Clique lớn nhất.

III. ĐỀ XUẤT HEURISTIC GIẢI BÀI TOÁN CLIQUE LỚN NHẤT

Phần đề xuất thuật giải hiệu quả cho bài toán tìm Clique lớn nhất, nhóm tác giả dựa trên nền tảng thống kê và phân tích đề xuất các heuristic phù hợp cho bài toán tìm Clique lớn nhất.

1. Một số đại lượng mô tả sự phân bố của dữ liệu trong thống kê

Quy tắc ba Sigma 68-95-99,7 (Empirical Rule). Đây là quy tắc đã kiểm chứng thường được sử dụng trong thống kê để dự báo các kết quả cuối cùng.

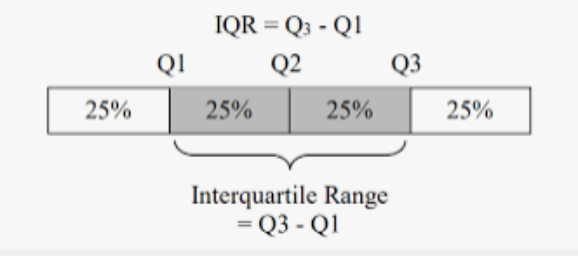


Hình 2. Quy tắc ba Sigma - giá trị trung bình μ và độ lệch chuẩn σ

Hình 2, theo quy tắc ba Sigma có 68% quan sát nằm trong độ lệch chuẩn đầu tiên ($\mu \pm \sigma$), 95% quan sát nằm trong hai độ lệch chuẩn đầu tiên ($\mu \pm 2\sigma$) và 99,7% nằm trong ba độ lệch chuẩn đầu tiên ($\mu \pm 3\sigma$). Ngoài ra, quy tắc ba Sigma còn cho thấy 15,70% các giá trị lớn nhất thuộc $[\mu + \sigma, \mu + 3\sigma]$.

Hình 3, minh họa các phân vị (Quartiles) và độ trải giữa (Interquartile Range - IQR). Các phân vị là đại lượng mô tả sự phân bố và sự phân tán của tập dữ liệu, gồm có 3 giá trị: phân vị thứ nhất (Q1), thứ hai (Q2) và thứ ba (Q3) – chia tập dữ liệu thành 4 phần có số lượng quan sát đều nhau; độ trải giữa của tập dữ liệu là hiệu số giữa phân vị thứ ba và thứ nhất, công thức tính theo phương trình sau:

$$IQR = Q3 - Q1 \quad (1)$$



Hình 3. Các phân vị (Quartiles) và độ trải giữa (InterQuartile Range)

Một vài quan sát điển hình từ Bảng II: ta thấy bộ dữ liệu *"hamming10-4"* có 1.024 đỉnh và các đỉnh có bậc bằng nhau 848, lời giải của cả 2 giải thuật đều cho chất lượng

Bảng II
KẾT QUẢ THỰC NGHIỆM CỦA HEU1 VÀ HEU2 TRÊN 10 BỘ DỮ LIỆU TỪ DIMACS

Dữ liệu	$\|V\|$	$\|E\|$	deg_{min}	deg_{max}	**median**	**iqr**	ω_{Best}	ω_{Heu1}	ω_{Heu2}
brock400 2	400	59.786	274	328	299	10,00	29	24	24
brock800 2	800	208.166	472	566	512	18,00	24	19	20
brock800 4	800	207.643	481	565	519	18,25	26	19	19
C2000.9	2.000	1.799.532	1.751	1.848	1.800	18,00	80	66	62
C4000.5	4.000	4.000.268	1.895	2.123	2.001	42,00	18	16	16
gen400 p0.9 55	400	71.820	334	375	360	13,25	55	50	50
gen400 p0.9 65	400	71.820	333	378	361	14,00	65	48	50
gen400 p0.9 75	400	71.820	335	380	359	13,00	75	52	54
hamming10-4	1.024	434.176	848	848	848	0,00	40	34	33
keller6	3.361	4.619.898	2.690	2.952	2.724	50,00	59	43	49

Bảng III
BẬC CỦA CÁC ĐỈNH TRÊN ĐỒ THỊ G THEO THỨ TỰ CÁC ĐỈNH.

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Bậc	7	2	2	2	2	2	2	4	3	4	3	7	2	2	2	2	2	2

Bảng IV
CÁC ĐỈNH TRÊN ĐỒ THỊ G ĐƯỢC SẮP XẾP GIẢM DẦN THEO BẬC.

Đỉnh	**1**	**12**	**8**	**10**	**9**	**11**	**2**	**3**	**4**	**5**	**6**	**7**	**13**	**14**	**15**	**16**	**17**	**18**
Bậc	7	7	4	4	3	3	2	2	2	2	2	2	2	2	2	2	2	2

Bảng V
KẾT QUẢ THỰC NGHIỆM TRÊN 10 BỘ DỮ LIỆU TỪ DIMACS

Dữ liệu	$\|V\|$	$\|E\|$	deg_{min}	deg_{max}	ω_{Best}	ω_{Heu1}	ω_{Heu2}	ω_{Heu1*}	ω_{Heu2*}
brock400 2	400	59.786	274	328	29	24	24	26	28
brock800 2	800	208.166	472	566	24	19	20	22	24
brock800 4	800	207.643	481	565	26	19	19	23	24
C2000.9	2.000	1.799.532	1.751	1.848	80	66	62	75	77
C4000.5	4.000	4.000.268	1.895	2.123	18	16	16	17	18
gen400 p0.9 55	400	71.820	334	375	55	50	50	52	53
gen400 p0.9 65	400	71.820	333	378	65	48	50	61	63
gen400 p0.9 75	400	71.820	335	380	75	52	54	67	71
hamming10-4	1.024	434.176	848	848	40	34	33	36	38
keller6	3.361	4.619.898	2.690	2.952	59	43	49	53	57

xấp xỉ nhau 85% và 82,25% ; bộ dữ liệu *"gen400 p0.9 75"* có 400 đỉnh và độ trải là 55 (Range = degmin - degmax = 380 – 335 = 55) cho thấy các đỉnh tương đồng rất nhiều – cả 2 giải thuật đều cho lời giải chất lượng thấp 69,33% và 72%.

2. Thuật giải HEU* cho bài toán tìm Clique lớn nhất

Trong phần này, nhóm tác giả trình bày thuật giải đề xuất tìm Clique lớn nhất dựa vào một số heuristic đề xuất như sau:

Heuristic_0: (dựa vào tính chất 1) Điều kiện ngắt trong quá trình tìm Clique lớn nhất, nếu chỉ số Clique ω_{Best} >

Algorithm 3: MCP-HEU* - xác định kích thước Clique lớn nhất của đồ thị.

Inputs: Đồ thị G = (V, E).
Outputs: Kích thước Clique lớn nhất - ω_G.
begin
  $\omega_{Best} = 0$;
  Chọn tập S theo heuristic_1 hoặc heuristic_2;
  foreach $v_i \in S$ **do**
    $U = \{\varnothing\}$;
    foreach $v_j \in adj(v_i)$ **do**
      If $deg(v_j) \geq \omega_{Best}$ $U = U \cup \{v_j\}$;
      while $V \neq \{\varnothing\}$ **do**
        Chọn $v \in U$ có bậc lớn nhất;
        U = U - {v};
        Chỉ giữ lại đỉnh $u \in U$: (u kề với đỉnh v và $deg(u) \geq \omega_{Best}$);
        $\omega_{Best} = \omega_{Best} + 1$;
      end
      if $|U| \geq \omega_{Best}$ **then**
        $\omega_{Best} = |U|$
      end
    end
  end
  return ω_{Best};
end

$deg(v_i) + 1$ (v_i là các đỉnh còn lại trong tập S).

Dựa vào quy tắc ba Sigma, nhóm tác giả đề xuất heuristic sau:

Heuristic_1: Chọn k = 16% (làm tròn từ 15,70%) đỉnh có bậc lớn nhất - nghĩa là chọn từ đỉnh có bậc lớn nhất đến đỉnh có bậc xấp xỉ giá trị $(\mu+\sigma)$ được làm tròn xuống (lấy phần nguyên);

Ví dụ 1: theo đồ thị G có 18 đỉnh và 26 cạnh, với k = 16% tương ứng tập S có 3 đỉnh từ đỉnh có bậc lớn nhất đến đỉnh có bậc xấp xỉ $(\mu + \sigma) = 4$ ($\mu = 2,89; \sigma = 1,64$).

Tương tự, dựa vào giá trị trung vị median và độ trải giữa iqr (xác định Q1, Q2, Q3) – nhóm tác giả đề xuất heuristic cho việc chọn đỉnh tiềm năng trong tập S:

Heuristic_2: Chọn k = 16% đỉnh có bậc lớn nhất theo tỷ lệ 8 : 5 : 3 trên từng khoảng phân vị - chọn ngẫu nhiên 8% đỉnh có bậc thuộc [Q3, degmax]; 5% thuộc [Q2, Q3] và 3% thuộc [Q1, Q2]. Tỷ lệ 8 : 5 : 3 có thể tùy biến.

Ví dụ 2: theo đồ thị G có 18 đỉnh và 26 cạnh, median = 2, iqr = 1 (Q1=2, Q2 = 2, Q3 = 3), degmin = 1, degmax = 7. Khi đó, chọn ngẫu nhiên 8% đỉnh có bậc thuộc [3, 7]; 5% đỉnh có bậc thuộc [2, 3] và 3% đỉnh có bậc thuộc [2, 2] – kết quả nhận được tổng số đỉnh thuộc tập S có thể lớn hơn k = 16% do làm tròn (8% chọn 2 đỉnh; 5% chọn 1 đỉnh và 3% chọn 1 đỉnh – có 4 đỉnh được chọn cho tập S).

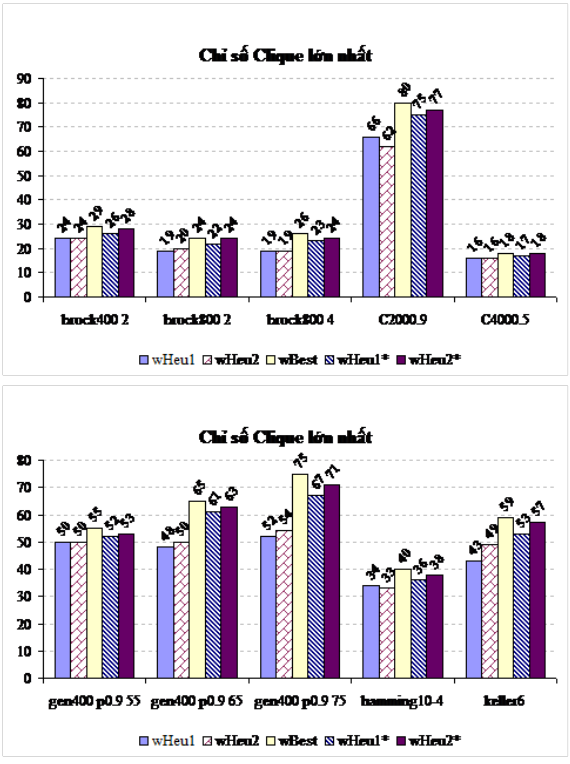
Trong nghiên cứu này, nhóm tác giả chọn giải thuật HEU2 cho thực nghiệm cải tiến thay thế các heuristic vừa trình bày ở trên.

Thuật giải MCP-HEU1* là thuật giải ở dòng 2 sử dụng heuristic_1; thuật giải MCP-HEU2* là thuật giải dùng heuristic_2 ở dòng 2. Ở cả 2 thuật giải, tập S được chọn trước và đây là tập chứa các đỉnh tiềm năng thuộc Clique lớn nhất.

IV. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Thực nghiệm trên máy tính Core i7-3540M CPU 3.0 GHz, 4Gb RAM, thuật toán cài đặt trên C#, Microsoft Visual Studio 2015. Dữ liệu thực nghiệm. Dữ liệu được chọn từ hệ thống DIMACS cho thách thức giải bài toán Clique lớn nhất gồm có 37 bộ dữ liệu [7]. Tuy nhiên, nhóm tác giả chỉ chọn 10 bộ dữ liệu (giải thuật HEU1_improve, HEU2_improve cho lời giải chất lượng chưa tốt, xấp xỉ 70%) theo Bảng V cho phần thực nghiệm so sánh độ chính xác cùng với 2 giải thuật HEU1_improve, HEU2_improve [18] – thuận tiện cho việc so sánh kết quả thực nghiệm, nhóm tác giả viết tắt tên của 2 giải thuật trên là HEU1 và HEU2.

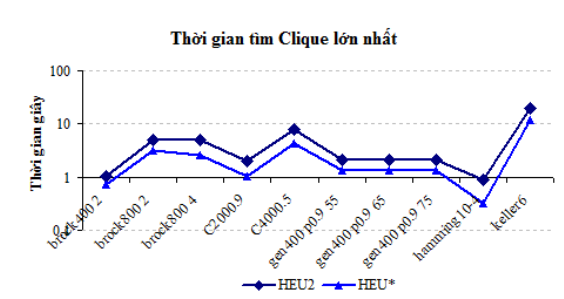
Bảng V, mô tả 10 bộ dữ liệu thực nghiệm từ hệ thống DIMACS - gồm các thông tin sau: Tên bộ dữ liệu, $|V|$ là số đỉnh, $|E|$ là số cạnh, degmin là bậc cực tiểu, degmax là bậc cực đại, ω_{Best} là chỉ số Clique tốt nhất hiện nay [7], ω_Heu1, ω_Heu2, ω_Heu1* và ω_Heu2*.



Hình 4. Chỉ số Clique của các thuật giải trên 10 bộ dữ liệu từ DIMACS

Hình 4, cho thấy chất lượng lời giải của 2 thuật giải HEU1* và HEU2* từ 88,46% đến 100% tốt hơn HEU1,

HEU2. Ngoài ra, thuật toán cũng cần thực nghiệm thêm trên nhiều bộ dữ liệu có mật độ khác nhau, cũng như trên nhiều dữ liệu cỡ lớn.



Hình 5. Thời gian (giây) tìm MCP của các thuật giải trên 10 bộ dữ liệu

Hình 5, cho thấy thời gian xác định chỉ số Clique của thuật giải HEU* nhanh hơn HEU2 rất nhiều – do heuristic_1 và heuristic_2 được chọn trước với tỷ lệ phù hợp trên các khoảng.

Tóm lại, các heuristic được đề xuất mang lại hiệu quả cao về chất lượng lời giải và thời gian thực hiện nhanh hơn các giải thuật gần đây.

V. KẾT LUẬN

Nhóm tác giả đã phân tích ưu và nhược điểm của một số thuật giải gần đây và đề xuất các heuristic tìm Clique lớn nhất với chất lượng lời giải tốt hơn. Tỷ lệ ở heuristic_2 có thể điều chỉnh tùy biến và mở rộng trên bốn khoảng phân vị.

Dựa trên kết quả đạt được từ thực nghiệm: Tương lai, nhóm tác giả mở rộng giải thuật để có thể giải bài toán Clique lớn nhất trên đồ thị vô hướng có trọng số cạnh và đỉnh; đồng thời nghiên cứu đề xuất các heuristic cũng như kết hợp các heuristic nhằm nâng cao chất lượng lời giải.

TÀI LIỆU THAM KHẢO

[1] W. Myrvold, P. Tania, W. Neil, "A Dynamic Programming Approach for Timing and Designing Clique Algorithms", *Proc of ALEX98*, 88-95, 1998.

[2] Strickland D. M., E. Barnes, J. Sokol, "Optimal Protein Structure Alignment Using Maximum Cliques", *Oper. Res*, 53, 389-402, 2005.

[3] D. Kumlander, "Comparing the best maximum clique finding algorithms, which are using heuristic vertex colouring", *10th WSEAS Inter Conf on COMPUTERS*, 932-937, 2006.

[4] E. Tomita, Y. Sutani, T. Higashi, M. Wakatsuki, "A simple and faster branch-and-bound algorithm for finding a maximum clique with computational experiments", *IEICE Trans. Inf. Syst.*, 96(6), 1286–1298, 2013.

[5] B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W. Liao, A. Choudhary, "Fast Algorithms for the Maximum Clique Problem on Massive Graphs with Applications to Overlapping Community Detection. Internet Mathematics", 11(4-5), 421–448, 2015.

[6] K. Schiff, "An Ant Algorithm for the Maximum Clique Problem in a Special Kind of Graph. Journal of Automation", *Mobile Robotics and Intelligent Systems 9*, 20-23, 2015.

[7] http://iridia.ulb.ac.be/ fmascia/maximum_clique/DIMACS-benchmark, 2015.

[8] P.S. Segundo, A. Lopez, P.M. Pardalos, "A new exact maximum clique algorithm for large and massive sparse graphs", *Comput. Oper. Res*, 66, 81–94, 2016.

[9] C. Li, C. Di, S. Hao, C. Jiahui, Z. Yang, "Heuristic Maximum Clique Based Identity Switches Awareness for Tracking", *Inter Conf ACAAI 2018*, 129-133, 2018.

[10] F. Fakhfakh, M. Tounsi, M. Mosbah, K. A. Hadj, "Algorithms for Finding Maximal and Maximum Cliques: A Survey", *In: Abraham A., Muhuri P., Muda A., Gandhi N. (eds). ISDA 2017*, AISC, 736, 745-754, 2018.

