Proposal of a similarity measure for unified modeling language class diagram images using convolutional neural network

Rhaydae Jebli, Jaber El Bouhdidi, Mohamed Yassin Chkouri
SIGL Laboratory, ENSA, Abdelmalek Essaadi University, Tetuan, Morocco

Article Info

Article history:

Received Sep 21, 2023
Revised Nov 30, 2023
Accepted Dec 26, 2023

Keywords:

Convolutional neural network
Deep learning
Similarity assessment
Similarity measure
Unified modeling language class diagram

ABSTRACT

The unified modeling language (UML) represents an essential tool for modeling and visualizing software systems. UML diagrams provide a graphical representation of a system's components. Comparing and processing these diagrams, for instance, can be complicated, especially as software projects grow in size and complexity. In such contexts, deep learning techniques have emerged as a promising solution for solving complex problems. One of these crucial problems is the measurement of similarity between images, making it possible to compare and calculate the differences between two given diagrams. The present work intends to build a method for calculating the degree of similarity between two UML class diagrams. With a goal to provide teachers a helpful tool for assessing students' UML class diagrams.

Corresponding Author:

Rhaydae Jebli
SIGL Laboratory, ENSA, Abdelmalek Essaadi University
Tetuan, Morocco
Email: rhaydaejeb@gmail.com

1. INTRODUCTION

Assessment of learning occupies an integral part of education, it serves numerous goals, including measuring a student's progress, it determines whether the student achieves the specific learning outcomes. It can help them determine their own strengths and weaknesses, and then improve their performance. Moreover, it plans future steps for the improvement of learning by collecting and analyzing the development of the learner [1]. The assessment of students' unified modeling language (UML) class diagrams is characterized by complexities due to the resort nature of their visual representations. UML class diagrams encapsulate four essential aspects: class structure, relationships, behavior, and interfaces, requiring a comprehensive assessment approach. Manual assessment methods, traditionally used to measure students' UML diagrams, are time-consuming and often subjective, underlining the need for objective and effective assessment techniques.

In this context, the unique challenges associated with UML class diagrams have led to interest in deep learning approaches. Deep learning, a sub-field of artificial intelligence (AI), enables complex patterns and relationships to be identified within complex data sets. Recently, AI has become a reality, and most of its applications are constantly improving progress in all aspects of education and the learning process [2]. It can potentially take over everyday tasks quickly while making recommendations on how to close learning gaps. Artificial intelligence has started developing novel teaching and learning solutions, which are currently being tested in a range of environments. It brings several interesting perspectives to the automated assessment of student assignments.

In the area of comparing and assessing UML diagrams, some previous research has been proposed to automate the assessment of students' UML class diagrams using different approaches. These include the work [3], which proposes an automated grading approach for UML class diagrams. The approach employs a grading algorithm that compares a student's answers to a template solution using syntactic, semantic, and structural matching. Syntactic matching has been computed through a Levenshtein distance, semantic matching has been computed using the WordNet database, and structural matching covers property similarity. The results show that the tool automatically scored 20 students, with a 14% difference from the score received by the instructor. Stikkolorum *et al.* [4] discusses a research study on the use of machine learning for grading UML class diagrams. A regression model and several classification models were trained such as the random forest algorithm, and a selection of features e.g., class name, attribute name, multiplicity value, and UML element types were chosen and used as elements for the machine learning experiments. The results demonstrate that classification with trained data is approaching accuracy, but is not yet sufficient (69%). Al-Khiaty and Ahmed [5] proposes a greedy-based algorithm for matching UML class diagrams using different similarity metrics namely lexical, internal, and neighborhood similarity, as well as combinations of these metrics. The results of this study illustrate that the proposed algorithm outperforms the simulated annealing algorithm in terms of matching accuracy and computation time. The authors highlight the relevance of similarity measures combined with appropriate weights to improve matching accuracy. There have been also contributions to automated assessment of different types of UML models, such as the work [6], that introduce an architecture of automated assessment of use case diagram, providing students with direct feedback in terms of quantitative and qualitative measures. This study presents a new approach to the evaluation of use case diagrams, treating them as mixed graphs and simultaneously taking into account the syntactic and semantic aspects of the labels. The architecture comprises two key sub-steps, the graph generator and the evaluation agent.

