ISSN 1870-4069

Features Used to Classify UML Diagrams from Images: a Systematic Literature Review

Juan Carlos Suárez Hernández, Ángel J. Sánchez-García, Oscar Alonso-Ramírez

Universidad Veracruzana,
Facultad de Estadística e Informática, Veracruz,
Mexico

{angesanchez, oalonso}@uv.mx, juancasu_900@hotmail.com

Abstract. This study presents the method and the result of a Systematic Literature Review to identify UML diagrams classified in a process of reusing software design artifacts, and the image characteristics used for the classification task. Twenty-one studies were selected and analyzed. In these studies, 3 types of UML diagrams, 2 types of feature approaches, and 8 classifiers were identified. As results, it was obtained that the classified UML diagrams were class diagram, sequence diagram and component diagram. The two approaches found are the predictive characteristics and the rules approach. The classifiers with the highest performance were decision tree J48, Logistic regression, Decision tables, K-Nearest Neighbor (K-NN), Random Forest, Bagging, Support Vector Machine (SVM) and Naïve Bayes.

Keywords: Classification, UML diagram, systematic literature review, software design, features.

1 Introduction

In this new revolution called Industry 4.0, software plays an important role, since it is the way in which all devices interact. Software Design is one of the most important stages in Software Engineering. This stage is characterized by the development of UML diagrams, which are the guide for developers when programming allowing to save time and effort. Just as there is code reuse, a UML diagram of one system or parts of it can be used for the redesign of another system.

Few companies and software designers use UML diagram image repositories to review previous work and be able to reuse those diagrams. Many times, when terms like "UML diagrams" in browsers are searched, search engines show many erroneous results because they search by tags and not by image content. To solve this, there are the UML diagram image classifiers that recognize if the image contains a UML diagram or not. There are some proposals for UML diagram classifiers. For example, in [1] a classifier for class diagrams is presented.

In this classifier, 23 characteristics and 6 classification algorithms were used, among which decision tree J48, logistic regression, decision tables, Random Forest, Support Vector Machine (SVM), and REP tree were found. They report a 96% accuracy rate on

Table 1. Research questions.

Question	Motivation
Q1. - What UML diagrams have been classified using images?	To know the different UML diagrams classified in the literature
Q2. - What image characteristics have been used to classify class diagrams, sequence diagrams, and communication diagrams?	To know the characteristics of the images used to classify UML diagrams
Q3. - What algorithms are used to classify class diagrams, sequence diagrams, and communication diagrams?	To know the algorithms used to classify images that contain UML diagrams

Table 2. Identified keywords and related terms.

Concept	Related terms
UML diagram	
Image	
Classification	Categorization
Characteristic	Attribute, Feature
Algorithm	
Class diagram	
Sequence diagram	
Communication Diagram	

images that were class diagrams and 91% on images that were not class diagrams. Another example is [2], where authors present a tool, which classifies UML diagrams without considering the characteristics of diagrams.

They use the analysis of grayscale histogram, color histogram, number of straight lines and rectangles among others. Such tool has to have a 95 % accuracy analyzing large sets of images reporting less than 1 second per image. This paper is organized as follows: Section II presents the method followed to carry out the Systematic Literature Reviews (SLR). Section III describes the results. Finally, Section IV draws conclusions and exposes the future work.

2 Research Method

To carry out this review, the method proposed in [3] by Kitchenham and Charters was followed, which is a guide used in SE to carry out SLR. This SLR was conducted in three phases: planning, execution of the search and report. The elements of the guide taken in each phase are presented in this section.

2.1 Planning Phase

This phase consists of the following stages: research questions, search strategy, selection of primary studies criteria and quality assessment. Next, each of the phases will be addressed.

Table 3. Search string proposed.

String	Recall	Prec.	%R	%P
Classification AND image AND algorithms AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	0.0015	0.25	0.15%	25%
Image AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	0.0005	0.5	0.05%	50%
Classification AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	0.0005	0.5	0.05%	50%
Classification AND Algorithms AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	**0.0009**	**0.5**	**0.09%**	**50%**

Table 4. Inclusion criteria.