[11] S. Fotoohi, S. Saeidi, "Discovering the Maximum Clique in Social Networks Using Artificial Bee Colony Optimization Method", *IJITCS* 11, 1-11, 2019.

[12] C. Dai, J. Wu, D. Pi, S. Becker, C. Lin, Q. Zhang, B. Johnson, "Brain EEG Time-Series Clustering Using Maximum-Weight Clique", *IEEE transactions on cybernetics*, 1-15, 2020.

[13] M. O., Edwin, R. A. Mora-Gutiérrez, B. Obregón-Quintana, S. de-los-Cobos-Silva, E. A. Rincón-García, P. Lara-Velázquez and M. A. G. Andrade, "Mexican University Ranking Based on Maximal Clique", 327-395, 2020.

[14] V. Balash, A. Stepanova, V. Daniil, S. Mironov, F. Alexey, S. Sidorov, "Testing a Heuristic Algorithm for Finding a Maximum Clique on DIMACS and Facebook Graphs", *Wseas Transactions on Systems and Control*, 15, 93-101, 2020.

[15] Vũ Đình Hòa, Đỗ Trung Kiên, "Thuật toán song song giải bài toán xác định clique cực đại trên đồ thị", *Hội thảo Quốc gia: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông*, 426-442, 2009.

[16] Đàm Thanh Phương, Ngô Mạnh Tưởng, Khoa Thu Hoài, "Bài toán clique lớn nhất - ứng dụng và những thách thức tính toán", *Tạp chí KHCN - Chuyên san Khoa học Tự nhiên - Kỹ thuật, Đại học Thái Nguyên, ISSN: 1859-2171*, Tập 102, Số 2, 13-17, 2013.

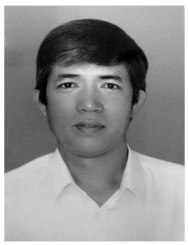
[17] Huỳnh Thanh Tân, Nguyễn Văn Thành, "Đề xuất thuật toán local search giải bài toán max clique", *Hội nghị KHCN Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR)*, 148-155, 2017.

[18] Phan Tấn Quốc, Huỳnh Thị Châu Ái, "Cải tiến một số thuật toán heuristic giải bài toán clique lớn nhất", *Hội nghị KHCN Quốc gia lần thứ XII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR)*, 64-72, 2019.

[19] Huỳnh Thị Châu Ái, "Đề xuất thuật toán giải gần đúng cho bài toán clique lớn nhất", *Luận văn Thạc sỹ chuyên ngành Khoa học Máy tính, Trường Đại học Sài Gòn*, 2019.

SƠ LƯỢC VỀ TÁC GIẢ

Phan Thành Huấn



Nhận bằng Thạc sỹ Đảm bảo Toán học cho Máy tính và Hệ thống tính toán năm 2016 tại trường Đại học Khoa học Tự nhiên, ĐHQG Tp.HCM. Hiện công tác tại trường Đại học Khoa học Xã hội và Nhân văn, ĐHQG Tp.HCM. Lĩnh vực nghiên cứu: Trí tuệ nhân tạo, khai thác dữ liệu, tính toán hiệu năng cao.
Email: huanphan@hcmussh.edu.vn

Huỳnh Thị Châu Ái

Nhận bằng Thạc sỹ Khoa học máy tính năm 2020 tại Trường Đại học Sài Gòn Tp.HCM. Hiện công tác tại Khoa Kỹ thuật Công nghệ, Trường Đại học Văn Hiến. Lĩnh vực nghiên cứu: Tính toán hiệu năng cao, giải thuật tối ưu.
Email: aihtc@vhu.edu.vn
Điện thoại: 0983 764 768

Châu Lê Sa Lin

Nhận bằng Thạc sỹ Hệ thống thông tin năm 2017 tại Trường Đại học Cần Thơ. Hiện công tác tại Khoa Công nghệ thông tin-truyền thông, Trường Cao đẳng Kinh tế - Kỹ thuật Cần Thơ. Lĩnh vực nghiên cứu: Tính toán hiệu năng cao, giải thuật tối ưu.
Email: clsalin@ctec.edu.vn
Điện thoại: 0945 017 660

Phân lớp ảnh bằng cây KD-Tree cho bài toán tìm kiếm ảnh tương tự

Nguyễn Thị Định [1,3], Văn Thế Thành[2], Lê Mạnh Thạnh[3]

[1] Khoa Công nghệ Thông tin, Trường ĐH Công nghiệp Thực phẩm TP.HCM

[2] Phòng quản lý khoa học và đào tạo sau Đại học, Trường ĐH Công nghiệp Thực phẩm TP.HCM

[3] Trường Đại học Khoa học, Đại học Huế

Tác giả liên hệ: Nguyễn Thị Định, nguyenthidinh.hcm@gmail.com

Ngày nhận bài: 15/04/2021, ngày sửa chữa: 01/06/2021, ngày duyệt đăng: 12/06/2021

Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.966

Tóm tắt: Trong bài báo này, một cấu trúc KD-Tree (*k - Dimensional Tree*) cải tiến được xây dựng nhằm phân lớp dữ liệu hình ảnh và ứng dụng cho bài toán tìm kiếm ảnh tương tự gọi là CKD-Tree (*Classification k-Dimensional Tree*). Quá trình xây dựng cấu trúc CKD-Tree được thực hiện theo phương pháp học bán giám sát làm cơ sở cho quá trình phân lớp dữ liệu hình ảnh tại các nút trong, đồng thời gom cụm dữ liệu tại các nút lá. Kết quả phân lớp này được áp dụng cho quá trình tìm kiếm tập ảnh tương tự từ một ảnh đầu vào sau khi trích xuất véc-tơ đặc trưng. Để minh chứng cho cơ sở lý thuyết đã đề xuất, chúng tôi tiến hành xây dựng cấu trúc CKD-Tree thực nghiệm trên bộ ảnh COREL (*gồm 1000 ảnh, 10 phân lớp*) và bộ ảnh Wang (*gồm 10800 ảnh, 80 phân lớp*). Kết quả thực nghiệm truy vấn ảnh được so sánh với các công trình khác cùng bộ dữ liệu nhằm minh chứng phương pháp đề xuất của chúng tôi là hiệu quả và áp dụng tốt trong các hệ tìm kiếm dữ liệu đa phương tiện.

Từ khóa: *CKD-Tree, image classification, image retrieval, similar smage.*

Title: **Image Classification Using Kd-Tree for Image Retrieval Problem**

Abstract: In this paper, an improved KD-Tree (*k - Dimensional Tree*) structure is built to classify image data and applied to the similar image retrieval which called CKD-Tree (*Classification k-Dimensional Tree*). The CKD-Tree structure is built on the semi-supervised learning method. It is not only the background of the process of classifying image data at the inner nodes but also clustering data at the leaf nodes. This classification result is applied to retrieval the similar image set from an input image after extracting the feature vector. Proving for proposed theorem, we had experimental research of CKD-Tree structure on the COREL (*including 1000 images, 10 subclasses*) and Wang image set (*including 10800 images, 80 subclasses*). The experimental results of image retrieval evaluated with the recently published methods on the same dataset. This proof shows that our proposed method is effective and appropriate for multimedia data retrieval systems.

Keywords: *CKD-Tree, image classification, image retrieval, similar image.*

I. GIỚI THIỆU

Ngày nay, các phương pháp tìm kiếm ảnh được thực hiện bởi nhiều công trình nghiên cứu; trong đó tìm kiếm ảnh dựa trên nội dung CBIR (*Content-based Image Retrieval*) [3, 6] là phương pháp tìm kiếm dựa trên các đặc trưng cấp thấp như màu sắc, hình dạng, kết cấu, v.v.. Bài toán tìm kiếm ảnh đã trở nên cấp thiết đối với con người vì được ứng dụng trong nhiều lĩnh vực như: y tế, giáo dục, giải trí, địa lý, ứng dụng y sinh, v.v.. Sự phát triển của các thiết bị điện tử như camera, smartphone, v.v. đã làm cho ảnh số gia tăng nhanh và trở nên quen thuộc, gần gũi với cuộc sống con người. Điều này đã mang lại nhiều cơ hội và thách thức cho lĩnh vực tra cứu ảnh. Vì vậy, nhiều hệ tìm kiếm ảnh đã công bố [2, 3, 4, 5] nhằm ứng dụng vào nhiều lĩnh vực trong đời sống và nâng cao hiệu suất truy vấn ảnh. Bài toán tìm kiếm ảnh tương tự là một trong những vấn đề quan trọng của hệ thống tra cứu dữ liệu đa phương tiện được nhiều nhóm nghiên cứu quan tâm [6, 7]. Trong cách tiếp cận của chúng tôi, một kỹ thuật phân lớp dữ liệu hình ảnh dựa trên cấu trúc KD-Tree cải tiến nhằm tạo ra một quá trình phân lớp dữ liệu theo mô hình cây; đồng thời ứng dụng cho bài toán tìm kiếm ảnh tương tự.

Đóng góp của bài báo gồm: (1) Đề xuất phương pháp phân lớp dữ liệu dựa trên cấu trúc CKD-Tree; (2) Đề xuất

các thuật toán xây dựng cấu trúc CKD-Tree; gán nhãn cho nút lá; huấn luyện trọng số và thuật toán tìm kiếm tập ảnh; (3) Đề xuất mô hình tìm kiếm ảnh tương tự dựa trên cấu trúc CKD-Tree; (4) Xây dựng thực nghiệm và chứng minh tính đúng đắn của phương pháp đề xuất dựa trên bộ dữ liệu COREL [26], Wang [27].

Phần còn lại của bài báo gồm: Phần II, khảo sát và phân tích ưu, nhược điểm của một số công trình liên quan. Phần III, trình bày thuật toán xây dựng cấu trúc CKD-Tree, gán nhãn nút lá và huấn luyện trọng số. Phần IV, trình bày mô hình truy vấn ảnh và thuật toán tìm kiếm ảnh tương tự dựa trên cấu trúc CKD-Tree. Mô tả dữ liệu, xây dựng thực nghiệm và kết quả được đánh giá trên bộ ảnh COREL (1000 ảnh) và bộ dữ liệu Wang (10800 ảnh) được trình bày trong phần V; Phần VI là kết luận và hướng phát triển.

II. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Phân lớp ảnh và ứng dụng cho bài toán tìm kiếm ảnh tương tự đã được nhiều công trình thực hiện với các kỹ thuật học máy khác nhau: phân lớp bằng thuật toán tìm kiếm láng giềng gần nhất k-NN (*k-Nearest Neighbors*) [10, 13]; phân lớp bằng kỹ thuật SVM (*Support Vector Machines*) [9]; phân lớp bằng Naïve Bayes [14]; phân lớp theo cấu trúc cây như cây quyết định (*Decision Tree*), cây tìm kiếm đa chiều (*KD-Tree*) [10, 12], v.v..

Yuqian Zhang và cộng sự (2016) [8] sử dụng phương pháp phân chia vùng ảnh để phác thảo và nhận diện khuôn mặt. Trong công trình này, cấu trúc KD-Tree được xây dựng trên tập ảnh phân vùng nhằm phân lớp và lưu trữ dữ liệu. Cây KD-Tree được xây dựng theo cấu trúc chỉ mục nhằm giảm thời gian tìm kiếm đồng thời

Kết quả nhận diện khuôn mặt với thuật toán tìm kiếm láng giềng k-NN. Thực nghiệm trên bộ ảnh khuôn mặt phác thảo CUFS (*Chinese University Face Sketch*) chứng minh rằng phương pháp đề xuất của tác giả là hiệu quả. Tuy nhiên, cấu trúc KD-Tree xây dựng theo mô hình phân lớp dữ liệu chưa kết hợp các kỹ thuật học máy nhằm tăng hiệu suất nhận diện khuôn mặt.

Dan Gao và cộng sự (2018) [9] thực hiện một so sánh giữa kỹ thuật phân lớp dữ liệu lớn bằng thuật toán SVM kết hợp cấu trúc KD-Tree cho bộ dữ liệu SDSS (*Sloan Digital Sky Survey*) và 2MASS (*Two-Micron All Sky Survey*). Xét tốc độ và độ chính xác thì việc phân lớp theo KD-Tree mang lại hiệu suất cao hơn so với kỹ thuật SVM. Bên cạnh đó, những phần tử phân loại sai của SVM và KD-Tree là trùng khớp nhau. Điều này chứng tỏ rằng phương pháp phân lớp dữ liệu bằng cây KD-Tree là hoàn toàn khả thi và hiệu quả, có thể so sánh với các phương pháp phân lớp trước đây.

Wenfeng Hou và cộng sự (2018) [27] đã thực hiện một phương pháp phân lớp dữ liệu và áp dụng cho bài toán tìm kiếm ảnh bằng cách kết hợp thuật toán tìm kiếm láng giềng k-NN và cấu trúc KD-Tree. Trong công trình này, tác giả kết hợp thuật toán k-NN và cấu trúc KD-Tree để xây dựng cây k-NN-KDTree theo mô hình phân lớp bằng phương pháp học bán giám sát. Tại mỗi tầng trên cây lần lượt chọn mỗi chiều $x_0, x_1, \ldots, x_k$ đại diện trong véc-tơ $f(x_0, x_1, \ldots, x_k)$ làm cơ sở cho quá trình so sánh và xây dựng cây nhị phân. Kết quả thực nghiệm so sánh hiệu suất tìm kiếm với phương pháp chỉ dùng thuật toán k-NN trên các bộ dữ liệu ảnh chứng minh rằng sự kết hợp k-NN và cây KD-Tree đã mang lại hiệu quả cao; đồng thời so sánh thời gian huấn luyện trên các bộ ảnh thì kết quả đề xuất phương pháp kết hợp k-NN-KDTree chỉ bằng ¼ so với phương pháp chỉ dùng thuật toán k-NN.

Fengquan Zhang và cộng sự (2019) [11] đã thực hiện một phương pháp đối sánh hình ảnh bằng cách xây dựng cấu trúc Vocabulary-KD; ứng dụng cho các bộ dữ liệu tăng trưởng; đồng thời cải thiện thời gian tìm kiếm. Cuối cùng tác giả đề xuất giải pháp đa tiến trình song song cho quá trình xây dựng và tìm kiếm đồng thời dựa trên cấu trúc Vocabulary-KD. Tuy nhiên, tác giả chưa đề cập đến quá trình xây dựng cấu trúc Vocabulary-KD theo phương pháp phân lớp với đa tiến trình nhằm tăng hiệu suất tìm kiếm từ kết quả phân lớp.

Reid Pinkham và cộng sự (2020) [26] đã thực hiện một phương pháp tối ưu hóa bộ nhớ và hiệu suất truy vấn các đối tượng 3D bằng hình ảnh theo phương pháp tìm kiếm láng giềng k-NN kết hợp cấu trúc KD-Tree. Trong công trình này, nhóm tác giả đã thực hiện tối ưu hóa thời gian truy vấn, tăng khả năng lưu trữ bằng cách thực hiện đa tiến trình song song, kết hợp tìm kiếm theo cấu trúc KD-Tree; với kiến trúc này dễ áp dụng cho các loại dữ liệu tăng trưởng đa chiều.

Từ các công trình nghiên cứu cho thấy phương pháp phân lớp dữ liệu hình ảnh sử dụng các kỹ thuật học máy kết hợp với cấu trúc KD-Tree là hoàn toàn khả thi, hiệu quả. Tuy nhiên, các công trình này hầu hết thực hiện phân lớp dữ liệu hình ảnh dựa cấu trúc KD-Tree chưa kết hợp thêm các kỹ thuật học máy hoặc chưa thực hiện phân lớp nhiều lần cho một đối tượng theo phương pháp học sâu nhằm nâng cao độ chính xác. Trong bài báo này, chúng tôi đề xuất một phương pháp phân lớp dữ liệu hình ảnh theo mô hình dạng cây gọi là cấu trúc CKD-Tree, đồng thời thực hiện tìm kiếm tập ảnh tương tự dựa trên cấu trúc đã xây dựng. Cấu trúc CKD-Tree đề xuất gồm các véc-tơ trọng số được lưu trữ tại các nút trong và dữ liệu hình ảnh được lưu trữ tại nút lá tạo thành các phân cụm. Cấu trúc CKD-Tree là một dạng phân lớp dữ liệu đa tầng nhằm phân lớp nhiều lần cho đối tượng theo mô hình Deep Learning, tại mỗi nhánh trên CKD-Tree thực hiện một lần phân lớp. Việc phân lớp này dùng để phân cụm dữ liệu và có thể phát hiện được các cụm

lân cận trong tìm kiếm tập ảnh tương tự. Điểm mới trong bài báo này là xây dựng cấu trúc CKD-Tree theo phương pháp phân lớp. Nút gốc và các nút trong chỉ lưu một véc-tơ trọng số ban đầu là ngẫu nhiên, sau đó được huấn luyện nhằm thực hiện quá trình phân lớp nhiều lần cho đối tượng tại các tầng trên cây. Nút lá lưu trữ tập ảnh tương tự gọi là các cụm tương đồng. Trong khi đó cấu trúc KD-Tree được công bố bởi các công trình trước đây dữ liệu được lưu tại tất cả các nút trên cây hoặc chưa thực hiện phân lớp cho đối tượng nhiều lần. Đồng thời, quá trình huấn luyện trọng số tại các nút trong theo phương pháp Gradient nhằm giảm sai số phân lớp hình ảnh được thực hiện song song với quá trình xây dựng cấu trúc CKD-Tree.

