IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 4, December 2024, pp. 4722∼4735
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i4.pp4722-4735

Image classification in cultural heritage

Muath Sabha[1], Muhammed Saffarini[1,2], Rami Yousef[2]
[1]Department of Multimedia Technology, Faculty of Information Technology, Arab American University, Jenin, Palestine
[2]Department of Computer Systems Engineering, Faculty of Engineering and Technology, Palestine Technical University—Kadoorie, Tulkarm, Palestine

Article Info

Article history:

Received Jan 28, 2024
Revised May 29, 2024
Accepted Jun 3, 2024

Keywords:

Convolutional neural network
Cultural heritage
Image classification
Image segmentation
K-nearest neighbors

ABSTRACT

In this paper, an automated supervised image classification technique, specifically for classifying images in the cultural heritage domain, is developed. The developed technique classifies images according to a particular date, culture, people and historical age. The proposed technique consists of two stages, feature extraction using the unsupervised segmentation technique, and the classification stage using supervised classification techniques. Common features are extracted, and their histograms are applied to three classifiers: k-nearest neighbor (KNN), logistic regression (LR), and decision tree (DT). When our technique was applied to a repository of images from cultural heritage, it showed reduced complexity and improved classification accuracy. DT has achieved a higher weighted average recall. This is also represented by the weighted average f-measure where DT has obtained 0.81. DT has outperformed the other classifiers in terms of classifying heritage images.

Corresponding Author:

Muath Sabha
Department of Multimedia Technology, Faculty of Information Technology, Arab American University
Jenin, Palestine
Email: muath.sabha@aaup.edu

1. INTRODUCTION

There are two main categories of image classification algorithms: unsupervised and supervised. In unsupervised classification, the system analyzes the images without the need of providing any sample. While the supervised learning is based on selecting samples and putting labels for images by the user and applying the training using training set (group of images that have labels). After that, the system predicts the class that an image belongs to. For humans, the classification method consists of many stages, starting with image segmentation, extracting features from the image, and then recognition and classification of the image, that is, assigning the testing image to the right category. In machine learning, it depicts the human way of classification.

In general, the supervised image classification consists of the following stages:

- Building the training set: the input will be N images; each image is labeled with one of the K distinguishing classes.
- Learning: using the dataset to identify the characteristics of each class. This step is called training the classifier or learning.
- Evaluation: estimating the quality of the classifier by providing prediction labels from the data set for a new set of images that it has never seen before. The true labels of these images are compared to the ones predicted by the classifier.

Many of the image classifications are used for known and semi-visible objects and shapes. This research is to classify images related to a particular date, culture, peoples, and historical ages by extracting the common features (clothes, tools, and buildings) of the image and then classify the features for which culture it belongs to. The problem is that each culture is defined by many symbols and each age is identified by different colors and shapes. Since there are many cultures that have things in common, and also often cultural images have different object sizes, the process of object identification needs more time.

In the following sections, we review some of the related techniques from the literature in section 2. We then describe our methodology in an extensive way at section 3. We also explain the algorithms that we use in our technique and the reasons behind using these algorithms. The results are shown and discussed in section 4. Finally the conclusions and future work are discussed in section 5.

2. LITERATURE REVIEW

Image classification could be categorized in two major research areas, computer vision and deep learning. In computer vision, image processing is used to segment the image and then to automatically describe by using other AI techniques. Deep learning is using multilayer decision making algorithms in artificial intelligence to understand the image. It is part of machine learning. This section tackles these two fields.

2.1. Segmentation

In image segmentation, Paschos [1] compare the performance of two color spaces (Lab, HSV, and RGB). Because RGB data are clearly ready as the explicitly data provided by the camera, the RGB color space is practically universally recognized by the image processing research society for displaying images. However, there are more regular cognitive spaces, such as Lab and Luv, where the difference in hue is quantified in analogy to how a human perceives such a difference.

In the research done by Delignon *et al.* [2], a comparison of different color spaces is made. The Lab color space has the potential to show results in a more natural style to human vision. Wang and Suter [3] proposed a new color image segmentation method that incorporates both global and local homogeneity. In the hue and density subspace of HSV, the method applied the average shift algorithm. The suggested technique additionally takes into account the color component's cyclic characteristic. Experiments with natural-color photographs have yielded interesting results. Ohashi *et al.* [4] uses the hill-climbing algorithm for image segmentation. It does segmentation by first determining the histogram for the image's three layers, then calculating the local maximum values in the histogram for the three-layer H, S, and V in the image, then the algorithm then connects image pixels with their respective local maximum values. However, there are several disadvantages to this method, including a flat part of the search space in which all surrounding states have the same value and the necessity for a large jump to establish a new section.