Data Base	Description
IEEE Xplore	www.ieee.org
Springer Link	www.springer.com
ACM Digital Library	dl-acm.org
Science Direct	www.sciencedirect.com

Research Questions

Table 1 shows the research questions with their motivation to identify classified diagrams, image features, and algorithms in UML diagram classification process.

Search Terms

Table 2 shows the keywords and related terms. This search is limited and focused only on certain UML diagrams that are shown in Table 2, since these are the most used diagrams in the software industry.

Search String

With these search terms obtained from the research questions, four different search string were created. A set of papers was selected through a manual search regarding the classification of UML diagrams from different databases.

Once the possible search strings were proposed, the precision and recall were evaluated, the results of which are shown in Table 3. As a result, when observing the precision and recall of each of the options, it was concluded that the best string is the last one in Table 3 (bold).

Search Strategy

The repositories selected for the search can be seen in Table 3. This is due the access and for containing computer and engineering papers. For the selection of primary studies, inclusion and exclusion criteria were established, which can be seen in Table 5 and 6.

Selection Procedure

The selection process is made up of the following stages:

Table 5. Inclusion criteria.

IC	Description
1	Studies published between the years 2011 to 2021
2	Studies written in English
3	The title and / or abstract have at least two search terms.

Table 6. Exclusion criteria.

EC	Description
1	Do not have access to the full text
2	It is a summary, workshop, opinion piece, presentation, book or technical report
3	It is a duplicate research

Table 7. Quality assessment criteria.

ID	Criteria
QAC 1	Does the paper have the objectives of the study established?
QAC 2	Is the research process used defined?
QAC 3	Are the references less than 5 years from the publication of the paper?
QAC 4	Is the methodology used described in detail?
QAC 5	Is the main objective achieved?

- Stage 1. Primary studies are filtered according to IC1.
- Stage 2. The primary studies are removed according to the ECI and EC2.
- Stage 3. The primary studies are filtered according to the ICI2 and IC3 and the primary studies are removed according to the EC3.

Quality Assessment

Table 7 shows the format for the quality assessment, which contains 5 questions that are answered with "yes" or "no" (rated as 1 or 0 respectively). So that each study can have a final score from 0, which means very poor, to 5, which means very good.

2.2 Execution

In this section, the process of executing the SLR will be commented, talking about the selection of studies, the quality evaluation. As it can be seen in Table 8, the papers were reduced as the selection stages of primary studies were applied.

When performing stage 3, it was observed that only 21 primary studies were preserved. Table 9 shows the selected papers by data source. Figure 1 shows that only 5 papers obtained a score of 4, the other 16 papers had ratings between 2 and 3.

3 Results

The answers to the questions proposed in the planning phase will be presented below. In Fig. 2, the primary studies that passed the selection process are shown. By year of publication. It can be seen that there is a growing trend in the analysis of software design diagrams for their classification from the year 2016.

Table 8. Selection process.

Source	Search	Stage 1	Stage 2	Stage 3
IEEE Xplore	10	9	9	9
ACM Digital Library	972	533	439	10
SpringerLink	3229	2019	232	2

Table 9. Primary studies by Source.

Source	Primary studies
IEEE Explorer	[1] [4] [5] [6] [7] [8] [9] [10] [11]
ACM Digital Library	[2] [12] [13] [14] [15] [16] [17] [18] [19] [20]
SpringerLink	[21] [22]

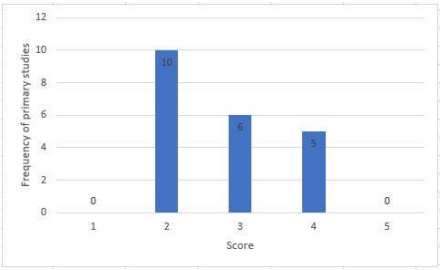

Fig. 1. Scores of quality assessment.