III. PHƯƠNG PHÁP XÂY DỰNG CẤU TRÚC CKD-TREE

1. Mô tả cấu trúc CKD-Tree

Cấu trúc CKD-Tree được xây dựng dựa trên cây KD-Tree nguyên thủy [1] là một cây nhị phân, cân bằng bao gồm: Một nút gốc (*Root*), tập nút trong $\{iNode\}$ và tập nút lá *{LNode}*. Ban đầu, cấu trúc CKD-Tree được tạo ra với một khung cây, nút gốc và nút trong lưu trữ véc-tơ trọng số, nút lá lưu trữ dữ liện hình ảnh với bộ dữ liệu huấn luyện (70% bộ ảnh thực nghiệm COREL, Wang). Một nút không phải là nút lá chia cây CKD-Tree thành hai phần gọi là cây con trái và cây con phải. Mỗi nút trong $(iNode_i)$ lưu trữ một véc-tơ trọng số n chiều $w_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Giá trị đầu ra y_i tại mỗi $iNode_i$ khi truyền vào một véc-tơ $f_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ được xác định bởi công thức (1). Trong đó, hàm truyền $Sigmoid(x)$ được sử dụng nhằm giới hạn miền giá trị đầu ra tại mỗi $iNode_i$ xác định bởi công thức (2); Vì CKD-Tree là cây nhị phân, cân bằng nên tại mỗi nút trong của CKD-Tree giá trị đầu ra được làm cơ sở để xác định đường đi tiếp theo thuộc một trong hai nhánh con trái hoặc phải. Do đó hàm dấu $Sign(y_i)$ được áp dụng ngay tại mỗi $iNode_i$ xác định bởi công thức (3). Đồng thời tại mỗi nút trong $iNode_i$ được ràng buộc bởi một ngưỡng t là điều kiện xác định số nhánh trên CKD-Tree.

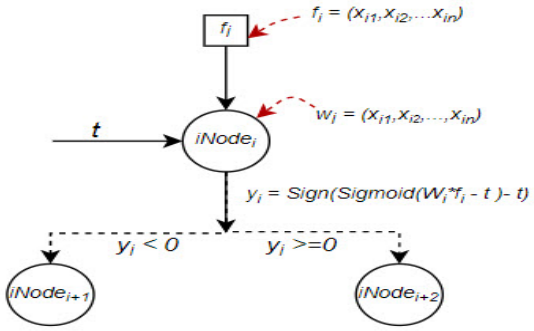
$$y_i = Sign(Sigmoid(w_i * f_j - t) - t) \quad (1) \tag{1}$$

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

$$Sign(y_i) = \begin{cases} 1, y_i \geq 0 \\ -1, y_i < 0 \end{cases} \tag{3}$$

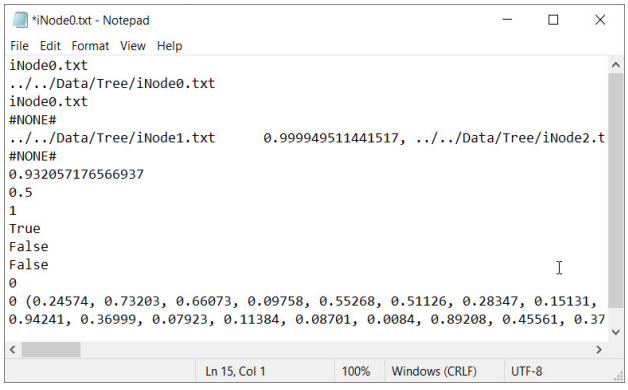
Trong quá trình tạo cây CKD-Tree, hàm $Sigmoid(x)$ được sử dụng nhằm xác định giá trị đầu ra tại mỗi $iNode_i$, sau đó thêm ngưỡng t bằng 0.5 là hằng số giúp tạo cây nhị phân, cân bằng trong miền giá trị [0..1]. Bên cạnh đó, cấu trúc KD-Tree trong công trình [18] xây dựng theo phương phân cụm, trong đó nút cha lưu giá trị trung bình trên mỗi nhánh, điều này dẫn đến hiệu suất gom cụm trên cây chưa cao, đó là điểm tồn tại của công trình này và phương pháp xây dựng CKD-Tree là một cải tiến hiệu quả.

Xét hàm dấu theo công thức (3) nếu $y_i \geq 0$ thì f_j qua nhánh phải; ngược lại f_j qua nhánh trái; xác định đường đi cho véc-tơ f_j tại $iNode_i$ nhằm tìm nhánh con thuộc tầng kế tiếp trên cây, quá trình này được mô tả như Hình 1.



Hình 1. Mô tả xác định nhánh con tại $iNode_i$

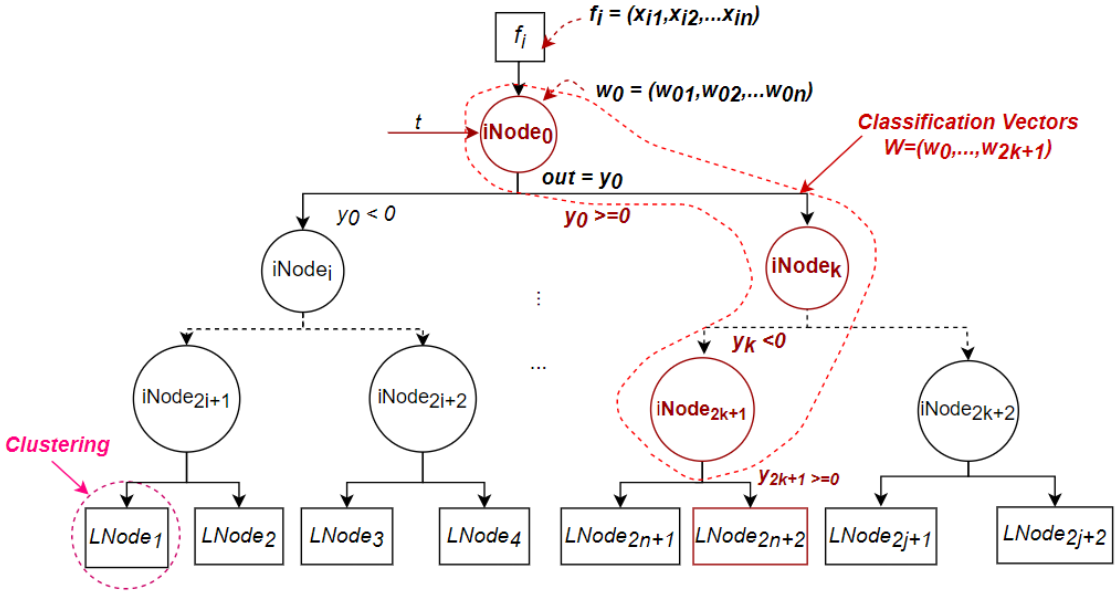
Vì giá trị đầu ra tại các $iNode_i$ thuộc miền [0..1] nên công thức (1) phải giới hạn hai lần giá trị t để miền giá trị của hàm dấu tại công thức (3) phân biệt giá trị âm, dương làm cơ sở xét f_j theo nhánh con trái hoặc con phải tại các nút trong. Hàm truyền $Sigmoid(x) = \sigma(x)$ có một số tính chất: 1) Miền giá trị là [0..1]; 2) Hàm $Sigmoid(x)$ tăng liên tục trên miền [0..1]; 3) Giá trị đạo hàm tồn tại với mọi điểm thuộc miền [0..1] giúp giảm chi phí huấn luyện trọng số.



Hình 2. Minh họa véc-tơ trọng số tại $iNode_i$

Véc-tơ $w_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ là trọng số được lưu trữ tại $iNode_i$ được minh họa như Hình 2. Các thành phần tại một nút và mối quan hệ giữa các nút trên cây CKD-Tree được mô tả như sau:

1) *Root* là nút không có nút cha, lưu trữ một véc-tơ trọng số (w_0), có hai nút con trái $(left)$, phải $(right)$ và có một mức trên cây $(level)$: $Root =< w_0, left, right, level >$



Hình 3. Minh họa quá trình phân lớp trên CKD-Tree

2) $iNode_i$ là nút trong có một nút cha $(parent)$; lưu trữ véc-tơ trọng số (w_i), hai nút con trái $(left)$, phải $(right)$ và có một mức $(level)$: $iNode_i =< parent, w_i, left, right, level >$

3) $LNode_i$ là nút lá có một nút cha $(parent)$, không có nút con; lưu trữ tập véc-tơ đặc trưng hình ảnh $f_1, f_2, \ldots, f_k$; có một mức $(level)$ và mỗi nút lá $LNode_i$ được gán một nhãn $label$: $LNode_i =< parent, f_1, \ldots, f_k, level, label >$

4) Hai nút $iNodei$ và $iNodej$ gọi là hai nút anh em nếu chúng có cùng một nút cha: $iNodei.parent = iNodej.parent$

5) Hai nút $iNodei$ và $iNode_j$ gọi là cha con nếu $iNode_j.parent = iNode_i$ hoặc $iNode_i.parent = iNode_j$

6) Hai nút $iNode_i$ và $iNode_j$ gọi là đồng cấp nếu $iNode_i.level = iNode_j.level$

2. Nguyên tắc xây dựng cấu trúc CKD-Tree

Trên cơ sở các thành phần của CKD-Tree, quá trình xây dựng được thực hiện theo các nguyên tắc và các ràng buộc sau:

a) Khởi tạo chiều cao cây bằng h thì số nút gốc $NumRoot = 1$; số nút trong $Num_iNode = 2^1 + \ldots + 2^{h-1}$, số nút lá tối đa trên cây là $NumLNode = 2^h$

b) Tại mỗi $iNode_i$ lưu trữ véc-tơ trọng số khởi tạo $iNode.w_i = (w_{i1}, w_{i2}, \ldots, w_{in})$. Ban đầu các véc-tơ trọng số w_i được khởi tạo ngẫu nhiên, sau đó được huấn luyện theo phương pháp giảm sai số trung bình tỷ lệ phân lớp tại các nút lá. Quá trình tạo cây cho các bộ dữ liệu ảnh khác nhau thì bộ trọng số này sẽ khác nhau.

c) Phần tử f_j được thêm vào CKD-Tree bằng cách duyệt từ nút gốc đến các tầng nút trong để tìm vị trí và lưu trữ f_j tại $LNode_k$. Tại mỗi nút $iNode_i$ trên cây, xác định giá trị đầu ra y_i bởi công thức (1)

d) Xác định đường đi cho véc-tơ f_j tại mỗi $iNode_i$: Nếu giá trị $y_i \geq 0$ đường đi của véc-tơ f_j qua nhánh phải; ngược lại đường đi của véc-tơ f_j qua nhánh trái.

e) Các bước (3) và (4) lặp lại đến tầng cuối cùng kiểm tra đúng nút lá thì ghi f_j vào $LNode_k$.

Dựa trên mô tả các thành phần và nguyên tắc xây dựng, cấu trúc CKD-Tree được minh họa như Hình 3, trong đó *Classification Vectors* là tập véc-tơ phân lớp $W =< w_0, w_2, \ldots, w_m >$, Clustring là một cụm nút lá, *out* $= y_i$ là giá trị đầu ra tại các $iNode_i$. Quá trình phân lớp cho một đối tượng được thực hiện theo nhiều tầng tại các nút trong $(iNode_i)$ của CKD-Tree và cuối cùng thực hiện gom cụm tại các nút *lá* trên CKD-Tree, mỗi nút lá là một cụm chứa các ảnh tương tự.

3. Thuật toán xây dựng CKD-Tree

Thuật toán xây dựng cấu trúc CKD-Tree – CKDT

Để xây dựng cấu trúc CKD-Tree, ban đầu khởi tạo bộ trọng số ngẫu nhiên $W_{kt} = W_{kt0}, W_{kt1}, \ldots, W_{ktp} : W_{kti} = (w_{i0}, w_{i1}, \ldots, w_{ik}); i = 0, \ldots n.$ Tập véc-tơ $F = f_i : f_i = (x_{i0}, x_{i1}, \ldots x_{ik}); i = 0, \ldots n$ của tập dữ liệu ảnh đã được trích xuất như được mô tả Phần V.1, mỗi véc-tơ đặc trưng gồm 81 chiều, giá trị ngưỡng t bằng 0,5 là hằng số không đổi theo trong công thức (1).

Thuật toán 1: Thuật toán CKDT

Đầu vào: Tập vector $F = f_i : f_i = (x_{i0}, x_{i1}, \ldots x_{ik}); i = 0, \ldots n$
Bộ trọng số khởi tạo $W_{kt} = W_{kt0}, W_{kt1}, \ldots, W_{ktp} : W_{kti} = (w_{i0}, w_{i1}, \ldots, w_{ik}); i = 0, \ldots n$
Đầu ra: Cây CKD-Tree
Function CKDT (F, W_{kt}, t)
begin
  int h; CKD-Tree = ø; $NumiNode = 0$;
  $MaxNumLNode = 2^h$;
  for $i = 0$ to $h - 1$ **do**
    $NumiNode = NumiNode + 2^i$;
  end
  for $j = 0$ to $NumiNode$ **do**
    $iNode_j = ø$;
  end
  for $k = NumiNode + 1$ to $2^h - 2$ **do**
    $LNode_k = ø$;
  end
  foreach (W_{kti}) **do**
    $iNode_i.w_i = w_{kti}$;
  end
  foreach $(f_i \in F)$ **do**
    foreach $(iNode_i)$ **do**
      $T = Sigmoid(w_{kti} * f_j + t)$;
      if $(Sign(T + t)) \geq 0)$ **then**
        $CKD - Tree = CKD - Tree \cup CKDT(f_j, CKDT.right, t)$;
      end
      else
        $CKD - Tree = CKD - Tree \cup CKDT(f_j, CKDT.left, t)$;
      end
    end
    if $LNode_i.leaf = true$ **then**
      $Insert(f_j, LNode_k)$;
    end
  end
  return CKD-Tree;
end

Thuật toán 2: Thuật toán gán nhãn - SL2L.

Đầu vào: Cây CKD-Tree, $ListLabels, ListLNode$
Đầu ra: Tập nút lá *LNode* đã gán nhãn *{ListLNodeAddedLabel}*
Function SL2L (*CKD-Tree, ListLabels, ListLNode*)
begin
  ListLNodeAddedLabel = null;
  foreach $LNode_i$ ***in*** *ListLNode* **do**
    foreach $Label_j$ ***in*** *ListLabels* **do**
      $TS = Count(Label_j, LNode_i)$;
    end
  end
  for $i = 0$ to n **do**
    for $j = 0$ to m **do**
      $MT(n, m)$ = **CreateMatrix**$(NumberLabel, NumberLNode)$;
    end
  end
  for $i = 0$ to n **do**
    for $j = 0$ to m **do**
      repeat
        SearchMax = $VT_{i,j}$;
        MaxLabel = Value $(VT_{i,j})$;
        $LNode_j.Label$ = SetLabel$(LNode_j, Label_i$, MaxLabel) ;
        ListLNodeNotAdd = ListLNode-$\{LNode_j\}$;
        ListLabelNotAdd =ListLabel-$\{Label_j\}$;
        foreach $(Label_k \; in \; LNode_k)$ **do**
          $VT(k, j) = 0$;
        end
        foreach $(LNode_k \; in \; ListLNodeNotAdd)$ **do**
          $VT(i, k) = 0$;
        end
      until $(VT_{i,j} = 0)$
    end
  end
  return ListLNodeAddedLabel
end

Mệnh đề 1: Độ phức tạp của thuật toán **CKDT** là $O(n * h)$, n là số véc-tơ xây dựng cây và h là chiều cao cây CKD-Tree.

***Chứng minh*:** Với mỗi véc-tơ thuộc tập n phần tử cần thêm vào cây CKD-Tree, thuật toán **CKDT** lần lượt duyệt qua tất cả các tầng từ gốc đến lá với chiều cao h. Do đó, độ phức tạp của Thuật toán **CKDT** là $O(n * h)$

4. Thuật toán gán nhãn nút lá

Nút lá $LNode_i$ lưu tập véc-tơ đặc trưng thuộc nhiều phân lớp. Vì vậy mỗi nút lá $LNode_i$ cần được gán một nhãn (*label*) theo số phần tử thuộc phân lớp nhiều nhất.

Bước 1: Mỗi nút lá $LNode_i$ đếm tần số xuất hiện của tất cả các nhãn trong bộ ảnh. Nhãn nào không xuất hiện trong $LNode_i$ thì gán tần số xuất hiện bằng 0.

Bước 2: Biểu diễn tần số xuất hiện các nhãn $Label_i$ theo từng nút lá $LNode_j$ bởi ma trận $MT(n, m) : \{n = NumberLabel; m = NumberLNode\}$

Bước 3: Tìm giá trị lớn nhất của ma trận$MT(n, m)$ là $VT_{i,j}$ tại dòng $Label_i$ và cột $LNode_j$. Gán nhãn $Label_i$ cho nút lá $LNode_j$ là giá trị $LNode_j.Label = Value(VT_{i,j})$

Bước 4: Chuyển các giá trị thuộc dòng i và cột j thành 0

Bước 5: Lặp lại **Bước 3** và **Bước 4** cho đến khi gán hết tập các nhãn $Label_i$ cho tập nút lá $LNode_j$

Thuật toán gán nhãn nút lá - SL2L: Để thực hiện gán nhãn cho nút lá tối ưu nhất, kỹ thuật dùng ma trận lưu trữ tần số xuất hiện của ảnh theo label tại các nút lá.

