



ORIGINAL

Toward Innovative Recognition of Handwritten Arabic Characters: A Hybrid Approach with SIFT, BoVW, and SVM classification

Hacia un reconocimiento innovador de caracteres árabes manuscritos: Un enfoque híbrido con SIFT, BoVW y clasificación SVM

Othmane Farhaoui¹ , Mohamed Rida Fethi¹ , Imad Zeroual¹ , Ahmad El Allaoui¹ 

¹L-STI, T-IDMS, FST Errachidia, Moulay Ismail University of Meknes, Morocco.

Cite as: Farhaoui O, Rida Fethi M, Zeroual I, El Allaoui A. Toward Innovative Recognition of Handwritten Arabic Characters: A Hybrid Approach with SIFT, BoVW, and SVM classification. Data and Metadata. 2023;2:176. <https://doi.org/10.56294/dm2023176>


Submitted: 11-08-2023

Revised: 22-09-2023

Accepted: 26-11-2023

Published: 10-12-2023

Editor: Prof. Dr. Javier González Argote 

Guest Editor: Yousef Farhaoui 

Note: Paper presented at the International Conference on Artificial Intelligence and Smart Environments (ICAISE'2023).

ABSTRACT

The goal of handwriting recognition has been a top priority for those who want to enter data into computer systems for more than thirty years. In several fields, the advent of handwriting recognition technology is highly anticipated. OCR technology has made it possible for computers to recognize characters as visual objects and collect data about their unique characteristics in recent years. In particular, several studies in this field have focused on Arabic writing. The use of machines to examine handwritten papers is the first step in the character identification process. The identification of specific Arabic characters is the main goal of this particular investigation. In computer vision, Arabic character recognition is very important since it's necessary to correctly recognize and classify Arabic letters and characters in manuscripts. In this research, an innovative approach based on identifying Arabic character characteristics using BoVW (bag of visual words) and SIFT (Scale Invariant Feature Transform) features is proposed. These features are clustered using k-means clustering to produce a dictionary. Following that, SVM (Support Vector Machine) is utilized to classify the word images in a visual codebook created using these terms. The proposed approach is an innovative method to deal with the difficulties associated with Arabic hand-writing recognition. The utilization of BoVW and SIFT features is expected to enhance the system's robustness in recognizing and classifying Arabic characters. The proposed approach will be experimentally evaluated using a dataset that includes a variety of Arabic characters written in various styles. The results of this study will offer important new perspectives on the effectiveness and practicality of the approach suggested.

Keywords: Character Recognition, Arabic Handwritten Characters, SIFT, K-means.

RESUMEN

Desde hace más de treinta años, el objetivo del reconocimiento de la escritura manuscrita es prioritario para quienes desean introducir datos en sistemas informáticos. En varios campos, la llegada de la tecnología de reconocimiento de escritura a mano es muy esperada. En los últimos años, la tecnología OCR ha hecho posible que los ordenadores reconozcan los caracteres como objetos visuales y recojan datos sobre sus características únicas. En particular, varios estudios en este campo se han centrado en la escritura árabe. El uso de máquinas para examinar documentos manuscritos es el primer paso en el proceso de identificación de caracteres. La identificación de caracteres árabes específicos es el objetivo principal de esta investigación en particular. En visión por computador, el reconocimiento de caracteres árabes es muy importante, ya que es necesario reconocer y clasificar correctamente las letras y caracteres árabes en manuscritos.

En esta investigación, se propone un enfoque innovador basado en la identificación de características de caracteres árabes utilizando características BoVW (bag of visual words) y SIFT (Scale Invariant Feature Transform). Estas características se agrupan mediante k-means clustering para producir un diccionario. A continuación, se utiliza SVM (Support Vector Machine) para clasificar las imágenes de palabras en un libro de códigos visuales creado con estos términos.

El enfoque propuesto es un método innovador para hacer frente a las dificultades asociadas al reconocimiento de la escritura manuscrita árabe. Se espera que la utilización de las características BoVW y SIFT mejore la robustez del sistema en el reconocimiento y clasificación de caracteres árabes. El enfoque propuesto se evaluará experimentalmente utilizando un conjunto de datos que incluye una variedad de caracteres árabes escritos en varios estilos. Los resultados de este estudio ofrecerán nuevas e importantes perspectivas sobre la eficacia y practicidad del enfoque sugerido.

Palabras clave: Reconocimiento de Caracteres, Caracteres Árabes Manuscritos, SIFT, K-Means.

INTRODUCTION

In the current digital era, a handwritten recognition system is essential since it saves time and reduces the task at hand for humans. Machines can recognize and transcribe handwritten text quickly and correctly by automating the recognition process. For companies and groups that manage a lot of handwritten documents, this is especially crucial. The deployment of a system of that kind could increase output and effectiveness. Consequently, one of the main concerns in the field of technology is the creation and enhancement of machine-learning algorithms for handwriting recognition.

Researchers have been investigating the recognition of Arabic handwriting, a challenging topic within the field of handwriting recognition, for several decades. Some studies have shown that the difficulty in recognizing Arabic handwriting stems from its cursive nature and the variations in the forms of the same character depending on its position within a word. Despite some progress, the recognition of Arabic handwriting continues to present challenges and remains only partially resolved. The Arabic language has a wide reach, with over 422 million speakers,⁽¹⁾ is the fourth most spoken language in the world and is a Semitic language.⁽²⁾ It has had a significant impact on many cultures. There are 28 basic letters in the Arabic alphabet, and each one can take on many forms. The literature contains several academic analyses of the Latin languages that produce modern findings. Nevertheless, despite a wide range of suggested strategies and tactics meant to address associated difficulties, the results fall short of expectations. The reason for this deficiency is that a lot of these methods were first developed to identify Latin and Chinese characters,^(3,4,6) which means they are not well-suited to the unique bending and frequently cursive nature of Arabic writing. As a result, adjustments are frequently required to support Arabic characters, which makes recognition even more difficult.