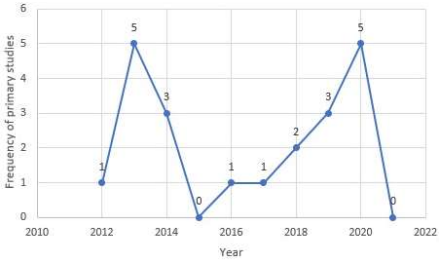

Fig. 2. Primary studies by year of publication.

3.1 Q1.- What UML Diagrams Have Been Classified Using Images

Because of the analysis of the selected papers in the SRL, which resulted in the following more classified diagrams:

- Class diagram
- Component diagram
- Diagram of sequence

As a result, they obtained 95% detection in class diagrams and 91% in non-class diagrams. In [2], authors propose an 8-rule approach to classify class and component diagrams. This approach provided 95% effectiveness classifying UML diagrams, making the best approaches so far.

3.2 Q3. - What Algorithms are Used to Classify Class Diagrams, Sequence Diagrams, and Communication Diagrams?

The algorithms used for classification in this topic are the following: J48 Decision Tree, Logistic Regression, Decision Tables, Random Forest, Support Vector Machine (SVM), REP Tree, OneR, Naïve Bayes, RBF Network, K-Nearest Neighbor (with one and five neighbors), Decision Stump, Random tree and Bagging.

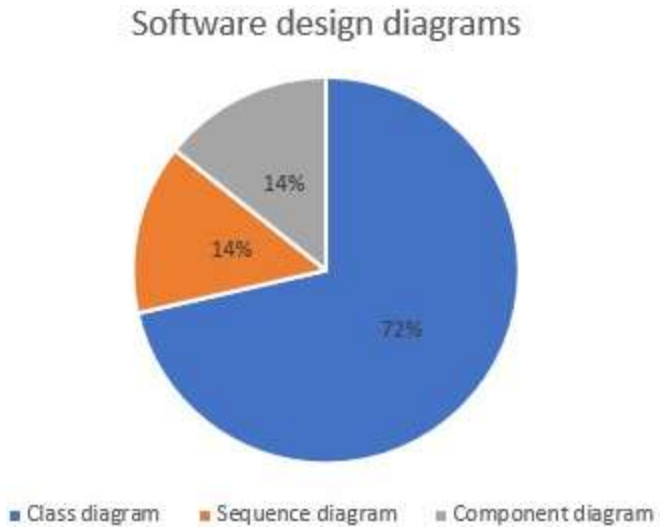

Fig. 3. Reported software design diagrams.

Table 10. Features found in primary studies.

Features in [1]	Features in [2]
Rectangles portion of images, percentage	Number of gray shades
Rectangles size variation, ratio	Number of color shades
Rectangles distribution, percentage	Number of vertical / horizontal segments.
Rectangles connections, percentage	Total number of vertical / horizontal segments at least 30 pixels long
Rectangles dividing lines, percentage	Number of horizontal segments at least 30 pixels long
Rectangles horizontally / vertically aligned, ratio	Number of vertical segments at least 30 pixels long
Average horizontal / vertical line size, ratio	Number of rectangles
Parent rectangles in parent rectangles, percentage	Number of main rectangles (not included in other rectangles)
Rectangles in rectangles, percentage	
Rectangles height-width ratio	
Geometrical shapes portion of image	
Lines connecting geometrical shapes, ratio	
Noise, percentage	
Colour frequency, percentage	

As it can be seen in Fig. 4, the most used algorithms are Decision Tables and the J48 Decision Tree with three appearances, followed by Logistic Regression, Naïve Bayes, OneR and REP Tree with two appearances each.

We must emphasize that, although they are the most used algorithms, they do not necessarily have the best results. Not much detail is given about the selection of the algorithms, more detail is given about the performance they had with the selected features.