Most thresholding approaches involve establishing boundaries based on gray values or pixel density of the image [5]. The color thresholding procedure is carried out based on acclimation and minor changes to the grey-scale thresholding approaches. The RGB data from the surrounding area and another item have been coupled to multilevel thresholding. The object was able to be separated from the background using image fragmentation technology.

Many researches make use of machine learning techniques to segments the images like [6], where the K-Means algorithm is applied to 2, 3 and k-cluster colored images for (k¿3). Silhouette analysis helps to assign the peaks for set k-cluster. However, there are some drawbacks, such as the difficulty in selecting the optimal number of clusters and the fact that the selection of the initial centroids is random. More techniques depend on the texture like [7]–[9].

According to Vadivel *et al.* [10], images are analyzed and segmented using the HSV color space. In this technique, features are extracted by choosing either the hue or the density as the dominating characteristic, which is determined by the pixel's saturation value. Image segmentation is done using statistical region merging (SRM) [11]. The algorithm looks at the statistical underpinnings of a process that is frequently described in computer vision. The algorithm is used to estimate the values in the regional extension and classify them together based on a smaller list standard. JSEG used color quantification to represent numerous classes that can be used to differentiate sections in the image. Color class labels are used to exchange pixel colors, forming a class-map of the image [12].

Singh and Khare [13] creates an autonomous threshold-based information parameter for image segmentation using a genetic algorithm to find the best number of segmentation areas. Only gray-scale photos are used in this technique. According to Sun *et al.* [14], multilayer thresholding is used to revisit gray-scale image segmentation. The authors used nature-inspired algorithms to try to solve the challenge of multilayer thresholding (NAs).

To produce facial picture histograms, a 256-level histogram must first be created, and then every eight contiguous levels are represented as one value to simplify computations and comparisons, as well as to speed up image processing without sacrificing image quality. Instead of 256 pins, there will be just 32 pins after this process [15]. This algorithm is commonly used in facial recognition, but we will adapt it and apply it to our method.

Many techniques are provided in [16], [17] based on the scale-invariant features transform (SIFT). Another algorithm divides the operation into two steps: recognition and segmentation, using a top-down technique for recognition and a bottom-up approach for segmentation. The first is the hypothesis and the second is the verification of the hypothesis, which results in the creation of a set of hypotheses such as object location and ground mask. It computes the list of probable segments created by the hypothesis step in the verification step [18].

2.2. Cultural heritage image classification using deep learning

Image classification in cultural heritage is an important task in the process of digitalization. It can be particularly challenging due to the large number of different image categories, the variability of the characteristics, and the need for high reliability. Deep learning techniques have been employed for architectural heritage images classification using pre-trained convolutional neural networks (CNN). Deep learning algorithms have demonstrated remarkable performance in image classification tasks.

Deep learning is a subdiscipline of the area of artificial intelligence that evolved from classical machine learning. To create a learning model, deep learning approaches mimic the natural processing capability of the human brain [19]. To comprehend deep learning, it is required to first define neural network, which is one of the machine learning strategies given for creating a self-learning learning model. Support vector machine (SVM), naïve Bayes (NB), decision tree (DT), k-nearest neighbor (KNN), and other machine learning techniques are examples. In terms of classification, neural network has performed admirably. Different adaptations of artificial neural networks (ANN), such as perceptron, multilayer perceptron, and backpropagation, have been presented; however, there is a serious drawback still obstructs the precise learning process. Classic ANN has a high level of ability to learn and classify a certain dataset exactitude (i.e. 99%). When the ANN is tested on a new dataset, however, it does not reach the same precision [20]. Overfitting or vanishing ingredients are terms used to describe this problem. When an ANN learns the characteristics and features of certain data in such a way that any change in the data affects the ANN's capacity to find significant patterns, a problem arises. Deep learning algorithms are one of the neural network architectures that make use of several hidden layers [21]. Figure 1 shows the basic architecture for deep learning networks.