Existing literature highlights the need for solutions adapted to deal with the complex features of UML class diagrams during the process that determines the similarity assessment. Recognizing this gap, in the current research landscape, our research attempts to build on it by suggesting an approach based on deep learning methodologies that is explicitly designed for the challenges presented by UML class diagrams, recognizing their complexity and distinctive attributes. Through the integration of deep learning techniques, we aim to improve the accuracy and efficiency of similarity assessments in the context of UML class diagrams, answering a critical need in the field.

We aim to tackle the problem of measuring the similarity between two UML class diagrams, a task that proves particularly challenging given the multiple aspects of these diagrams. The core problem revolves around the assessment required for UML class diagrams, going beyond superficial visual analysis. Deep learning techniques, in particular convolutional neural networks (CNNs), emerge as a potential solution due to their ability to learn complex hierarchical features from visual data and are widely used for image recognition and processing [7]. By training CNNs on annotated datasets specific to UML class diagrams, our research focuses on developing a model capable of recognizing complex features that define success and achievement in the realization of these diagrams.

The present work highlights the use of convolutional neural networks as a potential method for assessing students' UML class diagrams objectively quantifying the degree of similarity between diagrams, and detecting potential errors within student assignments. After training our model to learn meaningful features from a set of UML class diagram images, we propose a similarity measure based on the Euclidean distance between the image of the UML class diagrams produced by the students and the image of the UML class diagram provided by the teacher as a solution and then obtain the degree of similarity between them. Using CNNs and this similarity measure based on Euclidean distance, our methodology aims to improve the assessment of the student's UML class diagrams.

This paper is organized as follows: in section 2, we detail our research method, which includes an exploration of deep learning architectures, and the methodology employed to measure similarity using CNN architecture. Section 3 is dedicated to presenting our results and providing further discussion. Finally, section 4 concludes the paper by summarizing the main results and suggesting potential future research directions in the field of UML class diagram assessment.

2. METHOD

In this proposal, we have adopted deep learning, particularly the CNN model to study the possibility of measuring the similarity of UML class diagrams saved as images. CNNs are particularly well suited to image processing and can capture important features and patterns using convolution and pooling layers. Before describing and presenting our proposed method, we will begin by exploring different deep learning algorithms, including the recurrent neural network, the CNN, and the long short-time memory (LSTM).

2.1. Deep learning algorithms

Finding similarities between images is an active area of research and a difficult task for machines. However, deep learning models have enabled significant progress and remarkable results in computer vision in many real-world applications, making them a powerful tool for tackling the challenges associated with image similarity assessment, including in the domain of UML class diagrams. As part of our research methodology, we are exploiting these advances in deep learning to propose an approach adapted to the unique characteristics of UML class diagrams.

Deep learning is a sort of machine learning, inspired by human brain activity, it has progressively the most frequently used computational approach in the area of machine learning [8], and it consists of a set of algorithms for modeling high-level abstractions of data [9], as well as analyzing and learning huge volumes of unstructured data such as images, sounds, and text. Avoiding the necessity for manual feature extraction, deep learning algorithms are trained through neural network architectures which learn to automatically extract the features at different levels of abstraction from the input to carry out a certain job [10]. The neurons in this network are arranged in dozens or even hundreds of layers, and each one receives and processes information from the layer before it. The deep learning field covers a wide range of algorithms, each one designed to solve a specific task. Some of these algorithms are listed below [11].

2.1.1. Convolutional neural network (CNN)

The convolutional neural network (CNN) is a multilayer network that was biologically inspired by the animal visual cortex, and it was first evolved and created by LeCun *et al.* [12]. CNN is mainly designed for data like images [13], and for understanding visual content [14]. CNN's architecture is based on 5 types of layers: convolution layer, pooling layer, rectified linear unit (ReLU) correction layer, fully connected layer, and output layer. Early layers recognize features, and later layers recombine these features into higher-level attributes of the input. In practice, CNNs are better adapted for data sets with many nodes and parameters to be analyzed. The neurons in a CNN are arranged in three dimensions width, height, and depth [15].

2.1.2. Recurrent neural network (RNN)

A recurrent neural network (RNN) is another architecture of deep learning that was designed to model sequences of data such as speech and language. The main feature of a recurrent neural network is its memory, which enables it to retrieve all the data stored sequentially in the previous element. This means that RNNs are capable of exploiting data in a fairly long sequence [16]. RNNs are made up of a series of recurrent layers that are sequentially modeled to map the sequence toward other sequences.