Mệnh đề 2: Độ phức tạp của thuật toán SL2L là $O(l * k)$ với l là số nhãn của tập dữ liệu ảnh, k số nút lá.

Chứng minh: Thuật toán SL2L duyệt qua k nút lá. Mỗi nhãn l trong tập dữ liệu cần được duyệt để gán cho các lá. Do đó, độ phức tạp của Thuật toán SL2L là $O(l * k)$.

5. Huấn luyện trọng số

Sau khi xây dựng cây CKD-Tree ban đầu với bộ trọng số ngẫu nhiên thì hiệu suất phân lớp ảnh chưa cao, vì vậy cần huấn luyện bộ trọng số tại các nút trong nhằm tăng hiệu suất phân lớp cho một ảnh đầu vào. Quá trình huấn luyện trọng số theo mô hình cây CKD-Tree sử dụng đạo hàm tại mỗi nút trong của cây. Công thức đạo hàm của hàm Sigmoid(x) xác định tại công thức (4).

$$\begin{aligned} Sigmoid'(x) &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}}\frac{e^{-x}}{1+e^{-x}} \\ &= \sigma(x)(1-\sigma(x)) \end{aligned} \tag{4}$$

$W_i = (w_{i1}, w_{i2}, ..., w_{in})$ là véc-tơ trọng số tại $iNode_i$, thực hiện cập nhật giá trị véc-tơ trọng số bằng hàm $\sigma(x) = \sigma(W_i * f_i - t)$ và $\eta = const$ theo công thức (5).

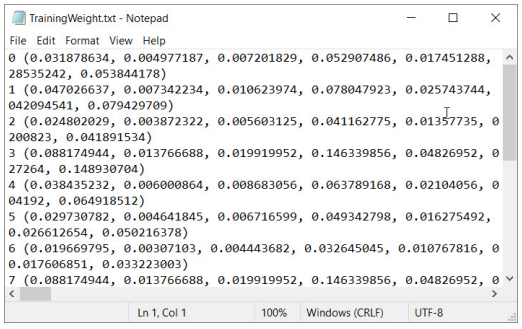
$$\begin{aligned} W_i =& W_i - Sign(\sigma(W_i * f_i - t) - t) * \eta * \sigma(W_i * f_i - t) \\ & * (1 - \sigma(W_i * f_i - t)) \end{aligned} \tag{5}$$

Sau khi thực hiện gán nhãn cho tất cả các nút lá, tính tổng sai số phân lớp theo công thức (6).

$$P_1 = AVG(\frac{\sum(f_{ki})}{\sum(f_{mi})}) \tag{6}$$

Trong đó:

$\sum(f_{ki})$ là số véc-tơ sai nhãn thuộc nút $LNode_i$ $\sum(f_{mi})$là số véc-tơ thuộc nút $LNode_i$. Mỗi lần cập nhật véc-tơ trọng số theo công thức (5), tính tổng sai số phân lớp theo công thức (6). Nếu $P_2 < P_1$ thì chọn phương án cập nhật véc-tơ trọng số mới; ngược lại giữ nguyên trọng số và chọn véc-tơ khác để thực hiện điều chỉnh trọng số. Kết quả huấn luyện bộ trọng số với bộ ảnh COREL, chiều cao cây bằng 4 minh họa bởi Hình 4.



Hình 4. Bộ véc-tơ trọng số đã huấn luyện trên tập ảnh COREL

Để tối ưu quá trình huấn luyện trọng số, cần theo một số quy tắc sau:

Quy tắc 1: Ưu tiên chọn véc-tơ có vị trí sai gần tầng lá nhất để điều chỉnh.

Quy tắc 2: Muốn điều chỉnh một véc-tơ từ f_j từ nhánh trái sang nhánh phải (hoặc ngược lại) tại vị trí thì chọn véc-tor có giá trị $|Sigmoid(W_i * f_j - t)|$ nhỏ nhất trong số các véc-tơ sai đường đi tại vị trí này để thực hiện điều chỉnh.

IV. MÔ HÌNH TÌM KIẾM ẢNH DỰA TRÊN CẤU TRÚC CKD-TREE

1. Mô hình truy vấn ảnh

Mô hình truy vấn ảnh dựa trên cấu trúc CKD-Tree được mô tả trong Hình 5, gọi là CBIR-CKDT (*Content Based Image Retrieval -Classification KD-Tree*). Hệ truy vấn gồm hai pha: 1) Pha tiền xử lý trích xuất đặc trưng bộ ảnh và xây dựng cấu trúc CKD-Tree; 2) Pha truy vấn thực hiện trích xuất đặc trưng ảnh đầu vào, sau đó thực hiện tìm kiếm trên cấu trúc CKD-Tree để trích xuất tập ảnh tương tự.

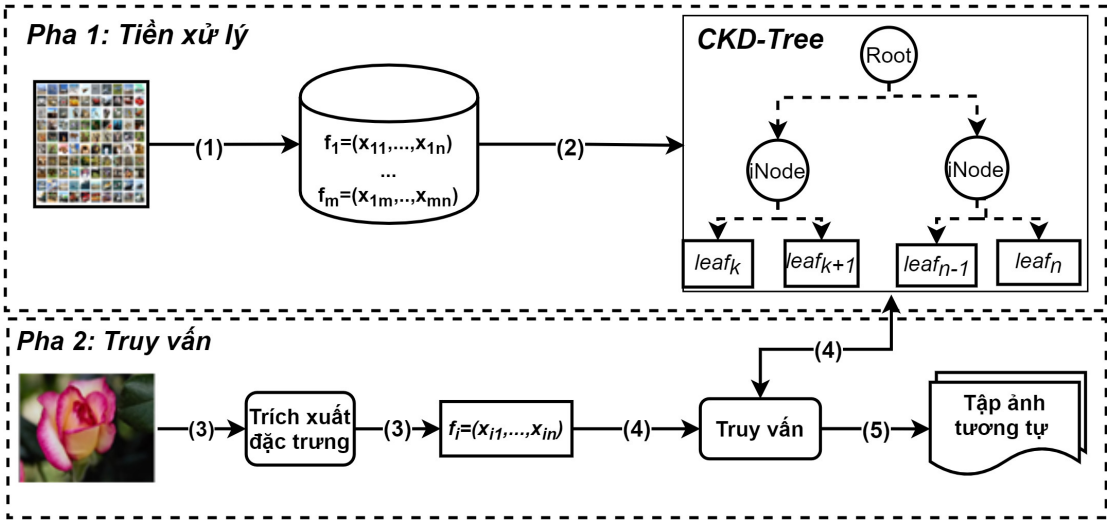
- **Pha tiền xử lý:** Xây dựng cấu trúc CKD-Tree
  a) Trích xuất vec-tơ đặc trưng của tập dữ liệu ảnh tạo cơ sở dữ liệu ban đầu, việc trích xuất đặc trưng dựa trên các đặc trưng cấp thấp của ảnh như màu sắc, diện tích, vị trí, bề mặt, đường biên của đối tượng ảnh, v.v.. Kết quả của quá trình trích xuất đặc trưng ảnh là tập các véc-tơ đặc trưng có 81 chiều và nhãn lớp.
  b) Xây dựng cấu trúc CKD-Tree nhằm lưu trữ dữ liệu hình ảnh tại nút lá. Phân lớp dữ liệu tại các nút trong.

- **Pha truy vấn:** Thực hiện tìm kiếm tập ảnh tương tự
  c) Trích xuất vec-tơ đặc trưng ảnh cần truy vấn.
  d)Truy vấn trên cây CKD-Tree để tìm tập ảnh tương tự bằng thuật toán tìm kiếm **SCKDT**.
  e) Kết xuất tập ảnh tương tự với ảnh truy vấn là tập ảnh thuộc nút lá chứa ảnh đầu vào.

2. Thuật toán truy vấn ảnh dựa trên cấu trúc CKD-Tree

Từ một ảnh đầu vào I_i sau khi thực hiện trích xuất vec-tơ đặc trưng f_i, quá trình thực hiện tìm kiếm tập ảnh tương tự của ảnh I_i theo các bước sau: 1) Duyệt từ nút gốc cho véc-tơ f_i đi qua nút trong tại mỗi tầng của cây CKD-Tree, tính giá trị đầu ra y_i theo công thức (3) tại mỗi $iNode_i$; 2) kiểm tra nếu $y_i \geq 0$ thì thực hiện tìm kiếm f_i qua nhánh phải của $iNode_i$; 3) ngược lại $y_i < 0$ thì thực hiện tìm kiếm f_i qua nhánh trái của $iNode_i$; 4) Khi duyệt hết các tầng nút trong, kiểm tra đến nút lá và trích xuất tập ảnh tương tự chính là tập ảnh lưu trữ tại nút lá chứa f_i.

Mệnh đề 3: Độ phức tạp của thuật toán **SCKDT** là $O(h)$; trong đó h là chiều cao của cây CKD-Tree.



Hình 5. Mô hình tìm kiếm ảnh dựa trên cấu trúc CKD-Tree

Thuật toán 3: Thuật toán truy vấn ảnh_*SCKDT*.

Đầu vào: Véc-tơ đặc trưng f_i của ảnh I_i , $CKD-Tree$
Đầu ra: Tập ảnh tương tự SI của ảnh I_i
Function SCKDT (f_i, CKD-Tree)
begin
  Result = ϕ ; double t = 0.5
  foreach f_i **do**
    foreach $iNode_i$ **do**
      $S = Sign(Sigmoid(Product(iNode_i.w_i, f_i) - t) - t)$
      if $S \geq 0$ **then**
        SCKDT $f_i, CKD-Tree.right$;
      end
      else
        SCKDT $f_i, CKD-Tree.left$;
      end
    end
  end
  if $Lnode_i.leaf = true$ **then**
    Result = Result $\cup LNode_i.f_i$;
  end

  return Result;
end

Chứng minh: Thuật toán SCKDT duyệt qua các nút từ gốc đến lá, cây CKD-Tree là cây cân bằng nên thuật toán SCKDT duyệt qua chiều cao h của cây. Do đó, độ phức tạp của thuật toán SCKDT là $O(h)$

V. THỰC NGHIỆM

1. Trích xuất đặc trưng hình ảnh

Trong bài báo này, chúng tôi tiến hành thực nghiệm với bộ dữ liệu ảnh COREL, Wang. Đầu tiên, các hình ảnh được trích xuất thành véc-tơ đặc trưng 81 chiều làm cơ sở xây dựng cấu trúc CKD-Tree cũng như làm dữ liệu đầu vào cho ảnh truy vấn. Các đặc trưng cấp thấp chúng tôi sử dụng gồm màu sắc, hình dạng, kết cấu, v.v.. Để trích xuất đặc trưng hình ảnh, đầu tiên chúng tôi thực hiện phân đoạn ảnh nhằm chia một ảnh thành nhiều vùng nhỏ góp dễ trích xuất đặc trưng. Sau đó các phương pháp trích xuất đặc trưng màu sắc, hình dạng, kết cấu [25] được thực hiện.

Phân đoạn ảnh (*Image Segmentation*): Phân đoạn ảnh là công việc đầu tiên trong trích xuất đặc trưng. Việc phân đoạn ảnh màu được thực hiện bằng cách phân chia hình ảnh thành các vùng riêng biệt nhau để từ đó trích xuất đặc trưng trên mỗi vùng. Để thực hiện được điều này, chúng tôi đề xuất phương pháp phân vùng dựa trên độ tương phản, nghĩa là vùng nào có độ tương phản thấp là hình nền, vùng nào có độ tương phản cao là hình đối tượng. Tuy nhiên, việc phân biệt hình nền và hình đối tượng sẽ bị nhập nhằng trong một số hình ảnh có xu hướng ngược lại nên chúng tôi sử dụng hai loại ảnh đối xứng nhau giữa hình nền và hình đối tượng. Đồng thời, để làm giảm độ nhiễu giữa các vùng quá sáng hoặc quá tối, một số điểm ảnh nằm trong vùng lân cận của giá trị lớn nhất và giá trị nhỏ nhất của độ tương phản thì được quy về giá trị tương đương [25].

Đặc trưng màu sắc (*Color*): Màu sắc là một trong những đặc điểm quan trọng nhất trong trích xuất đặc trưng cấp thấp hình ảnh. Màu sắc có mối quan hệ chặt chẽ với các ảnh đối tượng ảnh và ảnh nền. Đặc điểm của màu sắc là không thay đổi đối với kích thước và hướng của đối tượng [23]. Do đó, màu sắc dễ dàng phân tích và trích xuất bằng cách sử dụng mô men màu (*Color Moment*), biểu đồ màu (*Color Histogram*), không gian màu (*Color Space*), v.v.. Trong bài báo này, chúng tôi thực hiện trích xuất đặc trưng màu sắc theo MPEG-7 gồm 25 thuộc tính.

Đặc trưng hình dạng (*Shape*): Hình dạng đối tượng là một trong những đặc trưng cơ bản trong đặc trưng cấp thấp

của hình ảnh. Đặc trưng hình dạng được sử dụng để phát hiện các đối tượng tương tự từ cơ sở dữ liệu mà không phụ thuộc vào vị trí, góc quay hay sự co giãn của đối tượng ảnh [22, 24]. Các phương pháp trích xuất đặc trưng hình dạng thường được chia thành hai loại: dựa theo đường biên và dựa theo vùng ảnh. Kỹ thuật xác định đường biên đối tượng có thể dùng Gradient và phương pháp Laplacian. Vì vậy, để trích xuất các đặc trưng theo hình dạng, trước hết các đối tượng ảnh cần phải được phân đoạn thành những thành phần có cùng tính chất tương đồng dựa trên đường biên hay các vùng lân cận. Sau đó sử dụng các kỹ thuật phát hiện đường biên ảnh để xác định hình dạng cho đối tượng đã được tách ra khỏi hình nền [21]. Trong bài báo này, chúng tôi sử dụng phương pháp phát hiện biên đối tượng bằng phép biến đổi LoG (*Laplacian of Guassian*) [19] cho ảnh màu. Phép biến đổi LoG bất biến đối với sự biến đổi cường độ ảnh cũng như sự biến đổi tỉ lệ, phép quay, phép biến đổi affine. Vì vậy, giá trị Gaussian được xác định bởi công thức (7).

$$G\left(x, y, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \times \exp\left(-\frac{x^2 + y^2}{2.\sigma^2}\right) \tag{7}$$

Với σ là đạo hàm chuẩn, biểu diễn không gian tỷ lệ Gaussian $L(x, y, \sigma)$ của ảnh $f(x, y)$ theo công thức (8).

$$L(x, y, \sigma) = f(x, y) * G(x, y, \sigma) \tag{8}$$

Trong công thức (8), phép toán * là phép tích chập (*convolution*); *(x, y)* là tọa độ điểm ảnh. Toán tử Laplacian ∇^2 được tính toán theo công thức (9).

$$\nabla^2 = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{9}$$

Toán tử LoG được tính toán đầu tiên và sau đó được đối sánh với ảnh để tạo ra biểu diễn không gian tỷ lệ LoG theo công thức (10).

$$\nabla^2 G(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\pi\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{10}$$

Phương pháp LoG nhằm xác định đường biên, đặc trưng của đối tượng được trích xuất gồm chu vi đối tượng, vị trí tương đối của các đường viền.