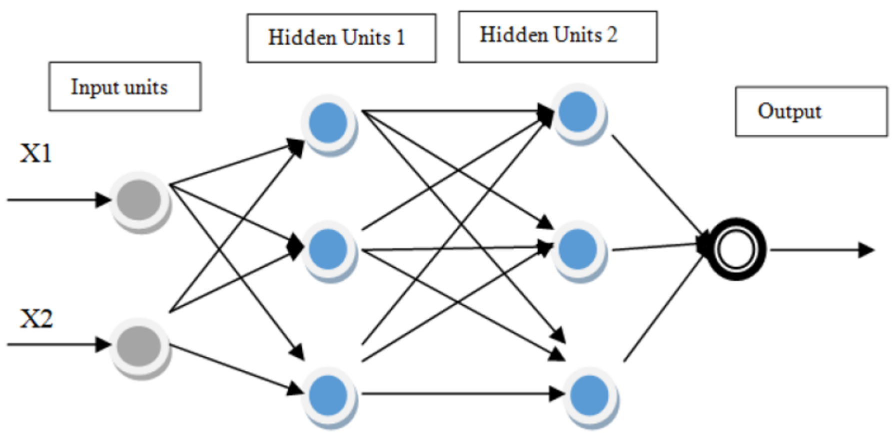

Figure 1. Basic architecture of a deep neural network

Recurrent neural network, long short-term memory (LSTM), deep belief network (DBN), and CNN are examples of deep learning architectures [22]. The sub-sections that follow will go through these architectures in greater depth. The CNN for image classification was used by [23] to classify architectural history images using a deep learning neural network. The results demonstrate a level of precision of up to 90%. Despite the fact that many researchers are tackling the problem of properly classifying and annotating the digital heritage, Belhi *et al.* [24] use a specific type of deep learning classifier, each of which is assigned the task of classifying a combination of asset classes. According to Kambau *et al.* [25] to classify images, audio, video, and text, the authors use the CNN and recurrent neural networks approaches. They have applied this by proposing a crowdsourcing platform to acquire data from the Indian Digital Heritage Space (IDHS) monuments, and designing a transfer learning-based image classifier. Kulkarni *et al.* [26] perform image classification and query-based image label retrieval. Using the pre-trained MobileNet V2 architecture and the ImageNet dataset, the proposed transfer learning technique achieves an inference accuracy of 98.75%.

Janković [27] consider image classification in terms of cultural heritage as one of the most essential jobs in the digital era. It is crucial to build classification methods that are accurate; thus, their research compares four classification algorithms with a CNN. They use: i) the multilayer perception, ii) averaged dependence estimators, iii) forest by penalizing attributes, and iv) the KNN rough sets and analogy-based reasoning. Belhi *et al.* [28] tackle the challenge of automatically classifying and annotating cultural heritage assets using their visual features as well as the metadata available at hand by using multitask neural network, where a CNN is designed for visual feature learning, and a regular neural network is used for textual feature learning.

According to Awad *et al.* [29], an older classification framework was presented. It depends on how the image is annotated and how human feedback is received. The classifier will use these truth-ground annotations as a training set once the human gives a list of image annotations. The classifier is presented with a new test image and is asked to predict the annotations on it. For each new image added to the training set, a human provides additional input. The block diagram of image classification is shown in Figure 2.

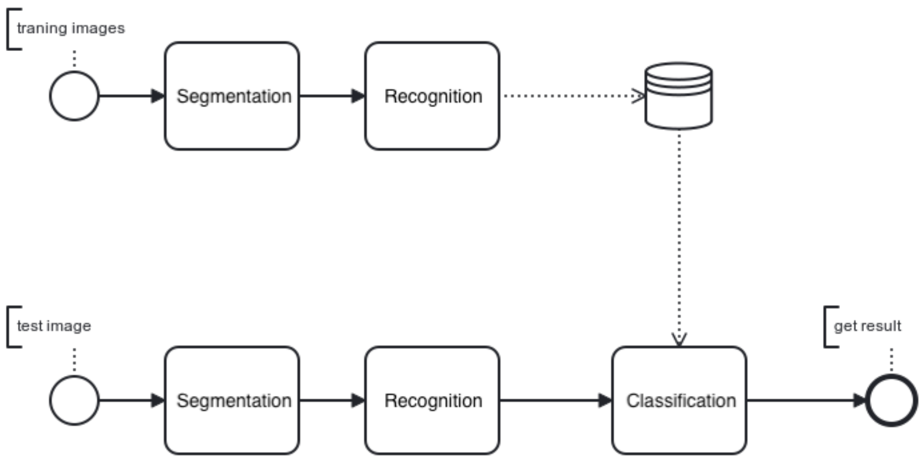

Figure 2. Block diagram for image classification

3. METHOD

In this research, RGB and Lab are tested in the expression of their performance in the ideal image processing function, such as the segmentation of images. There has been an only restricted act on the color sides of textured images. Our methodology is summarized in Figure 3. We first convert the image color space from RGB to CIELab. In Lab color space, there are two color channels; the first channel is blue toward yellow (opponents) and the second is green towards red channel, which is almost closer to how humans perceive images. In addition to that, we have the lightness L.