One of the most important topics in Arabic character recognition is the extraction of features, or primitives. Giving a description, like a vector, to describe the Arabic character, word, or sub-word is the first step in this process. The goal of this representation is to ensure that it is both unique and consistent, facilitating precise character recognition.^(7,8)

To exploit the capabilities offered by Computer Vision (CV), which encompasses the extraction and manipulation of objects present in various file formats such as documents or images, we employed OpenCV. Which stands as a highly significant library that fulfills this role comprehensively. With its support for multiple programming languages, OpenCV is predominantly utilized for tasks involving image processing. By harnessing the power of OpenCV, we were able to leverage the functionalities provided by CV, thereby enriching the feature extraction process and enhancing the overall effectiveness of the Arabic character recognition system.⁽⁹⁾

The present study aims to improve the recognition of isolated Arabic to increase the efficiency and accuracy of handwriting recognition systems. The proposed approach specifically addresses the Scale-Invariant Feature Transform (SIFT) method's efficiency,⁽¹⁰⁾ It has become known for being able to resist scaling, rotation, translation, illumination, and partial affine distortion. BOW and SVM classifiers are used for recognition. To ensure that the suggested method is applicable and effective, the study considers the distinctive qualities of Arabic writing. The rest of this paper is structured as follows: In Section 2, we review some related works in this area. Section 3 presents characteristic Arabic script, and Section 4 discusses our proposed method. Section 5 presents our experiment. A discussion about the experimental results. Finally, in Section 6 we finish with a conclusion.

Related work

For many applications, including text mining and search engines, character recognition was an essential part of text and information extraction systems. The application of machine learning (ML) techniques to the recognition, classification, identification, and verification of Arabic handwritten characters has garnered

increasing attention in recent times. This interest originates from the necessity to overcome the difficulties presented by Arabic script's unique and complicated character set, as well as the growing requirement for effective and precise recognition systems. Here we pre-sent some recent work in this field:

In terms of our suggested techniques and some of the interesting recent work we've completed that relies on the analysis and feature extraction of Arabic characters, this article⁽¹¹⁾ compares the effectiveness of window-based descriptors like SIFT, SURF, LBP, and GIST in identifying Arabic handwritten alphabets. To evaluate the effectiveness of several descriptors in combination with many classifiers, such as support vector machines, artificial neural networks, and logistic regression, a thorough experimental evaluation is carried out. The limited size of the dataset used in this study makes it difficult to generalize the results. The article⁽¹²⁾ introduces a novel Arabic handwriting recognition method that makes use of SIFT descriptors. To facilitate a segment-by-segment comparison between the training and testing images, the proposed technique splits the original image into five vertical frames. The system measures the degree of similarity between each segment of the testing image and its corresponding partition from the training images using the Euclidean distance. After that, the method uses the generated similarity measure and the class of the closest training image to classify the testing image. In the same context, the research in the article⁽¹³⁾ proposes using textural features in a comparison analysis to recognize Arabic handwritten words. Pre-processing, representation, segmentation, feature ex-traction, and classification are the five fundamental components of the methodology used. The researchers used multi- level local phase quantization descriptors, gray-level co-occurrence matrix, and local binary patterns to represent the image of the handwritten text. Furthermore, the paper⁽¹⁴⁾ explains two effective methods. It is suggested that Arabic handwritten character recognition can be increased by using SDP (spatial distribution of pixels) and LBP (local binary patterns). When examined in the Arabic division of the Isolated Farsi Handwritten Character Database, by splitting the image into segments, this technique generates a histogram of the local binary patterns within each segment. The histograms are then concatenated to create a feature vector, which is fed into a neural network classifier. In the same regard, the paper⁽¹⁵⁾ aimed to create a method that considers both the spatial and textural aspects of isolated Arabic handwritten character recognition. To extract features from the char-acter images, the authors suggested using local binary patterns (LBP) and a modified bitmap sampling technique. After that, they examined how well these two approaches performed separately and in combination. The experimental findings demonstrated that, when applied to the IFHCDB (Isolated Farsi/Arabic Handwritten Character Database) dataset, their suggested method produced good recognition accuracy. The same goal was served by other previous work using other ways of characteristic ex-traction, since the paper⁽¹⁶⁾ suggested using a combination of the Ridgelet transform and SVM to create a recognition system for handwritten Arabic words. While the classification step is based on a one-against-all multi-class SVM implementation, Transformation Ridgelet is used to produce related primitives for handwritten Arabic words. Vocabularies taken from the IFN/ ENIT database of Tunisian city names are subjected to experimental testing. In the same field, another work was conducted on a novel approach to model selection, as put forth by the authors in a study.⁽¹⁷⁾ Handwritten figure identification using an automatic support vector machine (Ayat system). The USPS (US Postal Service) and the INDCENPARMI database have been used to evaluate the suggested algorithm. There are 7291 test photos among the 9298 handwrit-ten figure images in this collection. Moreover, the paper⁽¹⁸⁾ presents an algorithm for Arabic letter and character recognition that makes use of support vector machines (SVM) and deep convolutional neural networks (DCN). This article uses both fully connected DCNN and dropout SVM to measure the similarity between input and existing models in order to handle the problem of detecting Arabic letters in the manuscript.

The different character recognition systems

The recognition system can be divided into offline and online categories. A machine learning character recognition system is one in which the input documents are supplied in formats such as scanned, printed, and handwritten.⁽¹⁹⁾ The offline mode is used for printed papers and manuscripts, while the online mode is used for real-time recognition while writing. These two kinds of handwriting recognition have different uses.

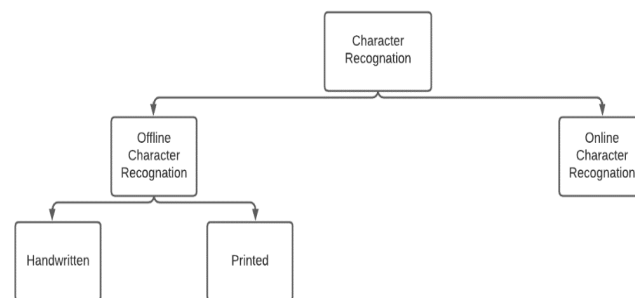


Figure 1. Arabic recognition system types and processing methods

Process of recognition

A recognition system typically involves the following steps:

Preprocessing, segmentation, feature extraction, acquisition, and classification.⁽²⁰⁾ The process of going from a procedure to a recognition system is summarized in the figure below.