Ngoài ra, để nhận dạng đối tượng dựa trên biên và làm mịn bề mặt, phép lọc *Sobel* [20, 21] được sử dụng. Phép lọc *Sobel* là toán tử Gradient dựa trên vùng láng giềng 3x3. Mặt nạ tích chập cho toán tử *Sobel* trên ảnh số tỷ lệ xám *Sobelx* và *Sobely* được xác định trong công thức (11).

$$Sobelx = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \; Sobely = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{11}$$

Các vị trí trên hai mặt nạ được áp dụng riêng biệt trên ảnh đầu vào để tạo hai thành phần Gradient Gx và Gy theo hướng ngang và dọc tương ứng theo công thức (12), (13).

$$Gx = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobelx_{i,j} . f_{x+1-2.y+j-2} \tag{12}$$

$$Gy = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobely_{i,j} . f_{x+1-2.y+j-2} \tag{13}$$

Lúc này, vị trí của ảnh đầu vào f được thể hiện trong công thức (14), trong đó trục là các vị trí nằm ngang và trục là các vị trí thẳng đứng.

$$\begin{bmatrix} (x-1, y-1) & (x-1, y) & (x-1, y+1) \\ (x, y-1) & (x, y) & (x, y+1) \\ (x+1, y+1) & (x+1, y) & (x+1, y+1) \end{bmatrix} \tag{14}$$

Độ lớn Gradient được tính bởi công thức (15).

$$G\left[f(x, y)\right] = \sqrt{Gx^2 + Gy^2} \tag{15}$$

Bảng I
CÁC GIÁ TRỊ TRÍCH XUẤT VÉC-TƠ ĐẶC TRƯNG CỦA HÌNH ẢNH

Mô tả đặc trưng	**Số giá trị**
Đặc trưng màu sắc theo MPEG-7	25
Phép lọc tần số cao để lấy ảnh đường nét	9
Phép lọc Gaussian để nâng cao cường độ ảnh	9
Đặc trưng cường độ các điểm ảnh theo láng giềng	9
Đặc trưng cường độ của đối tượng	9
Đặc trưng cường độ của hình nền	9
Đặc trưng diện tích đối tượng	1
Đặc trưng hình dạng của đường biên ảnh	1
Đặc trưng vị trí tương đối của đối tượng theo trục X, Y	2
Đặc trưng vị trí tương đối của hình nền theo trục X, Y	2
Đặc trưng chu vi của đối tượng	1
Đặc trưng chu vi của đối tượng theo phép lọc Sobel	1
Đặc trưng cường độ các điểm ảnh theo láng giềng dựa theo phép lọc Sobel	1
Đặc trưng chu vi của đối tượng theo phép lọc Laplacian	1
Đặc trưng đường nét ảnh theo phép lọc Laplacian	1

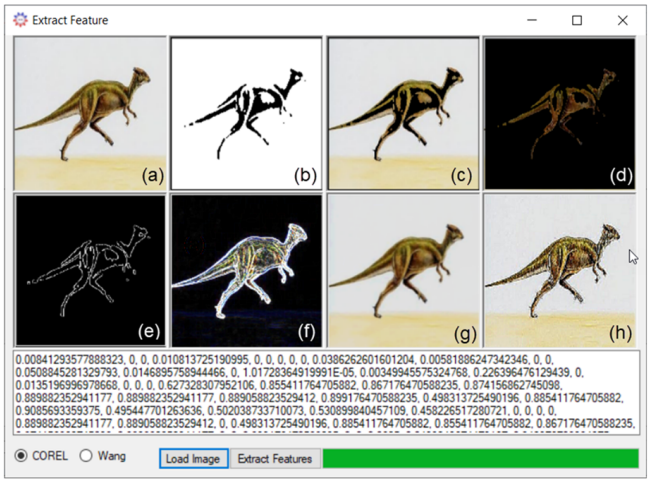
Để xác định vị trí tương đối của đối tượng theo trục X và trục Y, hàm khoảng cách trọng tâm CDF (*Centroid Distance Function*) được sử dụng. CDF tính toán khoảng cách từ tâm (x_0, y_0) đến các điểm đường viền của một hình dạng và được biểu diễn bằng công thức (16).

$$r(n) = \sqrt{(x(n) - x_0)^2 + (y(n) - y_0)^2} \tag{16}$$

Từ cơ sở lý thuyết, mỗi ảnh được trích xuất thành vec-tơ đặc trưng 81 chiều với các thành phần và số giá trị được mô tả trong Bảng I.

2. Môi trường xây dựng thực nghiệm

Thực nghiệm trích xuất đặc trưng và hệ truy vấn CBIR-CKDT được xây dựng trên nền tảng dotNET Framework 4.5, ngôn ngữ lập trình C#. Các đồ thị được xây dựng trên Mathlab 2015. Cấu hình máy tính: Intel(R) Core™ i5-5200U, CPU 2.2GHz, RAM 16GB và hệ điều hành Windows 10 Professional. Chúng tôi tiến hành thực nghiệm trích xuất véc-tơ đặc trưng cho bộ dữ liệu COREL và bộ ảnh Wang được minh họa bởi Hình 6. Quá trình xây dựng cây CKD-Tree được minh họa ở Hình 7. Cây CKD-Tree được xây dựng (*Create CKD-Tree*) xác định bởi chiều cao (*Height of CKD-Tree*), số nhánh (*Number of Brand*) và một ngưỡng (*Bias*). Hình 8 là giao diện tìm kiếm ảnh tương tự cho ảnh 450.jpg bộ COREL, Hình 9 là kết quả trích xuất tập ảnh tương tự của ảnh 450.jpg đầu vào.

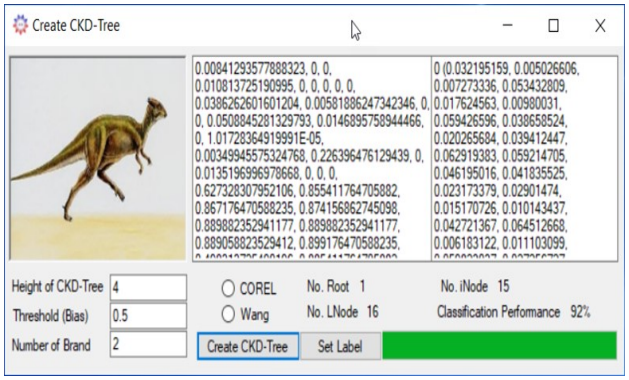


Hình 6. Véc-tơ đặc trưng ảnh 450.jpg bộ COREL

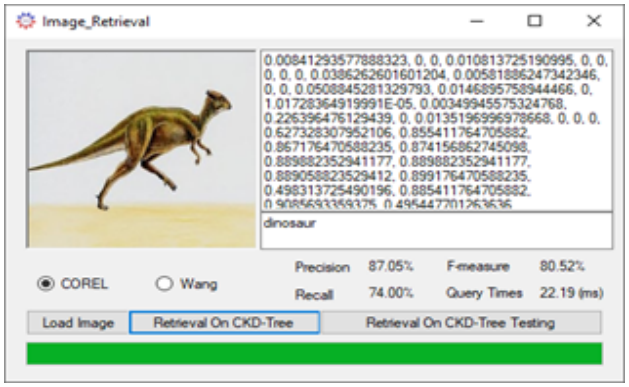
Kết quả trích xuất véc-tơ đặc trưng của ảnh 405.jpg bộ ảnh COREL được thể hiện bằng 8 ảnh được mô tả trong Hình 6 gồm: (a) ảnh gốc; (b) ảnh mặt nạ đối tượng (*ForeGround*); (c) ảnh đối tượng; (d) hình nền ảnh gốc; (e) ảnh đường biên đối tượng theo phép lọc Laplace cho ảnh mặt nạ đối tượng; (f) ảnh đường biên đối tượng và vân ảnh bề mặt đối tượng theo phép lọc Sobel; (g) Ảnh đường nét của ảnh gốc theo phép lọc Laplace và Gaussian; (h) ảnh phép lọc thông cao cho ảnh gốc.

Bảng II
HIỆU SUẤT TÌM KIẾM CỦA HỆ TRUY VẤN ẢNH CBIR-CKDT

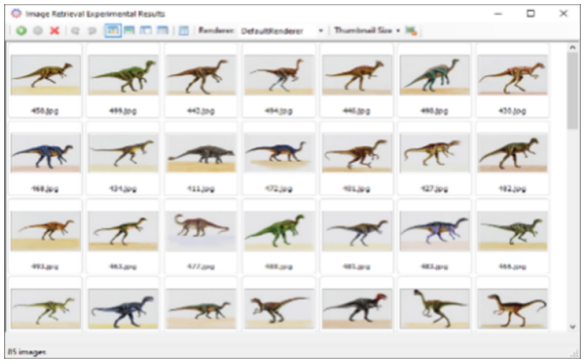
Tập ảnh	Độ chính xác trung bình	Độ phủ trung bình	Độ dung hòa trung bình	Thời gian truy vấn trung bình (ms)
COREL	0,7640	0,6878	0,7188	38
Wang	0,7327	0,6508	0,6869	87



Hình 7. Xây dựng cấu trúc CKD-Tree

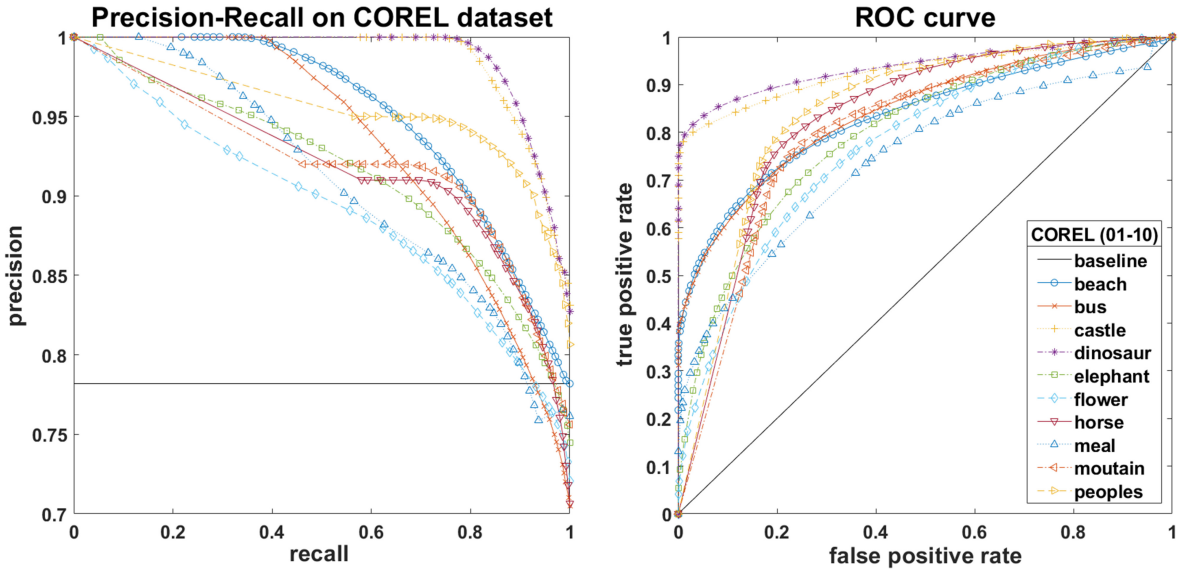


Hình 8. Tìm kiếm ảnh tương tự cho ảnh 450.jpg bộ COREL



Hình 9. Tập ảnh tương tự của ảnh 450.jpg bộ COREL

Hệ truy vấn ảnh CBIR-CKDT thực hiện truy vấn (*Retrieval On CKD-Tree*) ảnh đầu vào (*Load Image*) và kết quả tập ảnh tương tự; đồng thời thực nghiệm toàn bộ dữ liệu (*Retrieval On CKD-Tree Testing*). Kết quả thực nghiệm trên bộ ảnh COREL và bộ ảnh Wang được trình bày trong Bảng II cho thấy hiệu suất truy vấn trên bộ COREL cao hơn bộ dữ liệu Wang và thời gian truy vấn trên bộ ảnh COREL nhanh hơn bộ ảnh Wang là vì có sự khác nhau chiều cao cây CKD-Tree.



Hình 10. Precision-Recall và đường cong ROC bộ ảnh COREL

Bảng III
SO SÁNH HIỆU SUẤT TRUY VẤN GIỮA CÁC PHƯƠNG PHÁP TRÊN CÁC BỘ DỮ LIỆU

Phương pháp	Bộ dữ liệu	Độ chính xác trung bình
B_SHIFT , 2016 [15]	COREL	72%
CBIR-CKDT	**COREL**	**76,4%**
P. CHHABRA, 2020 [16]	Wang	63,2%
CBIR-CKDT	**Wang**	**73,27%**

Bảng IV
SO SÁNH THỜI GIAN TRUY VẤN GIỮA CÁC PHƯƠNG PHÁP TRÊN CÁC BỘ DỮ LIỆU

Phương pháp	Bộ dữ liệu	Thời gian truy vấn trung bình (ms)
B_SHIFT , 2016 [15]	COREL	5524,4
CBIR-CKDT	**COREL**	**37**
P. CHHABRA, 2020 [16]	Wang	144
CBIR-CKDT	**Wang**	**87**

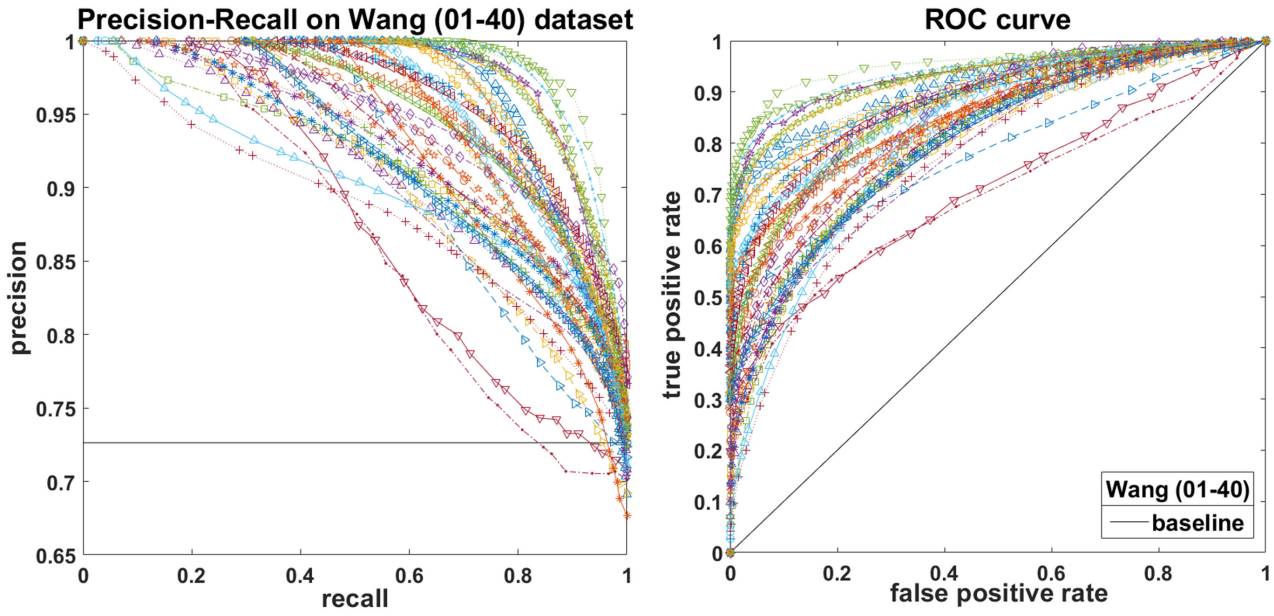
3. Đánh giá kết quả thực nghiệm

Kết quả thực nghiệm truy vấn ảnh của hệ CBIR-CKDT được minh họa bởi Hình 10, Hình 11 và Hình 12. Mỗi đường cong trên đồ thị mô tả kết quả độ chính xác (*Precision*) và độ phủ (*Recall*) thuộc một chủ đề trong bộ dữ liệu. Đường cong ROC (*ROC Curve*) cho biết tỷ lệ kết quả truy vấn đúng (*True positive rate*) và kết quả truy vấn sai (*False positive rate*). So sánh hiệu suất truy vấn giữa các bộ dữ liệu trên từng bộ ảnh COREL, Wang minh họa bởi Hình 13, Hình 15. Thời gian truy vấn trên hệ CBIR-CKDT được minh họa bởi đồ thị Hình 14, Hình 16.

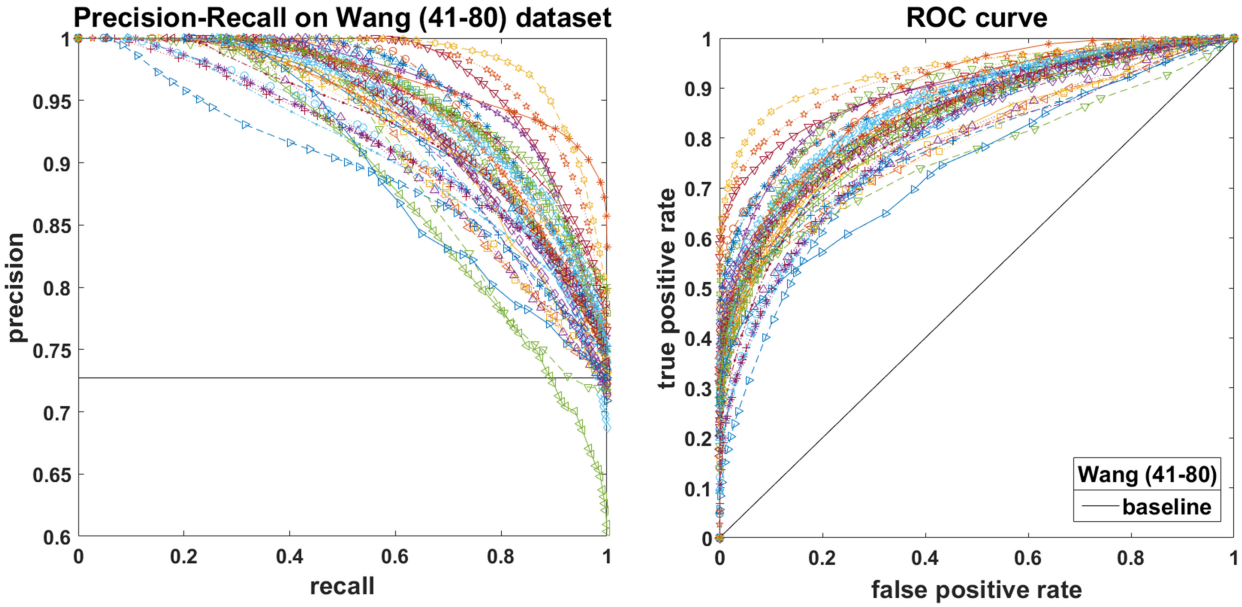
Hiệu suất và thời gian truy vấn trên hệ CBIR-CKDT được so sánh với công trình [15, 16] cho thấy hệ truy vấn ảnh CBIR-CKDT cao hơn về độ chính xác và độ phủ. Độ phủ của hệ B_SHIFT thấp là do tác giả chỉ lấy top 20 ảnh trong bộ dữ liệu truy vấn. Đồng thời độ chính xác trên hệ CBIR-CKDT cao hơn phương pháp B_SHIFT và P. Chhabra do: 1) CBIR-CKDT thực hiện phân lớp ảnh trước khi gom cụm để tìm tập ảnh tương tự; 2) CBIR-CKDT phân lớp cho đối tượng nhiều lần tại các nút trong của CKD-Tree theo mô hình Deep Learning; 3) CBIR-CKDT xây dựng cấu trúc dữ liệu lưu trữ ảnh số. Tuy nhiên, công trình [16] khi thực hiện trích xuất đặc trưng véc-tơ với đặc trưng SIFT, độ dài 8 thì độ chính xác là 86,2%. Do đó, kết quả trích xuất đặc trưng hình ảnh đóng một vai trò quan trọng trong nâng cao hiệu suất truy vấn ảnh. Bên cạnh đó, thời gian truy vấn trung bình của CBIR-CKDT thấp hơn là do quá trình tìm kiếm thực hiện theo sơ đồ hình cây KD-Tree được rẽ nhánh tại các tầng nên thời gian tối ưu.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, chúng tôi đã đề xuất một phương pháp phân lớp dữ liệu hình ảnh bằng cây KD-Tree cải tiến và áp dụng cho bài toán tìm kiếm ảnh tương tự là hoàn toàn khả thi và hiệu quả. Đây là một tính mới trong quá trình xây dựng cấu trúc CKD-Tree theo phương pháp phân lớp dữ liệu để hình thành các cụm tương đồng, quá trình này đã tích hợp phương pháp học có giám sát (huấn luyện trọng