Using *ab* from *Lab*, we can extract the lightness from the image to reduce the effect of shadows, darkness, and lightness inside the images. We intend to use pure color and to use the mixed colors inside each layer of the image to mimic the human visual system, which may produce more accurate segmentation. Moreover, the Lab gamut is larger than RGB gamut, which separates colors by different absolute relative distances than it in the RGB space [30].

There are many algorithms for image segmentation and there are also many algorithms for image classification. In this paper, some of these segmentation and classification algorithms are discussed and the results of the proposed algorithm are compared with the results of the most well-known algorithms. We will start with segmentation and we will discuss three main algorithms which are gray thresholding, color thresholding, and max value thresholding.

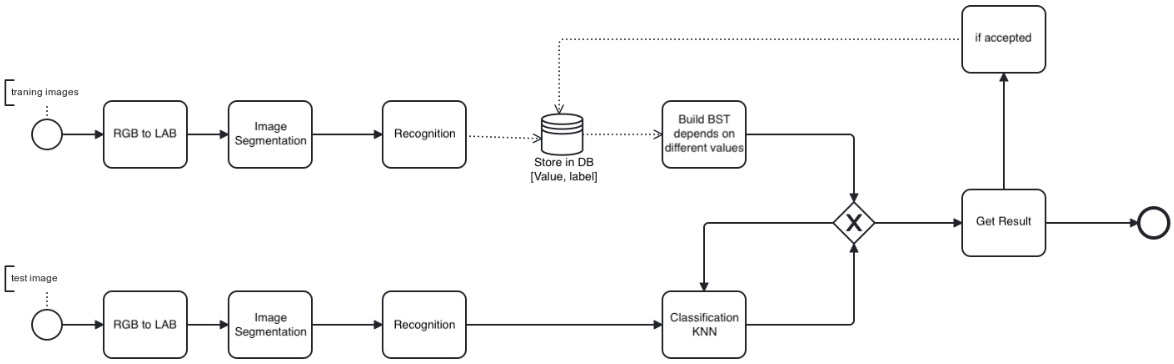

Figure 3. Our methodology steps

3.1. Segmentation

In this step, we segment the images and take the common features inside them, and then calculate the histogram for each Lab channel of the image. The Lab color space has three layers L, a, and b. The range values for L are [0 to 100], the range values for a are [-86.185 to 98.254] and the range values for b are [-107.863 to 94.482]. Each value in one layer may have a set of values in other layers. For example, we may have 30 pixels that have the value 10 on the b channel but different values in the a channel. We assumed that we have 201 different sampled values in the a channel, which are the integer values in the range of the a channel mentioned earlier. This is to reduce the number of histogram columns.

After this step, we merge every 3 adjacent columns for every layer, which reduces the number of columns in each channel. That will reduce the time needed to recognize the features as shown in the next stage from $N \times M$ to a constant value. We repeat the merging step until we reach the total number of columns of 29 columns for each layer. These 29 values in calculation do not take too much processing time. The merging process is performed by taking the average of adjacent columns. After that, we create a vector of 29 values, which results in a new histogram of 29 columns. This will reduce the time complexity in the next step as these vectors will be the new representation for each layer instead of the previous histogram for the a and b layers.

After that, we enter these new vectors (the new histogram) into equations that calculate the local minimum and maximum values in these histograms by finding the centers of the previous histograms to convert the columns to groups of points. That means, first we convert the histogram from the columns to the linear chart that contains continuous values, then we apply the local min-max value so that we can calculate the local min-max values. The aim of calculating the local min-max values is to see when the color inside Lab layers changes from one color to another. After calculating the min-max values, we store these values in vectors, thus producing these vectors of points, and we call them clusters (C). They consist of local min-max values for three layers in the Lab color histogram of the image. We assume that every cluster will have the values between every contiguous local min and max values in the histogram. At this stage, we found that the time for computation will be short and constant, with time complexity $O(C)$. We apply the equation of merging columns that give 20 columns, (not for all the columns in the first histogram,) then the number of the local min-max values equals the number of clusters that we will use to segment the images, into segments to use these segments for recognition and classification as we will see in the next steps.