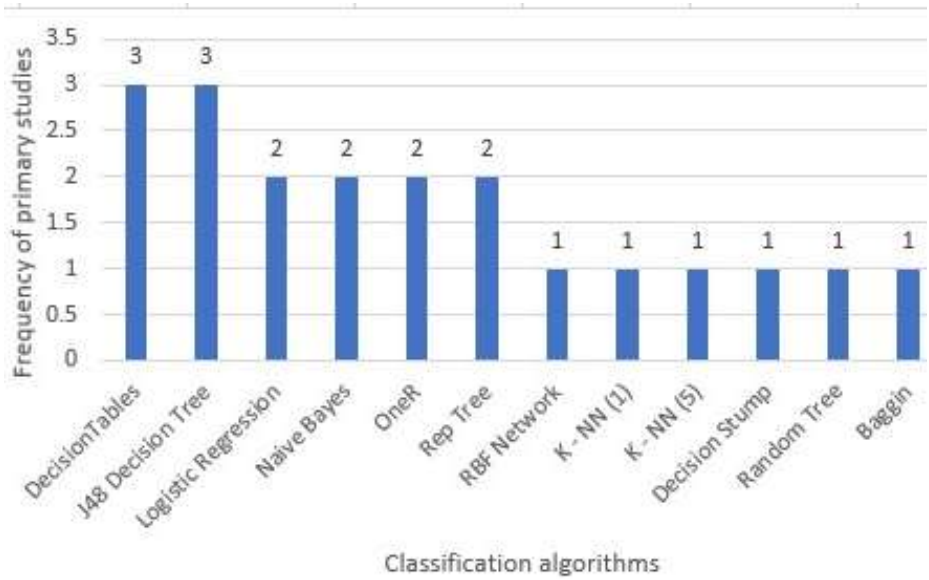

Fig. 4. Classification approaches identified.

4 Conclusions and Future Work

In this work, a Systematic Literature Review on the task of classifying software design diagrams saved in image format was performed and it was based on the method proposed in [3], where the following artifacts were obtained: the search string, the inclusion and exclusion criteria, and the data extraction and quality assessment.

The UML diagrams, when performing the analysis to the data extraction artifacts, the most classified UML diagrams were class diagrams, sequence diagrams and component diagrams.

The best-performing characteristics described by Ho-Quang in [1], as well as Moreno's rule features [2], which will be used together with the UML diagram classification algorithms, to measure performance with the different approaches in our future work.

The classification algorithms such as J48 decision tree, Logistic regression, Decision tables, K-NN (with one neighbor), Random Forest, Bagging, Support Vector Machine, and Naïve Bayes were the best at classifying UML diagram images as reported in [1].

However, we decided to select the following for the next stage of the reception work: J48 decision tree, Logistic regression and SVM. This because they show better results in this review.

Finally, as future work it is intended to extend the functionalities of the work proposed in [2], using a combination of reported algorithms and extending the number of classes in: Class Diagram, Component Diagram, Sequence Diagram and another Diagram.