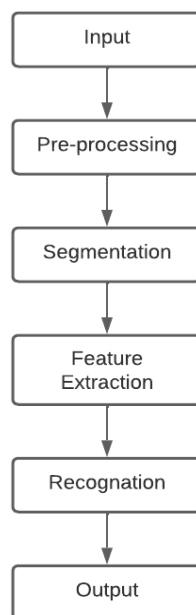


Figure 2. Main steps of a typical character recognition system

Characteristic Arabic script

Arabic writing is widely used worldwide, not just for Arabic but also for many other languages that have adapted to it, due to its status as the main language of over 400 million people around the world. Moreover, it serves as a secondary language for more than three times this number of Muslims around the world, as it was the language in which the Holy Quran was leaked. In 1973, Arabic was included as the sixth official language of the United Nations, in addition to the five original official languages (Chinese, English, French, Russian, and Spanish) that were established at the organization's inception. The Arabic letters present variable morphological structures that depend on the place where they are written. Each letter usually appears in four distinct forms: beginning, middle, final, and isolated. The morphological structures of the Arabic letters vary depending on where they are written. As seen in the table be-low, each letter typically appears in four different forms: initial, middle, final, and solitary. There are 28 characters in the Arabic language, and in the Arabic alphabet, 15 of the 28 letters have one or more dots and are in different positions.⁽²¹⁾ As seen in the figure below, many of the letters in the Arabic script are connected, and it is written from right to left. The Arabic script is written from right to left, and many of the letters are related. Most Arabic characters are associated with complementary parts above as « ق » and below as « ب » or inside the character, as « ك ». Some of these parts, like the dots, are used to differentiate similar Arabic characters as in the case of « ج » and « خ ».

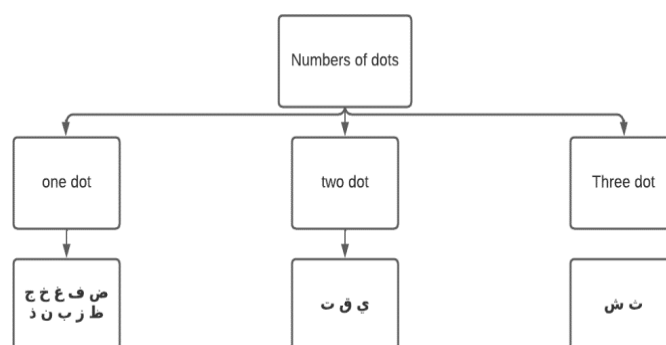


Figure 3. Arabic letters with diacritic points

Table 1. Arabic letters and their four forms

Name	Isolate	Beginning	Middle	end
Alif	ا	ا	ا	ا
Baa	ب	ب	ب	ب
Taa	ت	ت	ت	ت
Thaa	ث	ث	ث	ث
Jeem	ج	ج	ج	ج
Haa	ح	ح	ح	ح
Khaa	خ	خ	خ	خ
Dall	د	د	د	د
Dhaal	ذ	ذ	ذ	ذ
Raa	ر	ر	ر	ر
Zaay	ز	ز	ز	ز
Seen	س	س	س	س
Sheen	ش	ش	ش	ش
Saad	ص	ص	ص	ص
Daad	ض	ض	ض	ض
TTaa	ط	ط	ط	ط
Dhaa	ظ	ظ	ظ	ظ
Ayn	ع	ع	ع	ع
Ghyan	غ	غ	غ	غ
Faa	ف	ف	ف	ف
Qaaf	ق	ق	ق	ق
Kaaf	ك	ك	ك	ك
Laam	ل	ل	ل	ل
Meem	م	م	م	م
Noon	ن	ن	ن	ن
Haa	ه	ه	ه	ه
Waw	و	و	و	و
Yaa	ي	ي	ي	ي

METHODS

Dataset

We used the AHCD1 database⁽²²⁾ from the Kaggle website, which has 16,800 Arabic letters organized into 28 groups, to evaluate our approach. Every participant has written all characters (alef through yah) ten times. The database is split into two sets: a 3360- character test set with 120 photos per class and a 13440-character learning set with 480 images per class.

Scale Invariant Features Transform (SIFT)

Scale-invariant feature transform (SIFT) was developed by David Lowe in 2004.⁽¹⁰⁾ Transforms an image into a set of characteristic vectors (descriptors). The extracted descriptors are invariant to the usual geometric transformations (translation, rotation, and zoom-out). They have also been shown to be robust to a wide family of image transformations, such as slight changes in point of view, noise, blur, contrast, and scene deformation, so that they are sufficiently distinctive for particular needs. The SIFT algorithm consists of two successive and independent operations:

Detecting points of interest (key points) and extracting a descriptor associated with each of them. These descriptors are robust and are often used to match two images. Stitching two images together. Other common uses include object recognition and video stabilization.

Principle of the SIFT algorithm:

The SIFT algorithm is composed of two separate steps: the detection of points of interest and the extraction of a descriptor for each point detected.

Scale-space extrema detection:

In the initial stage, a search is performed at all scales and image locations using the difference of the Gaussian function (DoG) to identify potential points of interest that are invariant in orientation and scale. The difference of the Gaussian (DoG) is then calculated as:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (1)$$

And:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (2)$$

where $L(x, y, \sigma)$ and $L(x, y, k\sigma)$ are two images produced by convolving Gaussian functions with the input image $I(x, y)$ using σ and $k\sigma$ as the respective scales, and $G(x, y, \sigma)$ represents the Gaussian function with σ indicating the standard deviation of the Gaussian, and $I(x, y)$ denotes the input image.

The Gaussian function is represented by
 $G(x,y,\sigma)=1/(2\pi\sigma^2) \exp[-(x^2+y^2)/\sigma^2]$ (3)

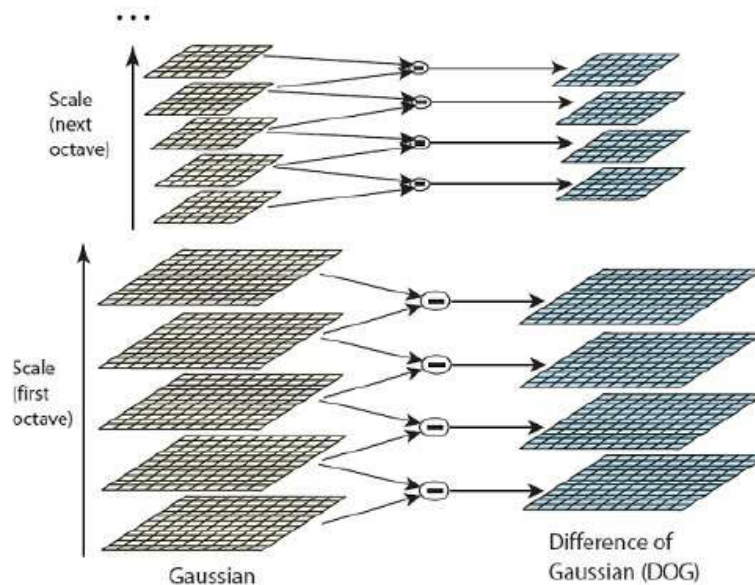


Figure 4. Gaussian filter differences at various scales.⁽¹⁰⁾

Keypoint localization:

In the second stage of the algorithm, a detailed model is utilized to identify and scale the location of each candidate key point. Key points that exhibit stability and resistance to image distortion are selected for further processing. To detect local maxima and minima, the difference of Gaussian (DoG) image undergoes a comparison process with a total of 26 pixels. This includes the surrounding 8 pixels in the same layer, 9 pixels in the upper layer, and another 9 pixels in the lower layer. If the gray-scale value is the highest or lowest among these pixels, the corresponding point is selected as a candidate feature point.

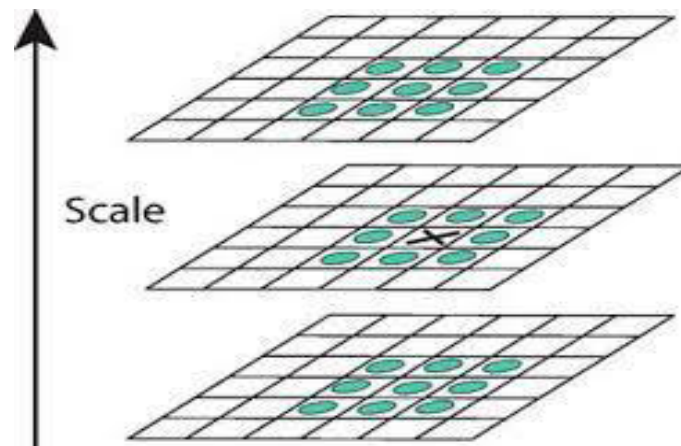


Figure 5. Maxima and minima of the discrete Gaussian difference are detected by comparing a pixel (marked by X) to its 26 neighbors.⁽¹⁰⁾

Orientation assignment:

During the third stage of the SIFT algorithm, the direction of every image gradient is computed. Depending on the local image gradient directions, one or more orientations are given at each key point. After that, all operations are performed on the transformed image data, taking into consideration the assigned orientation, scale, and location. This approach guarantees that the resultant images are invariable to the different transformations. For each key point, a histogram of gradient orientation $\theta(x, y)$ can be allocated using the gradient magnitudes $m(x, y)$ of the neighboring key points. Where L is the Gaussian smoothed image.

Module of gradient :

$$m=\sqrt{[(L(x+1,y)-L(x-1,y))]^2+[(L(x,y+1)-L(x,y-1))]^2}$$
 (4)

Gradient direction :

$$\theta(x,y)=\arctan((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))) \quad \forall (x,y) \text{ dans un voisinage de } (x_0,y_0)$$
 (5)

Key point descriptor:

A local characteristic descriptor is then calculated close to each key point. Each characteristic descriptor includes a characteristic vector with 128 dimensions, with the aim of uniquely characterizing the surroundings of the key point. In order to obtain orientation invariance, the construction of this descriptor relies on the local image gradient, which is then transformed depending on the orientation associated with the key point.

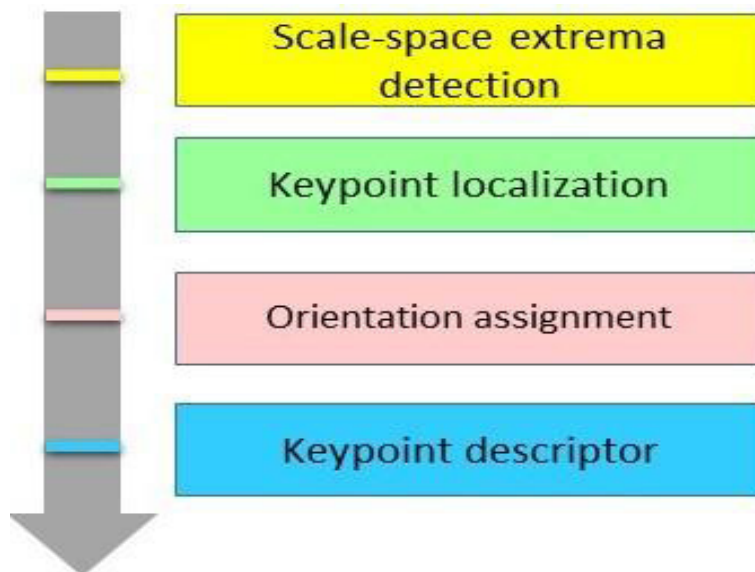


Figure 6. Steps followed for computing sift descriptor

Support Vector Machine (SVM)

The Support Vector Machine (SVM)⁽²³⁾ algorithm was originally introduced by Vapnik⁽²⁴⁾ and is a supervised classification technique that aims to identify the proper class assignment for a given sample. In order to effectively divide the data into different classes, this method involves the construction of a hyperplane in a high-dimensional space.

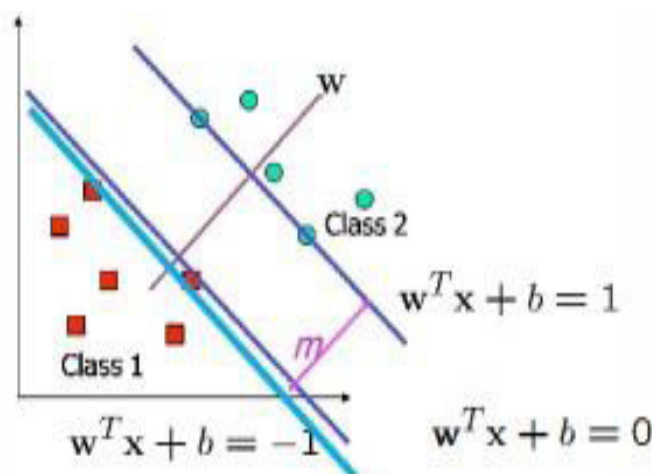


Figure 7. SVM classifier

Proposed Methodology

This paper presents an innovative approach to Arabic character recognition that combines two existing algorithms, Scale Invariant Feature Transform (SIFT) and Support Vector Machine (SVM), to create a hybrid algorithm, figure 8 shows the general structure of the proposed system for Arabic handwritten character recognition.