Hình 11. Precision-Recall và đường ROC bộ ảnh Wang (1..40)

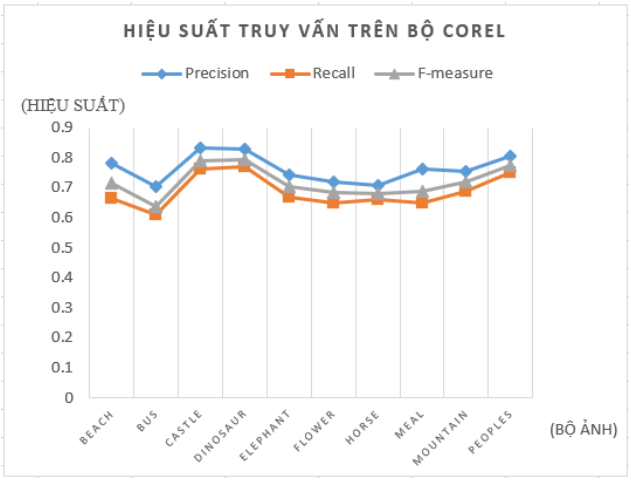


Hình 12. Precision-Recall, đường ROC bộ ảnh Wang (41..80)

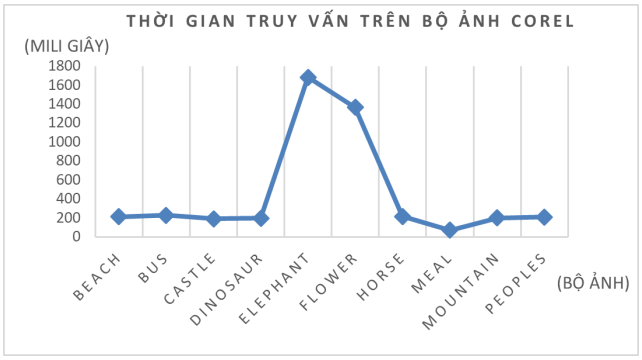
số trong xây dựng mô hình cây), bán giám sát (từ mô hình KD-Tree, thực hiện tìm kiếm ảnh theo nút lá để ra tập ảnh tương tự) và không giám sát (gom cụm theo trọng số tại nút lá sau khi huấn luyện trọng số). Hệ truy vấn ảnh CBIR-CKDT đã thực nghiệm trên bộ ảnh COREL, Wang được đánh giá dựa trên độ chính xác, độ phủ và độ dung hòa với bộ ảnh COREL lần lượt là: 76,04%, 68,78%, 71,88% và bộ ảnh Wang là: 73,27%, 65,08%, 68,69%. Hướng phát triển tiếp theo của chúng tôi là xây dựng một cấu trúc KD-Tree đa nhánh, cân bằng áp dụng cho các tập dữ liệu tăng trưởng theo phân lớp và không giới hạn số phân lớp ban đầu. Đồng thời thực hiện bài toán tìm kiếm ảnh theo ngữ nghĩa.

LỜI CẢM ƠN

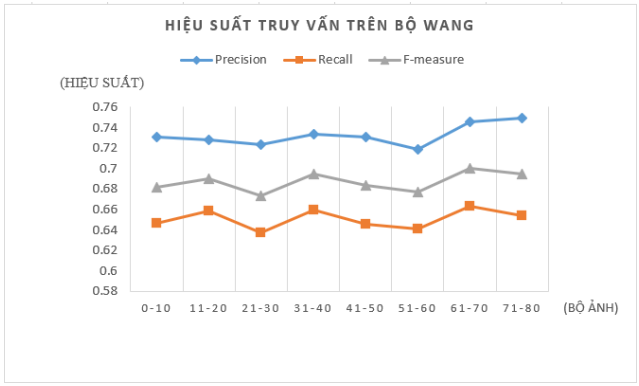
Chúng tôi xin trân trọng cảm ơn Khoa Công nghệ thông tin – Trường Đại học Khoa học - Đại học Huế, nhóm nghiên cứu SBIR-HCM đã góp ý chuyên môn cho nghiên cứu này.



Hình 13. Hiệu suất truy vấn trên bộ ảnh COREL



Hình 14. Thời gian truy vấn trên bộ ảnh COREL



Hình 15. Hiệu suất truy vấn trên bộ ảnh Wang

Chúng tôi xin trân trọng cảm ơn Trường Đại học Công nghiệp Thực phẩm TP. HCM, Trường Đại học Sư phạm TP.HCM đã tạo điều kiện về cơ sở vất chất giúp chúng tôi hoàn thành bài nghiên cứu này.

TÀI LIỆU THAM KHẢO

[1] Bentley, Jon Louis, "Multidimensional binary search trees used for associative searching", *Communications of the ACM*, vol. 18.9, pp. 509-517, 1975.

[2] Shen, Xiaohui, et al., "Spatially-constrained similarity measure for large-scale object retrieval", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 36.6, pp.1229-1241, 2013.

[3] P. Muneesawang, N. Zhang, L. Guan, "Multimedia Database Retrieval: Technology and Applications", *Springer, New York Dordrecht London*, 2014.

[4] Alzu'bi A, Amira A, Ramzan N, "Semantic content-based image retrieval: A comprehensive study", *J Vis Commun Image Represent*, Vol. 32, pp.20-54, 2015.

[5] Deloitte, "Photo sharing: trillions and rising", *Deloitte Touche Tohmatsu Limited, Deloitte Global*, 2016.

[6] A Patrizio, "Data center explorer", *Network World*, 03/12/2018, https://www.networkworld.com/article/3325397/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html.

[7] David Reinsel, John Gantz, John Rydning, "The Digitization of the World: From Edge to Core sponsored by Seagate", *IDC Technical Report*, 2018. https://www.seagate.com/as/en/our-story/data-age-2025/.

[8] Zhang, Yuqian, et al., "Fast face sketch synthesis via kd-tree search." *European Conference on Computer Vision. Springer, Cham*, pp. 64-77, 2016.

[9] Gao, Dan, Yan-Xia Zhang, and Yong-Heng Zhao, "Support vector machines and kd-tree for separating quasars from large survey data bases." *Monthly Notices of the Royal Astronomical Society*, Vol. 386.3, pp.1417-1425, 2008.

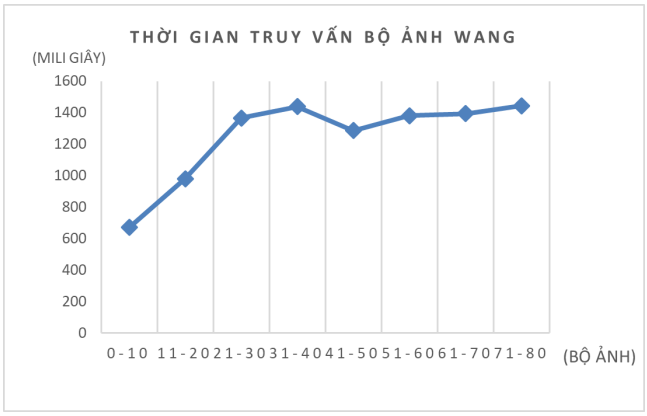
[10] Hou, Wenfeng, et al., "An advanced k nearest neighbor classification algorithm based on KD-tree", *IEEE International Conference of Safety Produce Informatization (IICSPI) IEEE*, pp. 902-905, 2018.

[11] Zhang, Fengquan, Yahui Gao and Liuqing Xu, "An adaptive image feature matching method using mixed Vocabulary-KD tree." *Multimedia Tools and Applications*, Vol. 1-19, pp. 16421 - 16439, 2019.

[12] Hemmer, Michael, and Ondrej Stava, "KD tree encoding for point clouds using deviations", *U.S. Patent*, No. 10, 496, 336, 3-Dec-2019.

[13] Pinkham, Reid, Shuqing Zeng, and Zhengya Zhang, "Quick-NN: Memory and Performance Optimization of kd Tree Based Nearest Neighbor Search for 3D Point Clouds", *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA) IEEE*, pp. 180-192, 2020.

[14] McCann, Sancho, David G. Lowe, "Local naive bayes nearest neighbor for image classification", *2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE*, pp. 3650-



Hình 16. Thời gian truy vấn trên bộ ảnh Wang

3658, 2012.
[15] Douik, Ali, Mehrez Abdellaoui, and Leila Kabbai, "Content based image retrieval using local and global features descriptor", *2016 2nd international conference on advanced Technologies for Signal and Image Processing (ATSIP) IEEE*, pp. 151-154, 2016.
[16] Chhabra, Payal, Naresh Kumar Garg, and Munish Kumar, "Content-based image retrieval system using ORB and SIFT features", *Neural Computing and Applications*, Vol. 32.7, pp. 2725-2733, 2020.
[17] Das, Rik, Sudeep Thepade, and Saurav Ghosh. "Novel feature extraction technique for content-based image recognition with query classification", *International Journal of Computational Vision and Robotics*, Vol. 7.1-2, pp. 123-147, 2017.
[18] Nguyễn Thị Định, Lê Thị Vĩnh Thanh, Nguyễn Văn Thịnh, Văn Thế Thành, "Một phương pháp phân cụm dựa trên cây KD-Tree cho bài toán tìm kiếm ảnh", *Tạp chí Khoa học Đại học Huế, Chuyên san Kỹ thuật và Công nghệ, ISSN: 2615-9732*, Tập 129, số 2A, 2020.
[19] Kong, H. A., "A generalized Laplacian of Gaussian filter for blob detection and its applications", *IEEE transactions on cybernetics*, Vol. 43(6), pp. 1719-1733, 2013.
[20] Bora, D. J., "A novel approach for color image edge detection using multidirectional Sobel filter on HSV color space", *Int. J. Comput. Sci. Eng*, Vol. 5(2), pp. 154-159, 2017.
[21] Gonzalez, C. I., "Edge detection methods based on generalized type-2 fuzzy logic", *Springer International Publishing*, pp. 21-35, 2017.
[22] He, L. F., "Fast basic shape feature computation", *In Computer Science and Artificial IntelligenceProceedings of the International Conference on Computer Science and Artificial Intelligence (CSAI 2016)*, pp. 22-48, 2017
[23] Vinayak, V., "CBIR system using color moment and color auto-Correlogram with block truncation coding", *International Journal of Computer Applications*, Vol. 161(9), pp. 1-7, 2017.
[24] Chaki, J., "A beginner's guide to image shape feature extraction techniques", *CRC Press*, pp. 89-131, pp. 15-68, 2019.
[25] Chaki, J., "Image Color Feature Extraction Techniques: Fundamentals and Applications", *Singapore: Springer*, pp. 29-80, 2021.
[26] Corel 1k Database, Available online: http://wang.ist.psu.edu/docs/related/. (n.d.).
[27] Wang, J. Z. (n.d.). James Z. Wang Group. Available: http://wang.ist.psu.edu/docs/home.shtml.

SƠ LƯỢC VỀ TÁC GIẢ

Nguyễn Thị Định

Sinh năm 1983, tốt nghiệp ngành Sư phạm tin học Trường Đại học Sư phạm TP.HCM vào năm 2006, nhận bằng Thạc sĩ ngành Truyền số liệu và mạng máy tính tại Học viện Công nghệ Bưu chính viễn thông TP.HCM vào năm 2011. Hiện là nghiên cứu sinh ngành Khoa học máy tính Trường Đại học khoa học, Đại học Huế. Lĩnh vực nghiên cứu: xử lý ảnh, tìm kiếm ảnh và cơ sở dữ liệu.
Email: dinhnt@hufi.edu.vn

Văn Thế Thành

Sinh năm 1979, tốt nghiệp chuyên ngành Toán tin tại Đại học Khoa học Tự nhiên - Đại học Quốc gia TP.HCM vào năm 2001, nhận bằng Thạc sĩ Khoa học Máy tính tại Đại học Quốc gia TP.HCM vào năm 2008. Năm 2016, nhận bằng Tiến sĩ Khoa học Máy tính tại Đại học Khoa học, Đại học Huế. Lĩnh vực nghiên cứu: xử lý ảnh, khai thác dữ liệu ảnh và tìm kiếm ảnh.
Email: thanhvt@hufi.edu.vn

Lê Mạnh Thạnh

Sinh năm 1953, nhận bằng Tiến sĩ ngành khoa học máy tính tại Đại học Budapest (ELTE), Hungary vào năm 1994. Nhận hàm Phó giáo sư tại trường Đại học Khoa học, Đại học Huế, Việt Nam vào năm 2004. Lĩnh vực nghiên cứu: cơ sở dữ liệu, cơ sở tri thức và lập trình logic.
Email: lmthanh@hueuni.edu.vn

Hệ thống lưu trữ hỗ trợ SQLite trên bộ nhớ NAND Flash

Hồ Văn Phi, Nguyễn Phương Tâm, Nguyễn Thị Hạnh, Lương Khánh Tý
Đại học Công nghệ Thông tin và Truyền thông Việt-Hàn, Đà Nẵng, Việt Nam
Tác giả liên hệ: Hồ Văn Phi, email: hvphi@vku@udn.vn
Ngày nhận bài: 28/12/2020, ngày sửa chữa: 03/04/2021, ngày duyệt đăng: 13/04/2021
Định danh DOI: 10.32913/mic-ict-research-vn.v2021.n1.938

Tóm tắt: Trong những năm gần đây, các thiết bị di động ngày càng phổ biến. Các thiết bị này thường sử dụng SQLite để làm cơ sở dữ liệu và bộ nhớ NAND Flash để lưu trữ dữ liệu. SQLite là một hệ cơ sở dữ liệu gọn nhẹ nhưng mạnh mẽ. SQLite đáp ứng đầy đủ mọi yêu cầu cơ bản của một hệ quản trị cơ sở dữ liệu. Bộ nhớ NAND Flash ngày càng phổ biến nhờ vào những ưu điểm nổi bật như tốc độ truy xuất nhanh, tiêu thụ điện năng ít, không mất dữ liệu khi mất nguồn. Do đó, việc sử dụng SQLite và NAND Flash là sự lựa chọn tốt cho thiết bị Android và iOS. Tuy nhiên, NAND Flash cũng có những nhược điểm cần khắc phục như không thể ghi đè, vòng đời bị giới hạn bởi số lần xoá trên các khối nhớ. Bên cạnh đó, cơ chế ghi tạm file journal mỗi lần commit của SQLite cũng làm ảnh hưởng đến bộ nhớ Flash. Trước khi thực hiện cập nhật dữ liệu vào cơ sở dữ liệu, SQLite tạo một file tạm gọi là journal chứa dữ liệu cập nhật trên bộ nhớ Flash. Việc này dẫn đến rất nhiều thao tác đọc/ghi trên bộ nhớ Flash làm cho hiệu suất của hệ thống giảm đáng kể. Để giải quyết vấn đề này, chúng tôi đề xuất một hệ thống bộ nhớ lai sử dụng FRAM cho hệ thống gọi là COSS(Commit-Optimized Scheme for SQLite). COSS có thể giảm bớt số lượng lớn thao tác đọc/ghi lên NAND Flash và thời gian thực hiện của cả hệ thống nhờ vào khả năng ghi đè và tốc độ cao của FRAM. Kết quả thực nghiệm cho thấy COSS đạt hiệu suất cao hơn so với hệ thống nguyên thuỷ thông thường.

Từ khóa: *SQLite, bộ nhớ Flash, FRAM, cơ sở dữ liệu điện thoại thông minh, thiết bị di động.*

Title: **A Design Method of Computational Semantics of Linguistic Words for Fuzzy Rule-based Classifier**

Abstract: Recently, the mobile devices are popular day by day. These devices use SQLite as a database management system and NAND Flash memeory as a storage medium. SQLite is a lightweight and powerful database engine. It provides all basic features of a database management system. NAND Flash memeories are popular because of the following advantages: fast access speed, nonvolatile, low power consumption. Thus, the combination of NAND Flash memory and SQLite is a good choice for mobile devices. However, NAND Flash memories also have some drawbacks that should be limited such erase-before-write characteristic, the lifecycle is limited by the number of block erasing. Beside that, the performance of NAND Flash is downgraded by temporarily writing journal files mechanism of SQLite. Whenever updating data, SQLite has to create and write the back up data to a file called Journal file. This causes a lot of Flash memory operations leading to reduction of overall performance of the system and shorten the lifecycle of NAND Flash. To address this problem, this study introduces a novel commit policy and hybrid storage system for SQLite based on a Flash memory and an FRAM, called COSS. COSS can minimize the overhead by deploying the hybrid storage of FRAM and NAND Flash memory. The experimental results show that COSS yields a good performance.