After finding min-max vectors, we use these vectors as the clusters by applying these vectors into k-means clustering (an unsupervised technique for clustering images without the user providing sample classes or labels, and uses an equation to get results such that Euclidean distance). After that, we apply the k-means to the whole original image, which is a two-dimensional array. We then take the number of local min-max values that will be the number of groups and the number of segments in the image which we want to apply this algorithm on, and the values for the vectors that are the outcomes of local min-max values as initial values for centroid in each cluster. We finally produce indices for clusters, which will enhance the final outcomes of the k-means that will represent the segments in the image we are working on. The L_2 norm is used in the k-means algorithm to group the Lab colors, considering the Lab as a Euclidean space.

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

We find the distance from every color to every centroid to get the minimum distance, and so to which cluster this color is related. If the color distance is shorter to another centroid, this means that this color does not belong to this centroid. This color is transferred to the cluster that has the minimum distance from its centroid. After that, we check if there is any new pixel to be transferred to another cluster, update the centers of clusters (as the first steps), and see if there are no transfers between pixels from any cluster to another. The result will be the vector containing the cluster indices that will represent the segments in the image on which we apply this algorithm. That means, this step will produce a new image with a group of segments where each segment is represented inside the image with a limited number of colors where each color represents the segment from the original image.

As shown in Figure 4, the new image will have a limited number of colors as labels, each color represents a segment inside the image. These segments will be used in the recognition and classification steps. Then we trim the background area adjacent to this object by trimming the area of the same color (so that the image contains a finite number of colors to represent each color segment). We apply the trimming process in the original image in the Lab color space by comparing each area of the new image (containing a finite number of colors) with the original image containing all colors. Then we get all pixels that have a color with value C that found in the index (i_s, j_s) and get all pixels from the original image with the same indices, but there will be a group of colors found within the indexes (i_s, j_s) in the original image. This operation will be applied for all three layers L, a and b in the (i_s, j_s) regions.

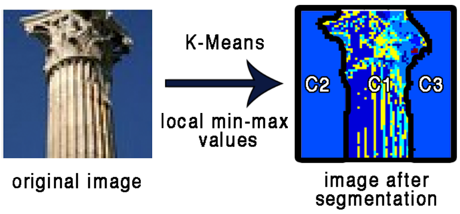

Figure 4. The result from applying K-means and taking local min-max values as the clusters

The segmented regions of the k-means output are transferred to the original image in the same place for all layers. The areas that were trimmed in the previous stages will be used as images and the transfer of these new images to the recognition and classification of calls in the next stage. After segmenting the images, each segment is taken separately (these segments should represent a meaningful segment; otherwise, it will be rejected).

3.2. Recognition step

At this stage, the segments extracted from the original image must be recognized as shown in the previous stage. As discussed in the previous sections, all segments will be treated as a new image. The Luminance histogram of this segment is calculated using a vector of 256 columns. Every 3 adjacent columns are merged by taking the average for these three column values, as shown in Figure 5.

The resulting 86 columns are merged again (each 3 columns) to produce 29 columns. This operation will be applied to all images in the data set. Every image will be represented by a vector of 29 columns, which means a vector of 29 values instead of segments of $N \times M$ pixels. We just need a vector of limited size of 29 values. The operation of this segment will convert from complexity of $O(N \times M)$ to $O(C)$. This means converting from the quadratic system to the constant system, which will increase the speed and the performance and make it more powerful. This vector will be used in the classification step instead of the original segment.