The conceptual bag-of-words model for handwritten Arabic character classification serves as an outline for the proposed image classification method. Feature extraction, BoF construction, and SVM classifier use are the three main elements that make up this technique. To get the details that make up the description of the image, feature extraction is done first. A dictionary of visual words is then created using the BoW model and the characteristics that were extracted. Next, utilizing calculated descriptors, we employ the SVM classifier in

the final module to classify images.

This study aims to explore the accuracy and potential of the BoW model in combination with the k-means approach and SIFT feature extraction methodology for the goal of categorizing and quantifying visual words. Additionally, the Support Vector Machine (SVM) approach is used to do the classification. The model is trained on a labeled dataset and evaluated using appropriate metrics for performance. By integrating a powerful classification algorithm, effective feature representation, and robust feature extraction, the suggested technique aims to improve image classification accuracy.^(25,26)

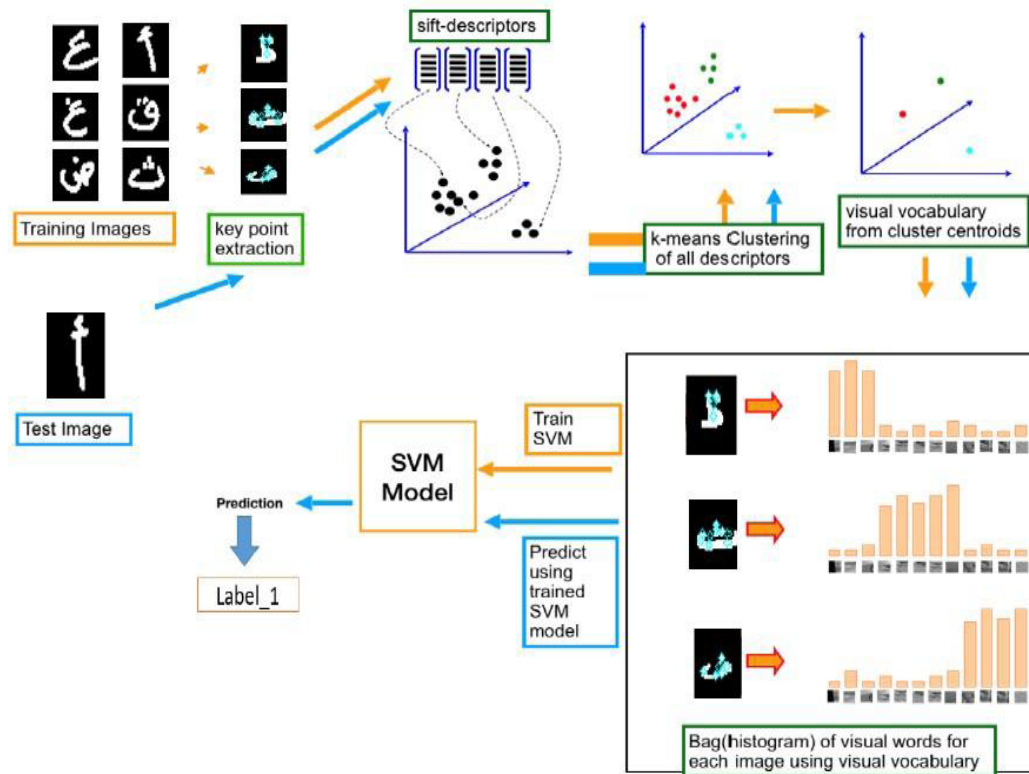


Figure 8. Overview of our proposed system

RESULTS AND DISCUSSION

The algorithms are implemented on an Intel processor (2,30 GHz) with 8 GB of RAM to obtain results. Python was used for the implementation because of its easy-to-use image-processing features. The uniqueness property of the proposed technique is verified by a series of tests using a particular dataset (AHCD1), including Arabic-character wheat images. There are 28 classes in this collection. The SVM classifier was used to measure the classification performance.

In order to fully evaluate the proposed approach, two distinct experiments were conducted:

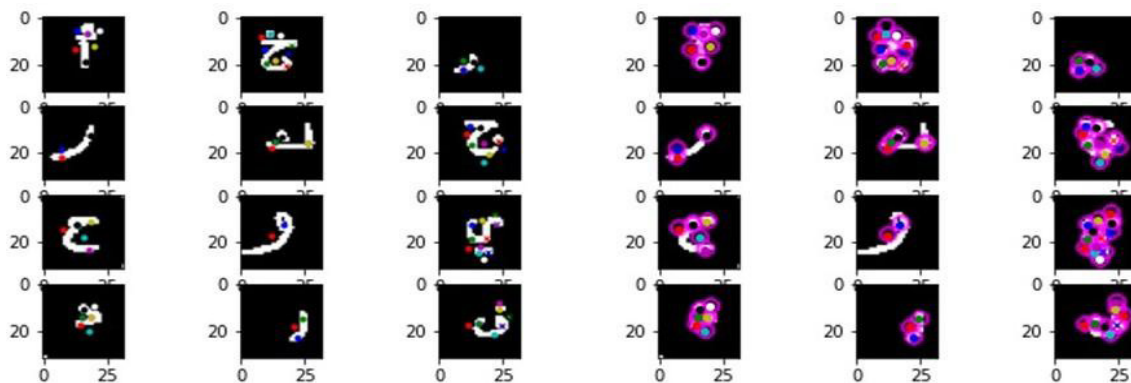










Figure 9. Display of some key points (kp) in the images

Without data processing

The dataset was used in the first experiment without any pretreatment or preprocessing. We run our model through a series of tests on the test set images, after which we ask the model to predict the class that corresponds with each image. To evaluate our model's performance, we test images that undergo geometrical adjustments such as rotation by 90° and 180°. Table 2 shows an example of class prediction for a few randomly selected test images. To evaluate the efficacy of our model, we evaluated images using geometric adjustments, including rotation.