2.1.3. Long short-time memory (LSTM)

Long short-time memory (LSTM) can be considered as an extended version of RNN that can learn and memorize long-term dependencies and handle the vanishing gradient problem [17]. In the hidden layer of LSTM, special units are used in addition to standard units known as memory blocks that contain memory cells that store the network's temporal state [18]. LSTMs retain information over time. They are useful in time-series prediction because they remember previous inputs [19], and it has been used widely to solve various problems in natural language processing [20].

2.2. Architecture

The similarity is by principle a complex notion, and it is not the easiest thing for a predictive computer program to deal with. The application of different CNN models and architectures has gotten a lot of attention in the recent few days, in various computer vision tasks such as image classification, image segmentation, and object detection, and it has shown great ability in the task of measuring image similarity. This study proposes a method to assess the student's UML class diagram. Since images are one of the most used ways of storing and sharing UML diagrams, we aim in this paper to measure the degree of similarity between the student UML class diagram and the teacher UML class diagram as a reference. Figure 1 presents the global architecture of our platform for assessing student UML class diagrams.

Our model, which is based on a CNN, is made up of a number of layers, all of which serve a particular purpose through the image-processing stage. Input layers, convolutional layers, activation function layers, batch normalization (BN) layers, pooling layers, and fully connected (FC) layers are mainly the components that are frequently employed in a CNN model [21]. The input is a UML class diagram image of size (256, 256) with three channels. Our model uses five convolutional layers, the first of which is made up of 32 filters of size 3×3, the second of which is made up of 64 filters of size 3×3, the third of which is made up of 128 filters of size 3×3, and so on. Each convolutional layer is followed by a batch normalization and a non-linear activation function, such as the ReLU function. After a few convolution layers, we also add pooling layers to reduce the spatial dimension of the feature maps [22]. Finally, at the output of the convolution process, we used a dense layer, often called the fully connected layer, in which each neuron is connected to all the neurons in the previous layer. Then we used the embedding metric layer between the embeddings generated by the model. The metric used is the Euclidean distance, which is a measure of similarity or dissimilarity between two vectors [23]. Figure 2 shows an example of the CNN architecture with the different layers used.