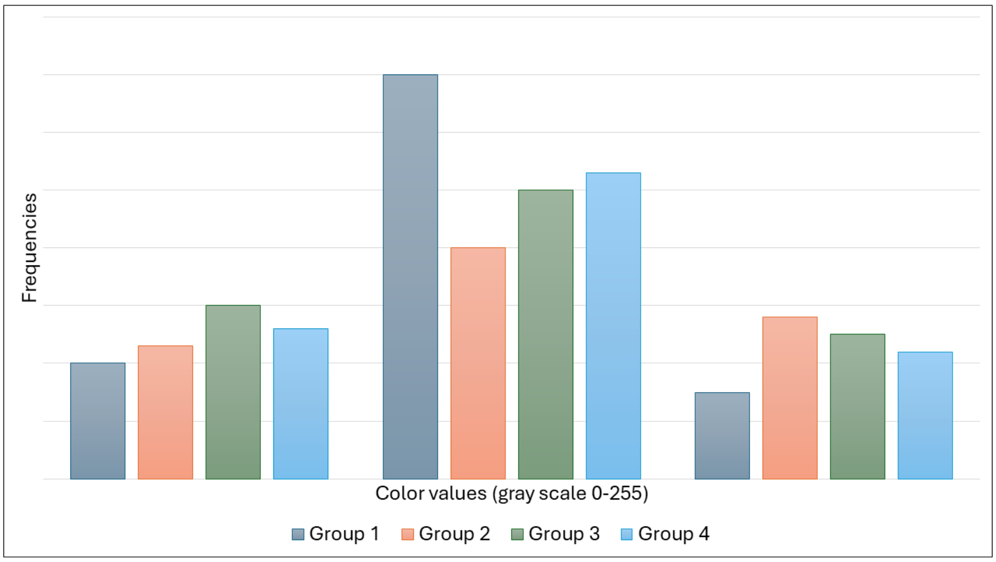

Figure 5. Merging each 3 columns of the histogram by taking their average

3.3. K-nearest neighbor calculations

The KNN algorithm will be used to classify these segments to determine to which class each segment belongs, of which cultural heritage is stored in the dataset. The vector that is produced in the previous step will be considered and all the vectors that are stored inside the dataset that hold labels to see which of these vectors that are nearest to the new vector by using this formula:

$$d_1(I_1, I_2) = \sum_P |I_1^p - I_2^p|$$

Where P is the cell number or the index inside the vector, $I_{1,2}$ represents the images or segments, and d is the result that has the label of specific cultural heritage that the new image belongs to. We subtract the vector from the 29 values that represent the segment produced in the last step with some vectors from a set of vectors in the complete data set.

3.4. Classification

After that, the last step produces a group of values, each value representing the sum of all 29 values produced from the subtracting operation. Finding the minimum value of these values and looking at the image that will get the min value from these operations is next. This image will hold the goal cultural heritage for this segment but not the whole original image because the whole original image has many segments, some of them not cultural features.

Instead of applying the KNN algorithm and subtracting the entire image of size $(x \times y \times 3)$ (three layers), we only need to subtract 29 values, then the time complexity will be O(C) instead of $O(N \times M)$ as mentioned earlier, which has a significant effect for a large number of pixels that will be calculated. Every culture has common characteristics, and the classification is to take every segment of the image tested and find its match from the vectors. It is labeled accordingly and then decided that this segment belongs to a group that represents a certain cultural heritage. The image is added to the data set, and the data set is grown by one.

After filling in the vector, determine to which cultural heritage group the image belongs. However, the time complexity will be $O(K \times N)$ where K is the number of segments in each image in data set and N is the number of images in data set. It will be huge for huge data sets. For example, if we have $100,000$ images in the data set and each image has 50 segments, then the number of iterations will be $100,000 \times 50$ that will produce $5,000,000$ which is huge in real time system. To overcome this problem, we use another methodology to store data which is a binary search tree.

This means that each segment inside specific cultural heritage images will be stored in a binary search tree that represents this cultural heritage. The aim of using the binary search tree is to apply its concept that the value of the root is larger than the left child and less than the right child that gives flexibility for our search, instead of taking $O(k)$ by using linear search, or using search item by item where the complexity is $O(Log(k))$ then it will take $log(K)$ for the iteration, instead of K iterations. This means that we do not need to test the whole tree, just $log(k)$ of elements from this tree (that the binary search tree is just testing the left or right branch). Then, the test will be by calculating the difference between the vector and the vectors in the data set, getting the summation, and testing it with the tree. That will help to make the decision to go to the left or right or to stop if the value is larger than the previous level. Every time we go deep in the left branches, the value should be lower than previous left level and vice versa for the right side; then in every test we will discard half of the vectors from vectors inside the tree that represent specific image; then its value of search will be smaller than the level that stands in it before.

We also applied three classifiers including KNN, logistic regression (LR), and DT. The splitting rate of training and testing has been set to $\%75$ and $\%25$ respectively. The evaluation of classification results will be based on the common information retrieval metrics including precision, recall, and f-measure.