Image test	True class	Predict class	Observation
 test 1	label_15	label_15	Test= ض result= ض “Positive”
 test 2	label_22	label_7	Test= ك result= د “Negative”
 test 3	label_21	label_21	Test= ق résultat= ق “Positif”
 test 4	label_5	label_5	Test= ج résultat= ج “Positif”
 test 5	label_4	label_23	Test= ث résultat= و “Négative”
 test 6	label_26	label_26	Test= و résultat= و “Positif”
 test7(rotation 90°)	label_1	label_1	Test= ا resultat= ا “Positive”
 test 8 (rotation 90°)	label_17	label_17	Test= ظ result= ظ “Positive”

We see through the results that our model succeeds in predicting the class to which the letters belong. Although it failed to determine the class of some images, for example, the image ‘test 2’, which belongs to class label 22, but the model predicts that it belongs to class label 7, the same thing for the image ‘test 5’ is that the model predicts that the image belongs to class label 5. But on the other hand, we observe that our model managed to predict the image classes ‘test 7’ and ‘test 8’. Although we applied a rotation with a degree of 90°, indicating the robustness of the SIFT method at rotation, this gives us an advantage and distinction over the rest of the other models.

We see that our system recognizes handwritten Arabic letters with a recognition rate of 53,42 %. The data indicates that the ‘ا’ character has the highest recognition rate. This is because of the character's simple morphology. The characters ‘د’, ‘ب’ and ‘ض’ have the lowest identification rates, however, this is explained by their complex forms and the presence of diacritical points.

The performance of the model is expressed in terms of three evaluation metrics: accuracy, recall, and f1-score for each class (see Figure. 10).

In order to determine how well our method works, we create the confusion matrix, which is shown in the table below in figure 11. The overall experiment rate as well as the average categorization rate for each of the 28 categories are shown in this matrix. The average classification rates for each class are shown by the figures on the diagonal. Our model gets an accuracy of 53,28 %, a runtime of 24 minutes, and an accuracy of 76,96 % for the training set and 53,28 % for the test set.

	precision	recall	f1-score	support
label_1	0.57	0.70	0.63	142
label_10	0.46	0.60	0.52	126
label_11	0.49	0.53	0.51	135
label_12	0.47	0.45	0.46	122
label_13	0.66	0.55	0.60	128
label_14	0.52	0.43	0.47	131
label_15	0.46	0.49	0.47	110
label_16	0.59	0.61	0.60	127
label_17	0.56	0.65	0.60	114
label_18	0.46	0.48	0.47	112
label_19	0.51	0.53	0.52	112
label_2	0.52	0.56	0.54	110
label_20	0.54	0.45	0.49	117
label_21	0.55	0.45	0.50	117
label_22	0.59	0.59	0.59	130
label_23	0.58	0.70	0.64	117
label_24	0.67	0.69	0.68	148
label_25	0.48	0.44	0.46	119
label_26	0.52	0.61	0.56	127
label_27	0.52	0.54	0.53	123
label_28	0.56	0.49	0.52	113
label_3	0.54	0.44	0.48	119
label_4	0.50	0.51	0.51	114
label_5	0.50	0.42	0.46	125
label_6	0.62	0.54	0.58	124
label_7	0.54	0.39	0.45	116
label_8	0.41	0.48	0.44	121
label_9	0.57	0.52	0.54	132
accuracy			0.53	3427
macro avg	0.53	0.53	0.53	3427
weighted avg	0.54	0.53	0.53	3427

Accuracy : 53.28 %
f1_score : 53.11 %

Figure 10. Classification report for experience 1

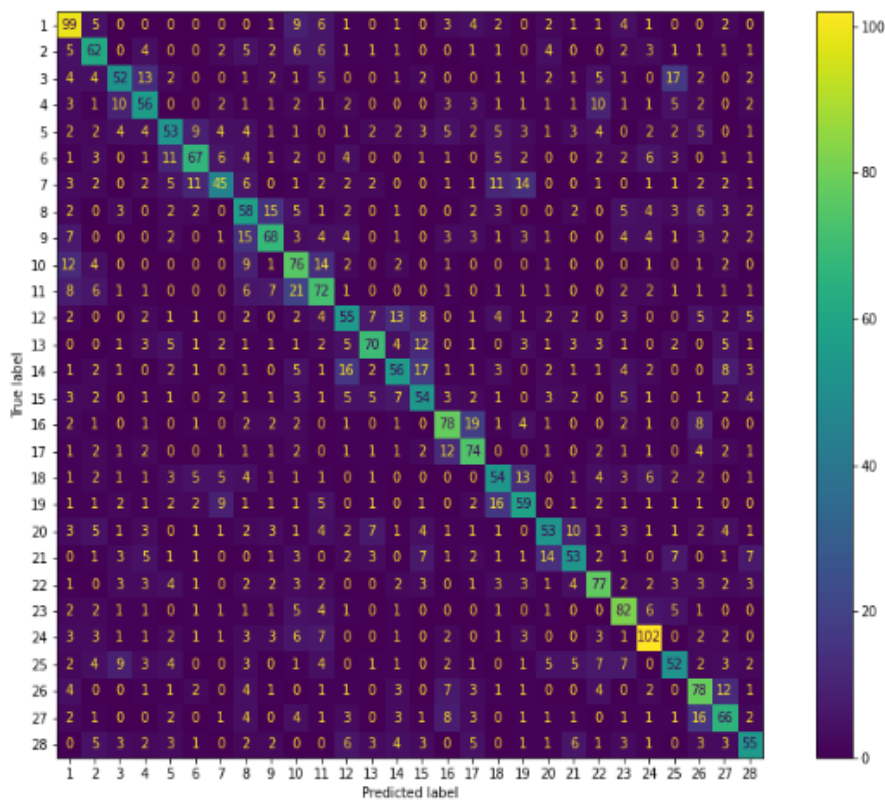








Figure 11. Confusion matrix for the first experiment 1

Application of a treatment

We eventually obtained the following results by using the same settings as the prior experiment and applying a Gaussian filter to the images with a size of (5,5):

Table 3 provides an example of class prediction for a few randomly selected im-ages from the test set. To evaluate the efficacy of our model, we evaluated images using geometric modifications such as rotation.