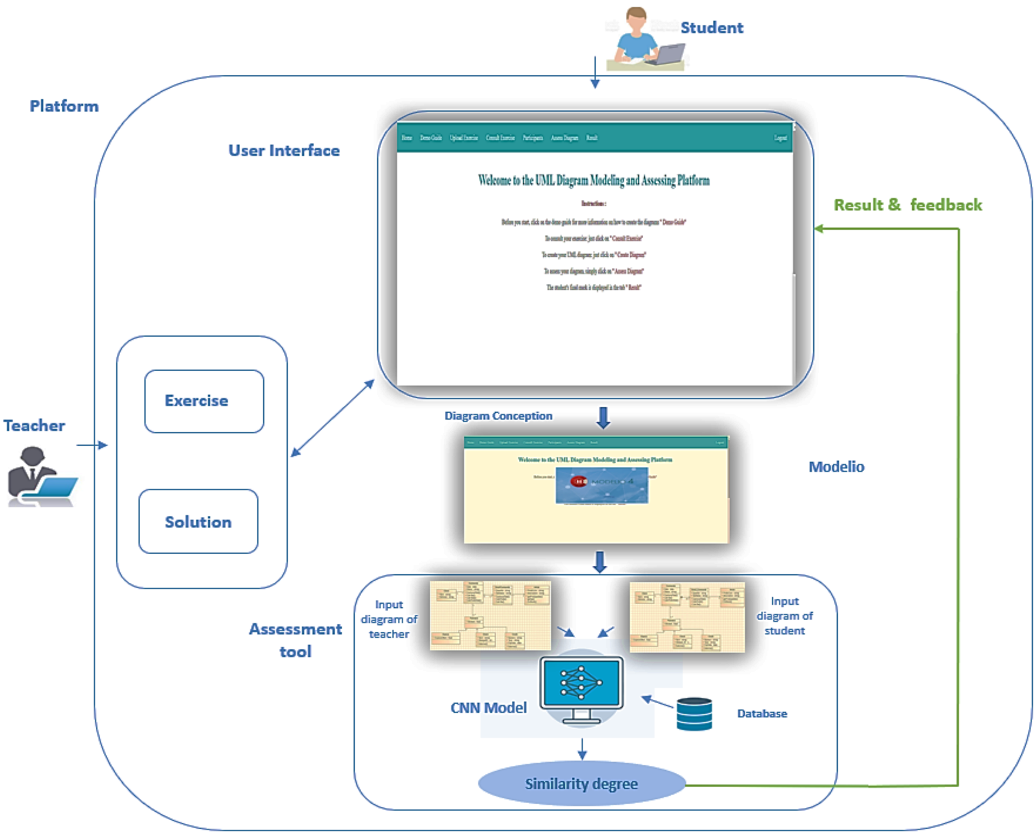

Figure 1. Platform architecture for assessing UML class diagram using CNN model

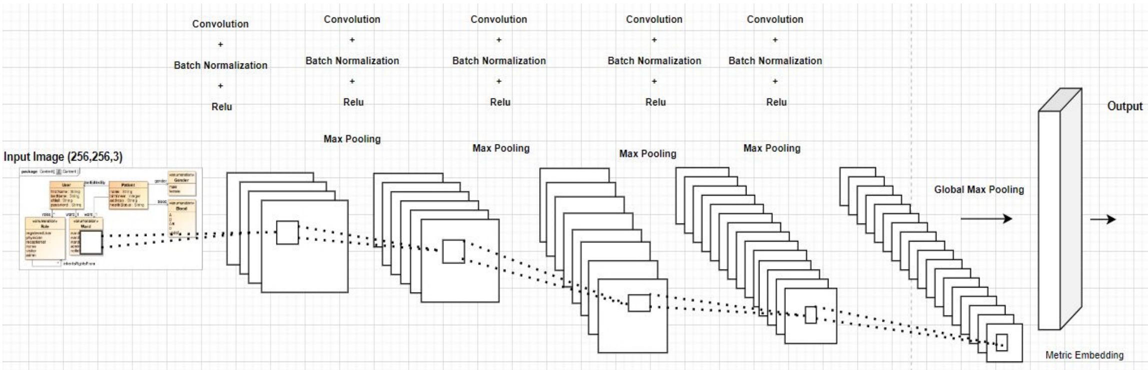

Figure 2. Our CNN model

2.3. Algorithm

In this study, we introduce an algorithm in Figure 3 for assessing the similarity between UML class diagrams, exploiting deep learning methodologies. The main component of our approach is based on the use of a CNN architecture. This CNN model is designed to learn the complex patterns and features in UML class diagram images. Pre-processed using standard techniques such as normalization, the images serve as input to the CNN, enabling it to identify differences and similarities.

Algorithm: Automatic comparison of UML Class Diagrams

Inputs: ImageUMLClassDiagram1, ImageUMLClassDiagram2
Output: ScoreSimilarity
Begin:
    Function Preprocess(image)
    Training_Set, Validation_Set = Split_Sets(dataSet)
    # Function to create a convolutional neural network model:
    CNNModel = CreateConvolutionalModel()
    TrainModel(CNNModel, Training_Set)
    Prediction of similarity with the trained model
    # Calculate Euclidean distance between predictions
    ScoreSimilarity = CalculateEuclideanDistance(PredictionImage1, PredictionImage2)
        If ScoreSimilarity is 1, then
            Match
        Else
            Return (ScoreSimilarity)
        End If
    Return ScoreSimilarity
End

Figure 3. Algorithm of UML diagram assessment

2.4. Data

For this study, we use UML class diagram images that were collected from both: a part of the dataset of the paper [24], and some that were accessible online. We finalized a list of 1,476 UML class diagram images. We use a validation division of 80% (1,181 images) of the images for training allowing the model to learn patterns and features from a varied set of UML class diagrams, and 20% (295 images) for validation allowing an unbiased assessment of the model's generalization performance. This strategy of data preservation and division is designed to guarantee the model's competence in recognizing and evaluating the similarity of UML class diagrams in various scenarios.