Keywords: *SQLite, Flash memory, FRAM, smart phone database, mobile devices.*

I. GIỚI THIỆU

Bộ nhớ Flash [1, 4-6] được sử dụng rất phổ biến hiện nay nhờ vào những ưu điểm nổi bật của chúng như tốc độ cao, tiêu thụ ít điện năng, kích thước nhỏ gọn và độ an toàn dữ liệu cao. Tuy nhiên, bên cạnh những điểm mạnh đó, bộ nhớ Flash vẫn có những điểm yếu cần lưu tâm đó là đặc tính xóa trước khi ghi – không thể ghi đè và vòng đời hữu hạn (khoảng từ 10.000 đến 100.000 lần xóa trên mỗi block). Bộ nhớ Flash có cấu tạo và cơ chế hoạt động hoàn toàn khác với đĩa cứng (HDD-Hard Disk Drive) thông thường. Do đó để giúp các hệ điều hành truy cập bộ nhớ Flash một cách tương tự như HDD, một phần mềm trung gian có tên là FTL (Flash Translation Layer)[1] được sử dụng để chuyển đổi địa chỉ logic sang địa chỉ vật lý ánh xạ giữa máy tính và bộ nhớ Flash.

SQLite là một hệ cơ sở dữ liệu được sử dụng phổ biến

trên các thiết bị di động sử dụng hệ điều hành Android và iOS. SQLite không cần máy chủ, không cần cấu hình, khép kín và nhỏ gọn. SQLite engine không phải là một quy trình độc lập như các cơ sở dữ liệu khác, chúng có thể liên kết một cách tĩnh hoặc động tùy theo yêu cầu của ứng dụng. SQLite truy cập trực tiếp các file lưu trữ của nó trên bộ nhớ do đó. Mặc dù hiệu suất của các thiết bị di động tăng đáng kể nhờ vào việc sử dụng SQLite và bộ nhớ Flash. Tuy nhiên việc ghi tạm file journal trên bộ nhớ Flash làm giảm đáng kể hiệu suất hoạt động do một số lượng lớn các thao tác (read/write/erase) trên bộ nhớ Flash được thực hiện mỗi khi cập nhật dữ liệu trên SQLite. Đây là một vấn đề cần được giải quyết để nâng cao hiệu suất của các hệ thống Android và iOS. Bài báo này giới một hệ thống lưu trữ kết hợp cho các ứng dụng SQLite sử dụng Flash và FRAM (Ferroelectric Random Access Memory)[2, 3, 7] gọi là COSS nhằm tối ưu hoá thao tác Commit cho các hệ thống sử dụng NAND Flash và SQLite. Mục tiêu của hệ thống này là làm giảm số lượng số lượng thao tác trên bộ nhớ Flash và làm chậm quá trình lão hóa của bộ nhớ Flash, đồng thời hệ thống này cũng giúp dữ liệu của SQLite không bị mất khi mất điện hoặc hệ thống bị treo. FRAM đóng vai trò là một bộ đệm nơi lưu trữ tạm thời các file journal của SQLite. Khi bộ nhớ FRAM đầy, tất cả các file journal trên FRAM sẽ được ghi vào bộ nhớ Flash. Khả năng ghi đè dữ liệu và ghi từng byte của FRAM giúp các ứng dụng giảm chi phí thực hiện và số lượng thao tác phụ phát sinh trên bộ nhớ Flash do các đặc tính vật lý của Flash. Kết quả thử nghiệm cho thấy hệ thống này đạt hiệu quả tốt và đảm bảo dữ liệu của SQlite luôn được an toàn. Phần còn lại của bài bào có cấu trúc như sau: Phần II giới thiệu các kiến thức căn bản và liên quan đến SQLite, Flash và FRAM. Phần III trình bày kiến trúc của hệ thống. Kết quả thực nghiệm và đánh giá hệ thống được trình bày trong Phần IV và cuối cùng, kết luận bài báo được trình bày trong Phần V.

II. KIẾN THỨC CƠ SỞ

Flash là một loại thiết bị bộ nhớ thứ cấp được sử dụng rất phổ biến hiện nay. Chúng có mặt trong các thiết bị điện tử như máy tính, điện thoại thông minh, các thiết bị nhúng, thẻ nhớ, USB, SSD,... Khác với đĩa cứng thông thường, bộ nhớ Flash gồm một dãy các thẻ nhớ NAND Flash. Bộ nhớ Flash được tổ chức thành các khối nhớ; mỗi khối nhớ chứa một số lượng cụ thể các trang nhớ (32, 64, 128 trang). Trang nhớ là đơn vị nhỏ nhất của thao tác đọc và ghi trong khi đó khối nhớ là đơn vị nhỏ nhất của thao tác xóa. Bộ nhớ Flash hỗ trợ 3 thao tác cơ bản là đọc, ghi và xóa [8-9]. Đọc là thao tác có tốc độ nhanh nhất tiếp đến là thao tác ghi. Một thao tác ghi mất khoảng 2ms; chậm hơn 10 lần so với thao tác đọc. Chậm nhất là thao tác xóa, chậm hơn 10 lần so với ghi. Hiện nay, bộ nhớ Flash có vòng đời khoảng 10.000-100.000 lần xóa khối. Nếu số lần xóa của một khối vượt quá ngưỡng này thì khối đó không còn khả năng sử dụng. Để phù hợp với các hệ điều hành khác nhau và đạt hiệu quả tối ưu, Flash cần phần mềm trung gian FTL. FTL có 3 chức năng chính đó là: ánh xạ địa chỉ (Address Mapping), thu hồi khối nhớ không hợp lệ và điều phối ghi dữ liệu trên các khối nhớ (Wear Leveling). Chức năng ánh xạ địa chỉ cung cấp các thuật toán ánh xạ giữa địa chỉ logic và địa chỉ vật lý trên bộ nhớ Flash. Đã có rất nhiều thuật toán ánh xạ địa chỉ [1, 4-6] được giới thiệu. Chúng có thể được gom thành 3 nhóm chính: Ánh xạ sector (Sector Mapping), ánh xạ khối (Block Mapping) và ánh xạ lai (Hybrid Mapping). Bộ thu hồi khối nhớ không hợp lệ có chức năng thu hồi các khối nhớ không hợp lệ để tái sử dụng. Khi số lượng khối nhớ hợp lệ trên bộ nhớ Flash thấp hơn ngưỡng cho phép, bộ thu hồi khối nhớ không hợp lệ sẽ được tự động kích hoạt để thu hồi và tái sử dụng các khối không hợp lệ. Wear Leveling được sử dụng để điều tiết các khối nhớ để đảm bảo các khối nhớ có tần suất được sử dụng là tương đương nhau. Wear leveling sẽ quyết định khối nhớ nào được sử dụng khi có yêu cầu ghi dữ liệu vào bộ nhớ Flash. FRAM là một công nghệ bộ nhớ mới sử dụng tính chất điện sắt. FRAM có các tính chất tương tự như DRAM nhưng dữ liệu lưu trên FRAM không bị mất khi không có nguồn cung cấp điện. Khác với Flash, FRAM cho phép truy xuất ngẫu nhiên đến từng đơn vị bit. Thuật ngữ “ferroelectric - điện sắt” không có nghĩa là FRAM chứa các phần tử sắt cũng không có nghĩa là FRAM bị ảnh hưởng bởi từ trường. FRAM cho phép phân vùng một cách linh hoạt cho cả mục đích chứa mã thực thi (Execution Code Partition) và dữ liệu. Có nghĩa là FRAM có thể được sử dụng để lưu trữ mã thực thi chương trình và dữ liệu đồng thời mà với các loại bộ nhớ khác phải sử dụng 2 loại riêng biệt: ROM để lưu mã chương trình và bộ nhớ ngoài lưu trữ dữ liệu. SQLite là một hệ quản trị cơ sở dữ liệu được sử dụng rộng rãi trong các thiết bị di động nhờ vào tính nhỏ gọn và mạnh mẽ của nó. Tuy nhiên, tần suất cập nhật và ghi dữ liệu ngẫu nhiên trên SQLite rất lớn nên hiệu quả của SQLite trên bộ nhớ Flash sẽ giảm đáng kể. Mỗi khi cập nhật dữ liệu, SQLite tạo một file gọi là file journal để lưu trữ tạm thời dữ liệu cập nhật. Sau khi thao tác cập nhật đã được commit, file journal sẽ bị xóa. Quá trình thực hiện atomic commit trong SQLite được thực hiện như sau:

1. Chiếm giữ khóa:

Trước khi SQLite có thể ghi dữ liệu vào cơ sở dữ liệu, nó phải nắm giữ khóa đọc (read lock) để đảm bảo cơ sở dữ liệu đã sẵn sàng. Sau đó sẽ chiếm giữ khóa chia sẻ (shared lock) để ngăn không cho giao dịch khác ghi vào mục dữ liệu hiện hành.

2. Đọc thông tin từ cơ sở dữ liệu:

Những dữ liệu cần thay đổi được đọc từ cơ sở dữ liệu.

3. Chiếm giữ khóa dành riêng (Reserved Lock)
Mục đích của việc chiếm giữ khóa reserved là nhằm ngăn chặn các giao dịch khác ghi dữ liệu và giao dịch hiện hành sẽ sớm ghi dữ liệu vào mục dữ liệu hiện hành.

4. Tạo file Journal:
Trước khi thực hiện thay đổi dữ liệu, SQLite tạo và ghi dữ liệu cũ vào file journal. Lúc này journal file được lưu tạm thời ở bộ đệm.

5. Thay đổi dữ liệu ở phía người dùng:
Sau khi dữ liệu cũ được khi vào file journal, dữ liệu được cập nhật ở phía người dùng.

6. Ghi journal file vào đĩa:
Để đảm bảo an toàn dữ liệu, file journal được ghi vào đĩa cứng.

7. Chiếm giữ khóa độc quyền (Exclusive Lock)
Trước khi ghi dữ liệu, SQLite cần chiếm giữ khóa độc quyền để đảm bảo không có giao dịch nào đang đọc hay ghi dữ liệu lên mục dữ liệu hiện hành.

8. Ghi dữ liệu thay đổi vào đĩa:
Toàn bộ dữ liệu thay đổi sẽ được ghi vào đĩa.

9. Xóa file journal:
Sau khi hoàn tất ghi dữ liệu vào file, hệ thống sẽ xóa file journal. Sau khi journal file đã bị xóa thành công, các khóa sẽ được giải phóng.

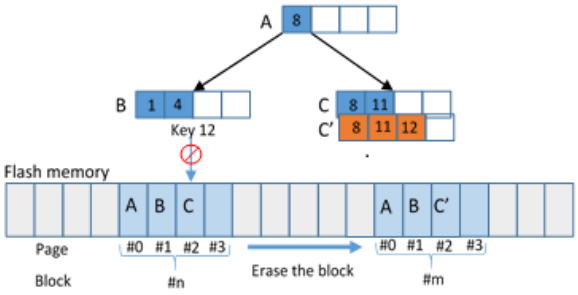
Nếu quá trình commit không thành công, hệ thống sẽ thực hiện rollback bằng cách đọc file journal vào ghi dữ liệu cũ vào cơ sở dữ liệu. Quá trình rollback bao gồm các bước như sau:

1. Tìm file journal "nóng". File journal "nóng" là file journal có các đặc điểm sau:
- Đang tồn tại hợp lệ
- Không rỗng
- Không chứa khóa reserved trên file cơ sở dữ liệu chính
- Tiêu đề đúng cấu trúc
- Không chứa tên file supper-journal (dùng trong commit nhiều giao dịch đồng thời).

2. Đọc thông dữ liệu cũ từ file journal và ghi vào cơ sở dữ liệu.

3. Xóa file journal "nóng": Sau khi tất cả mọi thông tin được ghi vào cơ sở dữ liệu, file journal cần được xóa để hoàn tất giao dịch.

Như đã đề cập ở trên, bên cạnh những điểm mạnh nổi bật, bộ nhớ Flash có hai nhược điểm lớn đó là đặc tính xóa trước khi ghi và hạn chế số lần xóa khối. Do đó, thực hiện thao tác commit của SQLite nguyên thủy trên bộ nhớ Flash sẽ dẫn đến hiệu quả hoạt động thấp. Hình 1 trình bày một ví dụ về việc cập nhật dữ liệu trên bộ nhớ Flash sử dụng thuật toán ánh xạ địa chỉ Block mapping.



Hình 1. Ví dụ cập nhật dữ liệu trên bộ nhớ Flash

Giả sử cần cập nhật trên cây chỉ mục trong cơ sở dữ liệu SQLite được lưu trữ trên bộ nhớ Flash. Nếu một bản ghi với giá trị khóa 12 được chèn vào cơ sỡ dữ liệu thì bản ghi đó được cập nhật trên nút C và ghi dữ liệu trên trang nhớ của bộ nhớ Flash. Vì đặc tính xóa trước khi ghi (không thể ghi đè) nên quá trình chèn 12 được thực hiện như sau: Đầu tiên các file hợp lệ (A, B) trong block #n được sao chép sang block #m; ghi file C' vào block #m và cuối cùng block #n sẽ bị xóa hoặc đánh dấu block không hợp lệ. Quá trình này phát sinh rất nhiều các thao tác trên Flash dẫn đến giảm hiệu suất của toàn hệ thống. Lưu ý rằng, vòng đời của NAND Flash bị giới hạn bởi số lần xoá trên các block. Do đó, việc giảm thao tác xoá giúp cho tuổi thọ của NAND Flash ít bị ảnh hưởng. Nhằm hạn chế việc phát sinh các thao tác như đã trình bày ở ví dụ trên, một giải pháp mới được đề xuất ở Phần III.

III. THIẾT KẾ VÀ CÀI ĐẶT COSS

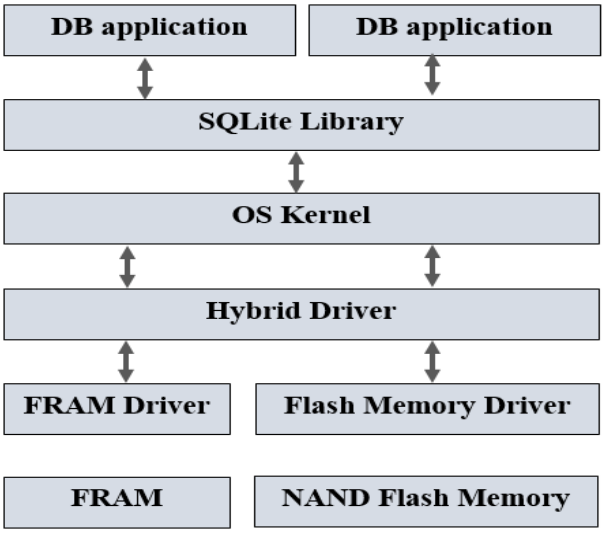
1. Thiết kế COSS

Phần này sẽ trình bày một cơ chế tối ưu hóa thao tác commit trên SQLite và hệ thống lưu trữ cho SQLite trên nền tảng NAND Flash và bộ nhớ FRAM. Mục tiêu của hệ thống này là nhằm giảm tối đa chi phí thực hiện cập nhật file journal mỗi khi cập nhật dữ liệu trong cơ sở dữ liệu SQLite và giảm số lượng thao tác trên NAND Flash. Để đạt được mục tiêu này, chúng tôi xây dựng một hệ thống lưu trữ lai sử dụng FRAM và NAND Flash. Trong hệ thống này, FRAM được dùng để lưu trữ file journal và NAND Flash được dùng để lưu trữ dữ liêu.

Hình 2 trình bày kiến trúc tổng thể hệ thống COSS. Hệ thống COSS bao gồm tất cả các thành phần của một hệ thống thông thường, driver cho hệ thống lai và các thành phần lưu trữ FRAM và NAND Flash.

Thành phần driver cho hệ thống lai sẽ quản lý và quyết định thiết bị lưu trữ nào (FRAM hay Flash) sẽ được sử dụng để ghi dữ liệu. Nếu file journal được tạo ra hoặc được cập nhật thì driver hệ thống lai sẽ ghi dữ liệu file journal lên FRAM. Ngược lại, dữ liệu của SQLite sẽ được ghi vào Flash. FRAM driver và Flash driver sẽ quản lý việc ghi dữ liệu lên FRAM và Flash tương ứng theo cơ chế của nó.

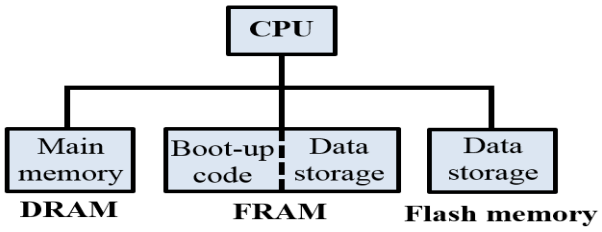
FRAM được sử dụng kết hợp với bộ nhớ Flash nhằm nâng cao hiệu suất của SQLite trên các hệ thống Android và iOS. Bởi vì dữ liệu có thể được ghi đè trên FRAM nên có thể tránh được nhược điểm xóa trước khi ghi của Flash. Mỗi khi file journal được tạo ra, COSS sẽ ghi file journal vào FRAM thay vì ghi vào Flash như bình thường. Điều này làm giảm đáng kể số lượng thao tác trên bộ nhớ Flash, đặc biệt là thao tác ghi và xóa trên Flash tiêu tốn nhiều thời gian và năng lượng.