Image test	True class	Predict class	Observation
 test 1	label_16	label_16	Test= ب résultat= ب "Positive"
 test 2	label_16	label_16	Test= ب résultat= ب "Positive"
 test 3	label_10	label_10	Test= ر résultat= ر Positif
 test2(rotation 180°)	label_28	label_28	Test= ي résultat= ي "Positive"
 test 3	label_26	label_22	Test= ك résultat= ك "Negative"
 test4(rotation 90°)	label_6	label_6	Test= ح résultat= ح "Positive"

Our database's accuracy improved from 53,28 % to 62,53 % following the applica-tion of a treatment, as we found during our research studies. The purpose of the ex-periments presented in this section is to showcase the predictive power of the system we put in place, which also produced good results and the capacity to predict the class for almost every image. Our model was able to correctly predict the classes for the majority of the test images, even when we rotated the test images by 90° and 180°.

	precision	recall	f1-score	support
label_1	0.71	0.79	0.74	122
label_10	0.59	0.81	0.69	117
label_11	0.58	0.62	0.60	127
label_12	0.62	0.54	0.58	127
label_13	0.60	0.62	0.61	119
label_14	0.59	0.51	0.55	114
label_15	0.72	0.63	0.67	107
label_16	0.65	0.66	0.66	120
label_17	0.69	0.63	0.66	123
label_18	0.61	0.63	0.62	140
label_19	0.63	0.62	0.63	137
label_2	0.61	0.74	0.67	112
label_20	0.60	0.58	0.59	115
label_21	0.60	0.57	0.58	109
label_22	0.68	0.63	0.65	124
label_23	0.65	0.77	0.71	116
label_24	0.67	0.73	0.70	113
label_25	0.57	0.55	0.56	109
label_26	0.60	0.66	0.63	130
label_27	0.62	0.70	0.66	135
label_28	0.62	0.47	0.54	134
label_3	0.63	0.67	0.65	114
label_4	0.68	0.64	0.66	129
label_5	0.61	0.51	0.56	132
label_6	0.63	0.54	0.58	142
label_7	0.62	0.44	0.52	138
label_8	0.62	0.76	0.68	127
label_9	0.53	0.54	0.53	108
accuracy			0.63	3440
macro avg	0.63	0.63	0.62	3440
weighted avg	0.63	0.63	0.62	3440
Accuracy :	62.53 %			
f1_score :	62.23 %			

Figure 12. Classification report for experience 2

The method we use operates effectively, as seen by the preceding figure, which shows a 62,60 % recognition rate for handwritten Arabic characters. Character 'ض' has a reported recognition rate, which may be attributed to its basic morphology. Although the character 'ن' has the lowest recognition rate, its complicated form and the presence of diacritical points justify this.

To determine the experiment's performance, we created the confusion matrix, which is shown in the table below figure 13. The overall rate for the experiment as well as the average classification rate for each of the 28 categories are included in this matrix. The average classification rates for each class are shown by the figures on the diagonal. With an execution time of 12 minutes, the average classification rate is 62,53 %, and the accuracy is 90,65 % for the training set and 62,52 % for the test set.

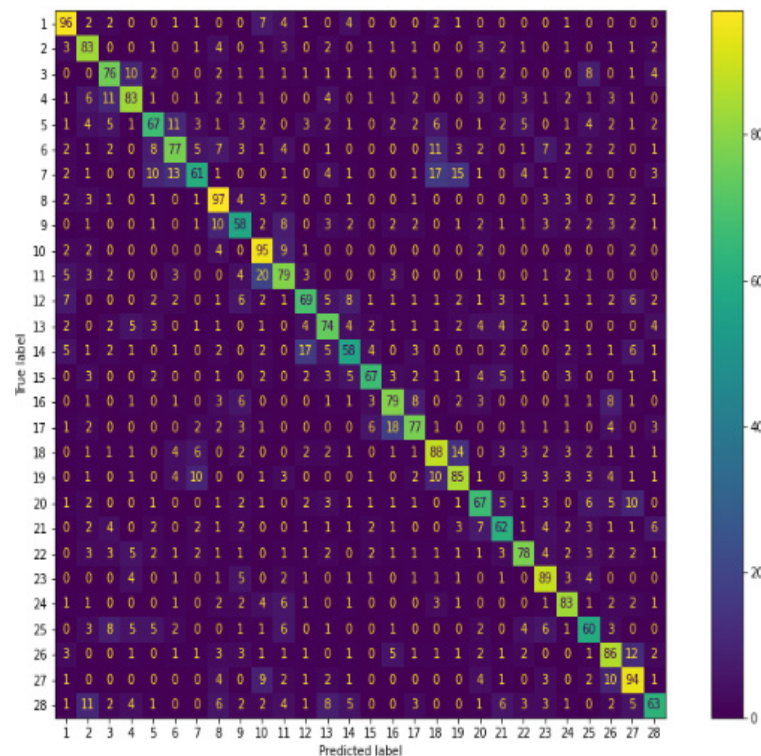


Figure 13. Confusion matrix for the second experiment 2

CONCLUSION

In this paper, we examined the efficacy of the Scale-Invariant Feature Transform (SIFT) method for extracting feature points, the K-means clustering method for clustering these points, and the Support Vector Machines (SVM) algorithm for classification. The efficacy of the Scale-Invariant Feature Transform (SIFT) method for extracting feature points, the K-means clustering method for clustering these points, and the Support Vector Machines (SVM) algorithm for classification were all examined in this work. The proposed approach has proved to be robust against challenges such as image rotation and variations in light. The testing results show potential for improvement by considering all character classes and using a larger database of handwritten character examples. In order to increase recognition, modifications and combinations have to be implemented first in later works. Moreover, sophisticated feature extraction and testing of classification methods are needed to improve recognition accuracy. Continued improvement and development of this method might lead to more advancements in image recognition.

REFERENCES

1. N. Boudad, R. Faizi, O. haj thami Rachid, et R. Chiheb, « Sentiment analysis in Arabic: A review of the literature », Ain Shams Engineering Journal, vol. 9, juill. 2017, doi: 10.1016/j.asej.2017.04.007.
2. M. Alheraki, R. Al-Matham, et H. Al-Khalifa, « Handwritten Arabic Character Recognition for Children Writing Using Convolutional Neural Network and Stroke Identification », Human-Centric Intelligent Systems, vol. 3, nov. 2022, doi: 10.1007/s44230-023-00024-4.
3. M. AbdElNafea et S. Heshmat, « Novel Databases for Arabic Online Hand-writing Recognition System », févr. 2020, p. 263-267. doi: 10.1109/ITCE48509.2020.9047778.