3. RESULTS AND DISCUSSION

Our research objective is to find a solution to assess automatically the student's UML diagrams. The solution will be supported by a developed web platform in which students are engaged as active participants in the assessment process, allowing them to achieve skills in examining their learning and becoming self-directed learners [25]. We conducted a severe set of experiments to test our methodology, Figure 4 presents the class diagram of a teacher as reference (left) and the class diagram of a student (right) containing an omission of an operation.

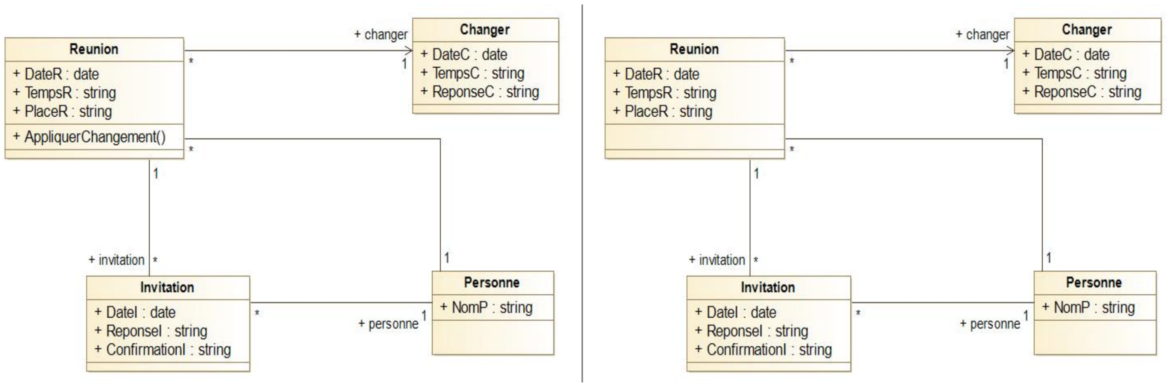

Figure 4. Class diagram of teacher (left) and student 1 (right)

The degree of similarity measured by our model is in Figure 5. The result illustrates the degree of similarity in the case of an omitted operation. Figure 5 displays the model's ability to identify and quantify the level of similarity between the UML class diagram generated by the student and the reference solution, focusing on its ability to detect and highlight variances such as omitted operations. Figure 6 presents the class diagram of a teacher as reference (left) and the class diagram of a student (right). This visual representation is specifically designed to highlight a notable difference in the student's diagram, namely the omission of two attributes.

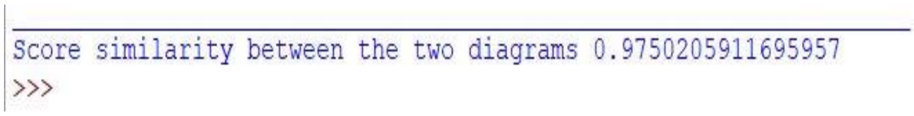

Figure 5. Degree of similarity between teacher UML class diagram and student class diagram

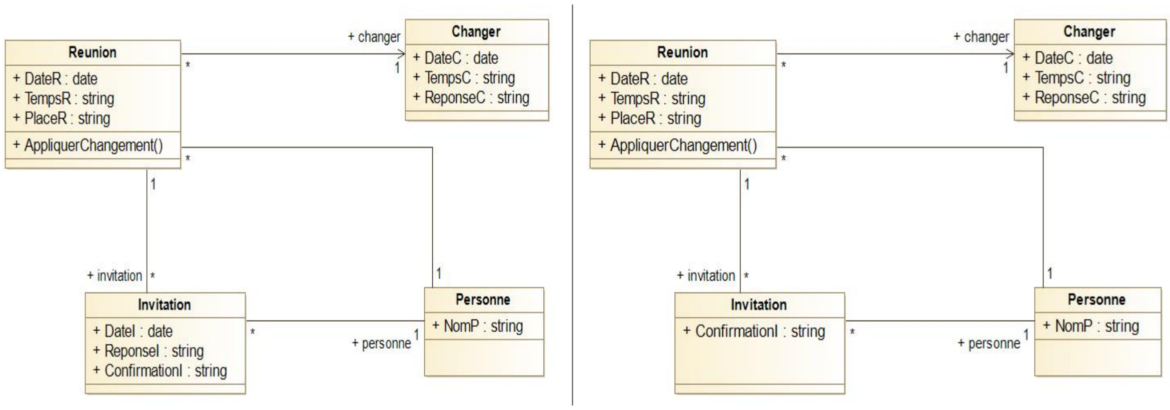

Figure 6. Class diagram of teacher (left) and student 1 (right)

The degree of similarity measured by our model is in Figure 7. Our model specifically defines the degree of similarity in the case of an omission of two attributes in the student's UML class diagram. The results provide an overview of how the model can respond in scenarios involving specific types of error, such as attribute omissions, in UML class diagrams.