References

1. Ho-Quang, T., Chaudron, M. R., Samuelsson, I., Hjaltason, J., Karasneh, B., Osman, H.: Automatic classification of UML class diagrams from images. In: Proceedings of Asia-Pacific Software Engineering Conference, vol. 1, pp. 399–406 (2014) doi: 10.1109/APSEC.2014.65
2. Moreno, V., Génova, G., Alejandres, M., Fraga, A.: Automatic classification of web images as UML diagrams. In: Proceedings of ACM International Conference Proceeding Series (2016) doi: 10.1145/2934732.2934739
3. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
4. Osman, M. H., Chaudron, M., Van Der Putten, P.: An analysis of machine learning algorithms for condensing reverse engineered class diagrams. In: Proceedings of IEEE International Conference on Software Maintenance, pp. 140–149 (2013) doi: 10.1109/ICSM.2013.25
5. Godara, D., Singh, R. K.: Improving change proneness prediction in UML based design models using ABC algorithm. In: Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, pp. 1296–1301 (2014) doi:10.1109/ICACCI.2014.6968320
6. Osman, M. H., Ho-Quang, T., Chaudron, M. R.: An automated approach for classifying reverse-engineered and forward-engineered UML class diagrams. In: Proceedings of 44th Euromicro Conference on Software Engineering and Advanced Applications pp. 396–399 (2018) doi: 10.1109/SEAA.2018.00070
7. Jindal, S., Khurana, G.: The statistical analysis of source-code to determine the refactoring opportunities factor (ROF) using a machine learning algorithm. In: Proceedings of IET Conference Publications, pp. 396–403 (2013) doi: 10.104 9/cp.2013.2244
8. Salami, H. O., Ahmed, M.: Class diagram retrieval using genetic algorithm. In: Proceedings of the 12th International Conference on Machine Learning and Applications, vol. 2, pp. 96–101 (2013) doi: 10.1109/ICMLA.2013.112
9. Narawita, C. R., Vidanage, K.: UML generator - An automated system for model driven development. In: Proceedings of 16th International Conference on Advances in ICT for Emerging Regions, pp. 250–256 (2017) doi: 10.1109/ICTE R.2016.7829928
10. Osman, M. H., Chaudron, M. R., Van Der Putten, P., Ho-Quang, T. (2014) Condensing reverse engineered class diagrams through class name based abstraction. In: Proceedings of 4th World Congress on Information and Communication Technologies, WICT 2014, 158–163, (2014) doi: 10.1109/WIC T.2014.7077321
11. Iqbal, S. Z., Huzam, D., Aleliwi, N., Alabdulkareem, A., Alalyani, S., Alfaris, S., Gull, H.: Smart fair guide: requirements and design of a smart fair management system. In: Proceedings of 3rd International Conference on Computer Applications and Information Security, pp. 1–7 (2020) doi: 10.1109/ICCAIS48893.202 0.9096722
12. Lopes, A., Steinmacher, I., Conte, T.: UML acceptance: Analyzing the students' perception of UML diagrams. In: Proceedings of ACM International Conference Proceeding Series, pp. 264–272 (2019) doi: 10.1145/3350768.3352575
13. Balaban, M., Maraee, A.: Finite satisfiability of UML class diagrams with constrained class hierarchy. ACM Transactions on Software Engineering and Methodology, vol. 22, no. 3, pp. 1–42 (2013) doi: 10.1145/2491509.2491518
14. Bian, W., Alam, O., Kienzle, J.: Is automated grading of models effective?: Assessing automated grading of class diagrams. In: Proceedings of 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 365–376 (2020) doi: 10.1145/3365438.3410944

15. Qi, K., Boehm, B. W.: Detailed use case points (DUCPs): A size metric automatically countable from sequence and class diagrams. In: Proceedings of International Conference on Software Engineering, pp. 17–24 (2018) doi: 10.1145/ 3193954.3193955
16. Saxena, V., Arora, D., Mishra, N.: UML modeling of load optimization for distributed computer systems based on genetic algorithm. ACM SIGSOFT Software Engineering Notes, vol. 38, no. 1, pp. 1–7 (2013) doi: 10.1145/24130 38.2413043
17. Nurwidyantoro, A., Ho-Quang, T., Chaudron, M. R.: Automated classification of class role-stereotypes via machine learning. In: ACM International Conference Proceeding Series, pp. 79–88 (2019) doi: 10.1145/3319008.3319016
18. De Oliveira Barbosa, M., Ramalho, F.: An approach to identify and classify state machine changes from code changes. In: ACM International Conference Proceeding Series, pp. 111–120 (2020) doi: 10.1145/3425269.3425282
19. Mannava, V., Ramesh, T.: Multimodal pattern-oriented software architecture for self-optimization and self-configuration in autonomic computing system using multi objective evolutionary algorithms. In: ACM International Conference Proceeding Series, pp. 1236–1243 (2012) doi: 10.1145/2345396.2345595
20. Relucio, F. S.: Unified modeling and framework design on procurement data standards implementation. In: ACM International Conference Proceeding Series, pp. 126–130 (2020) doi: 10.1145/3379247.3379281
21. Ott, J., Atchison, A., Linstead, E. J.: Exploring the applicability of low-shot learning in mining software repositories. Journal of Big Data, vol. 6, no. 1 (2019) doi: 10.1186/s40537-019-0198-z
22. Best, N., Ott, J., Linstead, E. J.: Exploring the efficacy of transfer learning in mining image-based software artifacts. Journal of Big Data, vol. 7, no. 1 (2020) doi: 10.1186/s40537-020-00335-4