Hình 2. Kiến trúc của COSS

Bằng cách sử dụng hệ thống lưu trữ lai này, COSS làm tăng đáng kể hiệu suất của SQLite trên các hệ thống Android/iOS của các thiết bị thông minh đồng thời làm chậm quá trình lão hoá của NAND Flash.

2. Cơ chế hoạt động của COSS

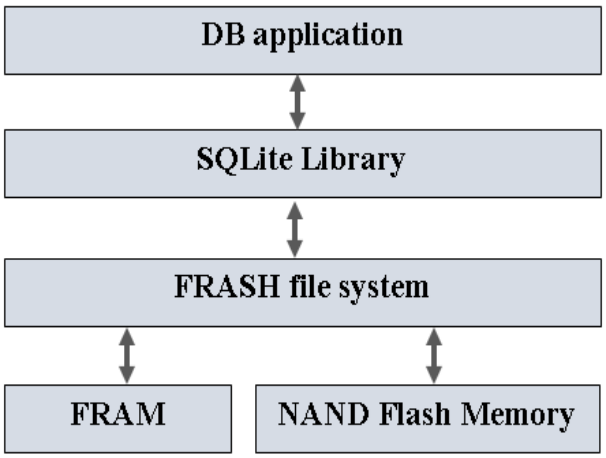


Hình 3. Hệ thống bộ nhớ kết hợp

Với hệ thống này, chúng tôi kết hợp một FRAM và một NAND Flash để lưu trữ dữ liệu. FRAM đã được sử dụng như một thiết bị lưu trữ trong các hệ thống nhúng. FRAM cho phép truy xuất từng byte và ghi đè dữ liệu do đó nó có thể hạn chế nhược điểm xóa trước khi ghi của Flash.

Như trình bày trong Hình 3, hệ thống sử dụng một FRAM cho hai mục đích là lưu trữ dữ liệu và mã thực thi chương trình. Trong trường hợp này hệ thống vẫn hoạt động tốt vì FRAM có thể đóng vai trò như cả ROM và thiết bị lưu trữ. Bằng cách sử dụng một phần của FRAM để lưu dữ liệu, hệ thống có thể lưu file journal tạm thời vào FRAM và cơ sở dữ liệu vào Flash. Với cơ chế này, hệ thống sẽ hoạt động với hiệu suất tốt hơn các hệ thống thông thường bởi vì FRAM nhanh hơn Flash, hạn chế các thao tác xóa trước khi ghi đồng thời giúp kéo dài tuổi thọ của Flash.

Để cài đặt hệ thống lưu trữ này, hệ thống cần một phương thức riêng để truy xuất cả Flash và FRAM. Như trình bày trong Hình 4, hệ thống sử dụng FRASH[10] là một hệ thống file đơn giản được thiết kế cho hệ thống lưu trữ kết hợp FRAM và Flash.



Hình 4. Hệ thống file

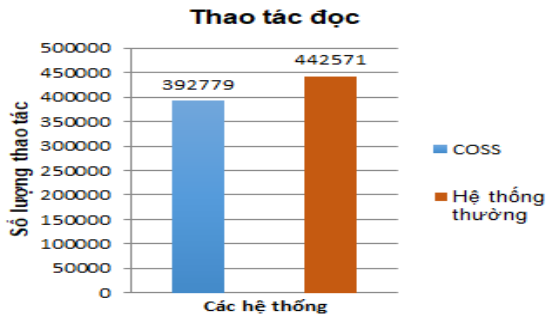
Về cơ bản, khi thực hiện quá trình commit, COSS ghi các file journal vào FRAM. Ngay khi dữ liệu được cập nhật và ghi vào Flash thành công thì các file journal trong phân vùng dữ liệu của FRAM được xóa để hoàn tất quá trình commit.

IV. ĐÁNH GIÁ HIỆU SUẤT

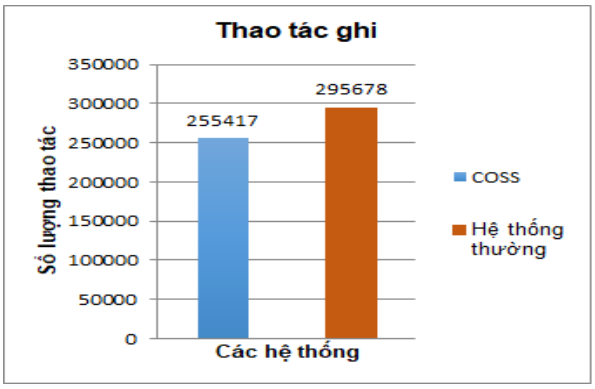
Phần này trình bày một số kết quả thực nghiệm của COSS và so sánh hiệu suất với hệ thống nguyên thuỷ. Tất cả các thực hiện được triển khai trên bộ nhớ FRAM giả lập được lấy từ RAM có dung lượng 8MB. Hiệu suất của các hệ thống được đánh giá dựa trên số lượng thao tác và hiệu suất sử dụng không gian nhớ trong các khối nhớ của NAND Flash. Trong các thử nghiệm, chúng tôi sử dụng bộ đếm thao tác và bộ đếm thời gian hệ thống của bộ nhớ Flash để đánh giá kết quả thực hiện. Để đảm bảo công bằng khi so sánh, chúng tôi thử nghiệm tất cả thực nghiệm trên cùng một môi trường là Windows 10 Pro N với vi xử lý Intel Core i5-7400, 8GB DDR và đĩa cứng SSD 512GB.

1. Hiệu suất đọc/ghi

Trong phần này, chúng tôi đánh giá hiệu quả của các hệ thống thông qua việc cập nhật 100.000 bản ghi trong cơ sở dữ liệu.



Hình 5. Số lượng thao tác đọc



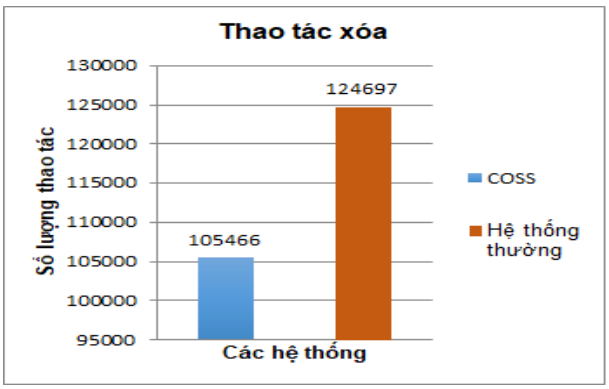
Hình 6. Số lượng thao tác ghi

Như Hình 5 đã thể hiện, số lượng thao tác đọc của COSS ít hơn khoảng 14,1% so với hệ thống nguyên thuỷ bởi vì COSS sử dụng FRAM có khả năng ghi đè. Nhờ vào khả năng ghi đè của FRAM, COSS tránh được số lượng lớn thao tác ghép bộ nhớ Flash dẫn đến một số lượng thao tác đọc được giảm bớt. Đối với thao tác ghi và thao tác xoá, COSS thực hiện ít hơn hệ thống nguyên thuỷ khoảng 17,9% và 20,5% như trong Hình 6 và Hình 7 sau đây.

Mặc dù COSS cần thêm một số thao tác ghi để ghi dữ liệu vào FRAM trước khi ghi vào Flash nhưng COSS không thực hiện các thao tác ghép trên Flash làm giảm đáng kể số lượng thao tác ghi và xoá. Điều này rất có ích cho bộ nhớ Flash bởi vì nó có thể kèo dài vòng đời của bộ nhớ Flash. Vòng đời của bộ nhớ Flash được tính dựa trên số lần xoá các khối nhớ. Do đó, việc giảm số lượng thao tác xoá trên các khối sẽ làm chậm lại quá trình lão hoá của bộ nhớ Flash.

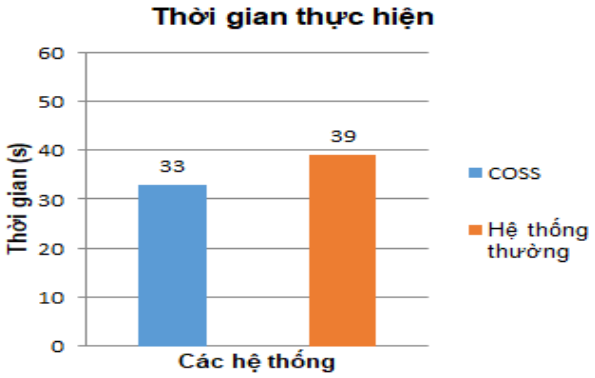
2. Thời gian thực hiện

Hình 8 trình bày thời gian thực hiện 100.000 thao tác cập nhật vào cơ sở dữ liệu. Bởi vì COSS giảm một số lượng lớn các thao tác trên Flash và tốc độ cao của FRAM nên COSS thực hiện nhanh hơn rất nhiều so với hệ thống nguyên thủy. Tốc độ truy xuất của FRAM tương tự như DRAM, nhanh hơn nhiều so với Flash. Mặt khác, không có thao tác ghép xảy ra trong FRAM - một thao tác mà tốn rất nhiều thời gian trên Flash. Điều này giúp cho COSS thực hiện cập nhật nhanh hơn khoảng 218% so với hệ thống nguyên thủy.



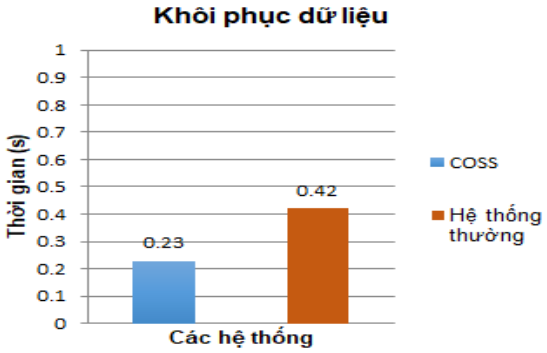
Hình 7. Số lượng thao tác xoá

3. Khôi phục dữ liệu



Hình 8. Thời gian thực hiện

Như đã trình bày ở phần trước, khả năng khôi phục dữ liệu là một điểm mạnh của FFB. Dữ liệu của COSS luôn được đảm bảo và khả năng khôi phục dữ liệu sau sự số rất nhanh. Hình 9 trình bày thời gian khôi phục dữ liệu sau sự cố. Nhìn chung, COSS khôi phục dữ liệu trong thời gian trung bình khoảng 0,23 giây; nhanh hơn hệ thống bình thường khoảng 82,6%. Bởi vì tất cả dữ liệu của COSS đều được ghi vào Flash hoặc FRAM (không mất dữ liệu) do đó dữ liệu có thể được khôi phục một cách dễ dàng bằng cách đọc từ FRAM và ghi vào Flash sau khi sự cố xảy ra.



Hình 9. Thời gian khôi phục dữ liệu

Thực tế, thời gian khôi phục dữ liệu phụ thuộc vào kích thước vùng nhớ của FRAM dùng để làm bộ đệm. Kích

thước vùng này càng lớn thì thời gian khôi phục càng lâu vì có nhiều dữ liệu lưu trữ tạm thời trong đó. Tuy nhiên, tốc độ truy xuất của FRAM khá cao nên thời gian khôi phục dữ liệu là chấp nhận được.

V. KẾT LUẬN

Bằng cách kết hợp 2 thiết bị lưu trữ để tạo thành một hệ thống lưu trữ kết hợp, COSS có thể nâng cao hiệu suất của SQLite và bộ nhớ NAND Flash trong các hệ thống di động. Trong hệ thống COSS này, với khả năng ghi đè từng byte và tốc độ cao của FRAM đã khắc phục được nhược điểm vật lý của NAND Flash đó là ghi từng trang (page) và xóa khối (block) trước khi ghi (tính chất làm ảnh hưởng xấu đến tốc độ và vòng đời của Flash). COSS không những giúp các hệ thống của sử dụng SQLite nâng cao hiệu suất mà còn đảm bảo dữ liệu luôn được an toàn. Tuy nhiên, một điểm hạn chế của COSS đó là phải sử dụng thêm một FRAM (một thiết bị lưu trữ mới chưa được sử dụng phổ biến và còn hạn chế về dung lượng).

Do đó, COSS có thể được áp dụng trong các ứng dụng mà ở đó yêu cầu về mặt tốc độ và an toàn dữ liệu đặt lên hàng đầu.

Trong tương lai gần, chúng tôi sẽ tập trung nghiên cứu xây dựng hệ thống drive cho COSS để nâng cao hiệu quả hoạt động của hệ thống và đô an toàn của dữ liệu.

TÀI LIỆU THAM KHẢO

[1] Shinde Pratibha et al. "Efficient Flash Translation layer for Flash Memory," *International Journal of Scientific and Research Publications*, Volume 3, Issue 4, April 2013

[2] Hồ Văn Phi "FFB: Hệ thống lưu trữ kết hợp cho các ứng dụng B-tree trên bộ nhớ NAND Flash". *Hội thảo quốc gia lần thứ XXI: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Thanh Hóa*, 7/2018, Trang 97-102.

[3] VanPhi Ho, Park, Dong-Joo. "An Efficient B-tree Index Scheme for Flash Memory". *International Journal of Software Engineering and Its Applications*, Vol.11, No.2 pp. 51-64, (2017)

[4] VanPhi Ho, Park, Dong-Joo, "WPCB-Tree: A Novel Flash-Aware B-Tree Index Using a Write Pattern Converter," *Symmetry* 2018, 10, 18.

[5] Chin-Hsien Wu et al. "An Efficient B-Tree Layer Implementation for Flash Memory Storage Systems," *ACM Transactions on Embedded Computing Systems*, Vol. 6, No. 3, Article 19, 2007.

[6] Xiaona Gong et al. "A Write-Optimized B-Tree Layer for NAND Flash," *Proceeding of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pp.1-4, 2011

[7] Volker Rzehak, "Low-Power FRAM Microcontrollers and Their Applications", *White Paper*, Texas Instruments Deutschland, June 2011.

[8] Yinan Li, Bingsheng He, Robin Jun Yang, Qiong Luo and Ke Yi. "Tree Indexing on Solid State Drives," *Proceedings of International Conference on Very Large Data Bases*, 2009.

[9] Hongchan Roh, Woo-Cheol Kim, Seungwoo Kim and Sanghyun Par.k "A B-Tree Index Extension to Enhance Response Time and The Life Cycle of Flash Memory," *Information Sciences*, vol. 179, no. 18, 2009, pp. 3136-3161.

[10] Kim, E.-K., Shin, H., Jeon, B.-G., Han, S., Jung, J., Won, Y. "FRASH: Hierarchical File System for FRAM and Flash". *ICCSA 2007*, Part I. LNCS, vol. 4705, pp. 238–251. Springer, Heidelberg (2007).

[11] Jung, Myoungsoo, Choi, Wonil, Shuwen Gao, Ellis Herbert Wilson III, David Donofrio, John Shalf and Mahmut Taylan Kandemir. "NANDFlashSim: High-Fidelity, Microarchitecture-Aware NAND Flash Memory Simulation". *ACM Transactions on Storage*, Vol.12, No.2, Article 6, 1.2016.

SƠ LƯỢC VỀ TÁC GIẢ

Hồ Văn Phi

Sinh ngày: 06/06/1980. Nhận bằng Thạc sỹ Khoa học máy tính năm 2009 tại Đại học Đà Nẵng. Nhận bằng Tiến sỹ KH máy tính năm 2017 tại Trường Đại học Soongsil, Hàn Quốc. Hiện công tác tại Trường ĐH CNTT và Truyền thông Việt Hàn, Đại học Đà Nẵng. Lĩnh vực nghiên cứu: Cơ sở dữ liệu trên Flash.
Điện thoại: 0905702411
E-mail: hvphi@vku.udn.vn

Nguyễn Phương Tâm

Sinh ngày: 03/06/1980. Nhận bằng Thạc sỹ Khoa học máy tính năm 2009 tại Đại học Đà Nẵng. Nhận bằng Tiến sỹ KH máy tính năm 2017 tại Trường Đại học Soongsil, Hàn Quốc. Hiện công tác tại Trường ĐH CNTT và Truyền thông Việt Hàn, Đại học Đà Nẵng. Lĩnh vực nghiên cứu: Cơ sở dữ liệu trên Flash.
Điện thoại: 0905702411
E-mail: hvphi@vku.udn.vn

Nguyễn Thị Hạnh

Sinh ngày: 01/10/1981. Nhận bằng Thạc sỹ Khoa học máy tính năm 2010 tại Đại học Đà Nẵng. Hiện đang công tác tại Trường ĐH CNTT và Truyền thông Việt Hàn, Đại học Đà Nẵng. Lĩnh vực nghiên cứu: Bảo mật dữ liệu.
Điện thoại: 0935688515.
E-mail: hanhnt@vku.udn.vn

Lương Khánh Tý

Sinh ngày: 07/08/1984. Nhận bằng Thạc sỹ KH máy tính năm 2011 tại Đại học Đà Nẵng. Hiện đang công tác tại Trường ĐH CNTT và Truyền thông Việt Hàn, Đại học Đà Nẵng. Lĩnh vực nghiên cứu: Cơ sở dữ liệu.
Điện thoại: 0974883609
E-mail: lkty@vku.udn.vn