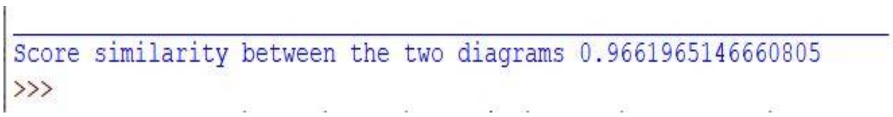

Figure 7. Degree of similarity between teacher UML class diagram and student class diagram

From these examples and many others of UML class diagrams evaluated and compared by our CNN model, we have observed that our method is effective for measuring the degree of similarity between two UML class diagrams, depending on the potential differences that can be identified between the UML class diagram of a teacher and the student UML class diagram, such as the omission of an element (attribute, operation), the difference in a class name, the difference in a relationship (aggregation, composition, and generalization). However, it is important to recognize certain limitations, such as the fact that variations in the spatial positioning of entities in UML class diagrams can lead to certain errors. As for future work, several promising directions are envisaged to improve the capabilities and applicability of our proposed deep learning algorithm for UML class diagram similarity assessment, such as expanding the dataset represents a key step toward improving the accuracy and robustness of our algorithm, and addressing the identified limitation related to spatial variations in entity positioning presents an opportunity for improvement.

4. CONCLUSION

Our research into the assessment of UML class diagrams using CNNs represents a significant advance in the field of education and software modeling. The method used in this paper is based on a deep learning approach namely a convolutional neural network, it was trained on images of UML class diagrams to learn to automatically extract the features from images and then it will be able to capture the similarity between these images, and provide a score of similarity. Our study has demonstrated the ability of CNNs to automate the assessment of UML diagrams objectively, offering a valuable way of saving time for educators.

REFERENCES

[1] R. Bourke, "Self-assessment to incite learning in higher education: developing ontological awareness," *Assessment and Evaluation in Higher Education*, vol. 43, no. 5, pp. 827–839, 2018, doi: 10.1080/02602938.2017.1411881.

[2] J. Shin, F. Chen, C. Lu, and O. Bulut, "Analyzing students' performance in computerized formative assessments to optimize teachers' test administration decisions using deep learning frameworks," *Journal of Computers in Education*, vol. 9, no. 1, pp. 71–91, Mar. 2022, doi: 10.1007/S40692-021-00196-7.

[3] W. Bian, O. Alam, and J. Kienzle, "Automated grading of class diagrams," in *Proceedings - 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion, MODELS-C 2019*, 2019, pp. 700–709, doi: 10.1109/MODELS-C.2019.00106.

[4] D. R. Stikkolorum, P. Van Der Putten, C. Sperandio, and M. R. V. Chaudron, "Towards automated grading of UML class diagrams with machine learning," in *CEUR Workshop Proceedings*, 2019, vol. 2491, pp. 1–13.

[5] M. A.-R. Al-Khiaty and M. Ahmed, "Similarity assessment of UML class diagrams using a greedy algorithm," in *2014 International Computer Science and Engineering Conference (ICSEC)*, Jul. 2014, pp. 228–233, doi: 10.1109/ICSEC.2014.6978199.

[6] V. Vachharajani and J. Pareek, "A proposed architecture for automated assessment of use case diagrams," *International Journal of Computer Applications*, vol. 108, no. 4, pp. 35–40, 2014.

[7] B. F. Ali B H and P. Ramachandran, "Compressive domain deep CNN for image classification and performance improvement using genetic algorithm-based sensing mask learning," *Applied Sciences (Switzerland)*, vol. 12, no. 14, 2022, doi: 10.3390/app12146881.

[8] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[9] R. Vargas, A. Mosavi, and R. Ruiz, "Deep learning: A review," *Advances in Intelligent Systems and Computing*, vol. 5, Jun. 2017.