4. Byerly, T. Kalganova, et I. Dear, « No routing needed between capsules », *Neurocomputing*, vol. 463, p. 545-553, nov. 2021, doi: 10.1016/j.neucom.2021.08.064.
5. Baldominos, Y. Sáez, et P. Isasi, « A Survey of Handwritten Character Recognition with MNIST and EMNIST », *Applied Sciences*, vol. 2019, p. 3169, août 2019, doi: 10.3390/app9153169.
6. V. Jayasundara, S. Jayasekara, H. Jayasekara, J. Rajasegaran, S. Seneviratne, et R. Rodrigo, « TextCaps : Handwritten Character Recognition with Very Small Datasets », in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, janv. 2019, p. 254-262. doi: 10.1109/WACV.2019.00033.
7. M. Arif, H. Hassan, D. Nasien, et H. Haron, « A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition », *ijacsa*, vol. 6, no 2, 2015, doi: 10.14569/IJACSA.2015.060230.
8. G. Raju, B. S. Moni, et M. S. Nair, « A novel handwritten character recognition system using gradient based features and run length count », *Sadhana*, vol. 39, no 6, p. 1333-1355, déc. 2014, doi: 10.1007/s12046-014-0274-1.
9. Benaissa, A. Bahri, et A. El Allaoui, « A Combined Approach of Computer Vision and NLP for Documents Data Extraction », in *Artificial Intelligence and Smart Environment*, Y. Farhaoui, A. Rocha, Z. Brahmia, et B. Bhushab, Éd., in *Lecture Notes in Networks and Systems*. Cham: Springer International Publishing, 2023, p. 7-13. doi: 10.1007/978-3-031-26254-8_2.
10. D. G. Lowe, « Distinctive Image Features from Scale-Invariant Keypoints », *International Journal of Computer Vision*, vol. 60, no 2, p. 91-110, nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
11. M. Torki, M. E. Hussein, A. Elsallamy, M. Fayyaz, et S. Yaser, « Window-Based Descriptors for Arabic Handwritten Alphabet Recognition: A Comparative Study on a Novel Dataset ». *arXiv*, 17 novembre 2014. Consulté le: 6 juin 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/1411.3519>
12. L. Chergui et M. Kef, « SIFT descriptors for Arabic handwriting recognition », *IJCVR*, vol. 5, no 4, p. 441, 2015, doi: 10.1504/IJCVR.2015.072193.
13. Korichi et M. L. Kherfi, « A Comparative Study on Arabic Handwritten Words Recognition Using Textures Descriptors ».
14. Y. Boulid, A. Souhar, et Y. Elkettani, « Handwritten Character Recognition Based on the Specificity and the Singularity of the Arabic Language », *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, p. 45-53, juin 2017, doi: 10.9781/ijimai.2017.446.
15. Y. Boulid, A. Souhar, et M. Ouagague, « Spatial and Textural Aspects for Arabic Handwritten Characters Recognition », *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, p. 86-91, juin 2018, doi: 10.9781/ijimai.2017.12.002.
16. H. Nemmour et Y. Chibani, « Handwritten Arabic word recognition based on Ridgelet transform and support vector machines », présenté à *Proceedings of the 2011 International Conference on High Performance Computing and Simulation, HPCS 2011*, août 2011, p. 357-361. doi: 10.1109/HPCSim.2011.5999846.
17. N. Ayat, « Un système neuro-flou pour la reconnaissance de montants numériques de chèques arabes ».
18. M. Shams, Amira. A., et Wael. Z., « Arabic Handwritten Character Recognition based on Convolution Neural Networks and Support Vector Machine », *IJACSA*, vol. 11, no 8, 2020, doi: 10.14569/IJACSA.2020.0110819.
19. S. Djaghbello, A. Bouziane, A. Attia, et Z. Akhtar, « A Survey on Arabic Handwritten Script Recognition Systems », *International Journal of Artificial Intelligence and Machine Learning (IJAIML)*, vol. 11, no 2, p. 1-17, 2021, doi: 10.4018/IJAIML.20210701.0a9.
20. N. Kumar et S. Gupta, « Offline Handwritten Gurmukhi Character Recognition: A Review », *International Journal of Software Engineering and Its Applications*, vol. 10, p. 77-86, mai 2016, doi: 10.14257/ijseia.2016.10.5.08.

21. Bataineh, « A Printed PAW Image Database of Arabic Language for Document Analysis and Recognition », *Journal of ICT Research and Applications*, vol. 11, p. 199-211, août 2017, doi: 10.5614/itbj.ict.res.appl.2017.11.2.6.

22. « Arabic Handwritten Characters Dataset ». Consulté le: 27 juin 2023. [En ligne]. Disponible sur: <https://www.kaggle.com/datasets/mloey1/ahcd1>

23. Q. Wu et D.-X. Zhou, « Analysis of support vector machine classification », *Journal of Computational Analysis and Applications*, vol. 8, avr. 2006.

24. V. N. Vapnik, « An overview of statistical learning theory », *IEEE Trans. Neural Netw.*, vol. 10, no 5, p. 988-999, sept. 1999, doi: 10.1109/72.788640.

25. Romero-Carazas R. Prompt lawyer: a challenge in the face of the integration of artificial intelligence and law. *Gamification and Augmented Reality* 2023;1:7-7. <https://doi.org/10.56294/gr20237>.

26. Gonzalez-Argote J. A Bibliometric Analysis of the Studies in Modeling and Simulation: Insights from Scopus. *Gamification and Augmented Reality* 2023;1:5-5. <https://doi.org/10.56294/gr20235>.

27. Gonzalez-Argote D, Gonzalez-Argote J, Machuca-Contreras F. Blockchain in the health sector: a systematic literature review of success cases. *Gamification and Augmented Reality* 2023;1:6-6. <https://doi.org/10.56294/gr20236>.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

None.

AUTHORSHIP CONTRIBUTION

Conceptualization: Othmane Farhaoui, Mohamed Rida Fethi, Imad Zeroual, Ahmad El Allaoui.

Research: Othmane Farhaoui, Mohamed Rida Fethi, Imad Zeroual, Ahmad El Allaoui.

Drafting - original draft: Othmane Farhaoui, Mohamed Rida Fethi, Imad Zeroual, Ahmad El Allaoui.

Writing - proofreading and editing: Othmane Farhaoui, Mohamed Rida Fethi, Imad Zeroual, Ahmad El Allaoui.