[10] S. Indolia, A. Kumar, S. P. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network- a deep learning approach," *Procedia Computer Science*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[11] D. Shetty, C. A. Harshavardhan, M. J. Varma, S. Navi, and M. R. Ahmed, "Diving deep into deep learning: History, evolution, types and applications," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 3, pp. 2835–2846, Jan. 2020, doi: 10.35940/ijitee.A4865.019320.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278–2324, doi: 10.1109/5.726791.

[13] Z. Wang and S. Ji, "Learning convolutional text representations for visual question answering," in *Proceedings of the 2018 SIAM International Conference on Data Mining*, 2017, pp. 594–602, doi: 10.1137/1.9781611975321.67.

[14] D. Bhatt *et al.*, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics (Switzerland)*, vol. 10, no. 20. Oct. 2021, doi: 10.3390/electronics10202470.

[15] N. Ganatra and A. Patel, "A comprehensive study of deep learning architectures, applications and tools," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 12, pp. 701–705, Dec. 2018, doi: 10.26438/ijcse/v6i12.701705.

[16] R. Zhu, X. Tu, and J. Xiangji Huang, "Deep learning on information retrieval and its applications," in *Deep Learning for Data Analytics*, H. Das, C. Pradhan, and N. Dey, Eds. Elsevier, 2020, pp. 125–153.

[17] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, pp. 1–20, 2021, doi: 10.1007/s42979-021-00815-1.

[18] D. Lee *et al.*, "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus," *China Communications*, vol. 14, no. 9, pp. 23–31, 2017, doi: 10.1109/CC.2017.8068761.

[19] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, May 2021, doi: 10.1007/s42979-021-00592-x.

[20] S. Minaee and Z. Liu, "Automatic question-answering using a deep similarity neural network," in *2017 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2017 - Proceedings*, 2018, vol. 2018-Janua, pp. 923–927, doi: 10.1109/GlobalSIP.2017.8309095.

[21] G. Haskins *et al.*, "Learning deep similarity metric for 3D MR–TRUS image registration," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 3, pp. 417–425, Mar. 2019, doi: 10.1007/S11548-018-1875-7.

[22] A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, "Applications of convolutional neural networks," in *International Journal of Computer Science and Information Technologies*, 2016, vol. 7, no. 5, pp. 2206–2215.

[23] S. Appalaraju and V. Chaoji, "Image similarity using deep CNN and curriculum learning," *arXiv preprint arXiv:1709.08761*, pp. 1–9, 2017.

[24] S. Shcherban, P. Liang, Z. Li, and C. Yang, "Multiclass classification of UML diagrams from images using deep learning," *International Journal of Software Engineering and Knowledge Engineering*, vol. 31, no. 11–12, pp. 1683–1698, Jan. 2021, doi: 10.1142/S0218194021400179.

[25] OECD Reviews of Evaluation and Assessment in Education, "Student assessment: Putting the learner at the centre," in *Synergies for Better Learning: An International Perspective on Evaluation and Assessment*, 2013, pp. 139–269.

BIOGRAPHIES OF AUTHORS

Rhaydae Jebli is a PhD student at the SIGL Laboratory, University Abdelmalek Essaadi, National School of Applied Sciences Tetouan-Morocco. She holds a master's degree (2017) from the University Abdelmalek Essaadi and BSc degree (2015) from the University Mohamed First. Her research interests are educational technologies, e-learning environments, and machine learning. She can be contacted at email: rhaydaejeb@gmail.com.

Jaber El Bouhdidi is an HDR professor of computer science at the SIGL Laboratory, University Abdelmalek Essaadi, National School of Applied Sciences, Tetouan, Morocco. His research interest includes web semantic, multi-agents' systems, e-learning adaptive systems and big data. He has several papers in international conferences and journals. He can be contacted at email: jaber.elbouhdidi@uae.ac.ma.

Mohamed Yassin Chkouri is a professor of Computer Science at the SIGL Laboratory, University Abdelmalek Essaadi, National School of Applied Sciences, Tetouan, Morocco, he is head of Computer Engineering Department, SIGL Research Laboratory Director and Academic coordinator Erasmus + e-VAL project. He can be contacted at email: mychkouri@uae.ac.ma.