4. RESULTS AND DISCUSSION

After around 150 experiments on 3200 images of cultural heritage from the Pinterest data set of 20 cultures, it is noticed that the segmentation step has several attributes to consider, such as the dependence on the color difference within the image. Another thing is that segmentation depends on image resolution, where higher resolution results in better classification. There is another attribute that determines the resolution of the color-space segmentation which is the channels created by the color space in the laboratory, which made the selection of the section more accurate and solved the problem created by the lightness. For segmenting certain regions, as shown in our methodology, we segment the images by taking the closed regions. As shown in Figure 6, the number of segments is 3, representing the columns, the sky, and the area between the columns. The first segment will recognize the column as roman culture, the other segments will get unknown label, the result will be a roman cultural image depending on the labeled segment.

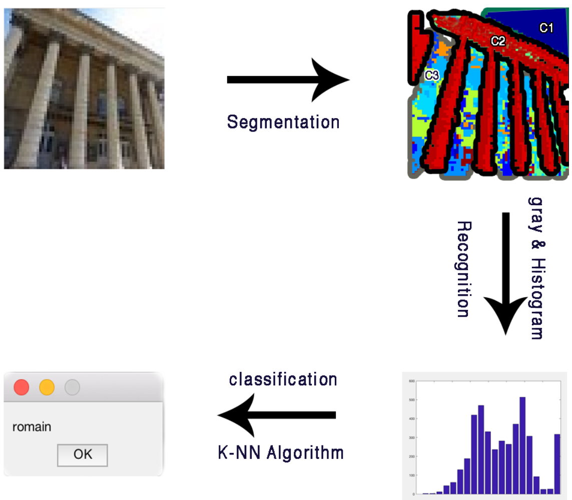

Figure 6. Recognition result from testing a roman culture image

4.1. Time complexity

In Figure 7, a comparison is made between the time complexity of the proposed method compared to the default technique (linear binary search tree). The time complexity is significantly reduced in the proposed technique, where the complexity of the proposed technique is $O(N \times Log(K))$, while it is $O(N \times K)$ in the linear binary tree. As shown in Figure 7, the gap between the two curves increases exponentially as the x-axis progresses (number of segments).

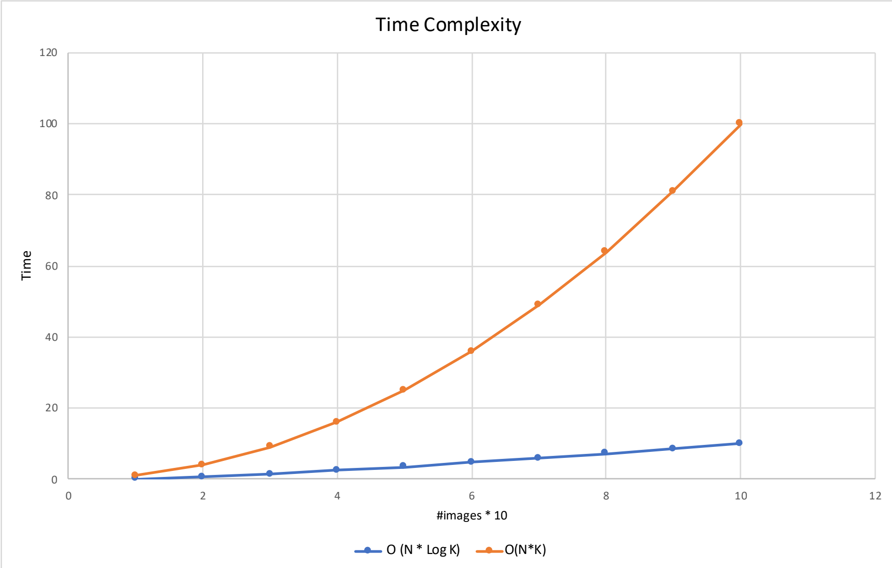

Figure 7. Time complexity for our method vs. the BST

Figure 8 compares the time complexity in the proposed methodology and that in the KNN algorithm. The KNN algorithm subtracts the entire image tested with images from the dataset that has N rows and M columns. However, the proposed method needs only to subtract a vector of 29 values that is static for all images inside the dataset. The gap between the two curves increases as the number of images increases, and this is another bonus point for the proposed methodology.

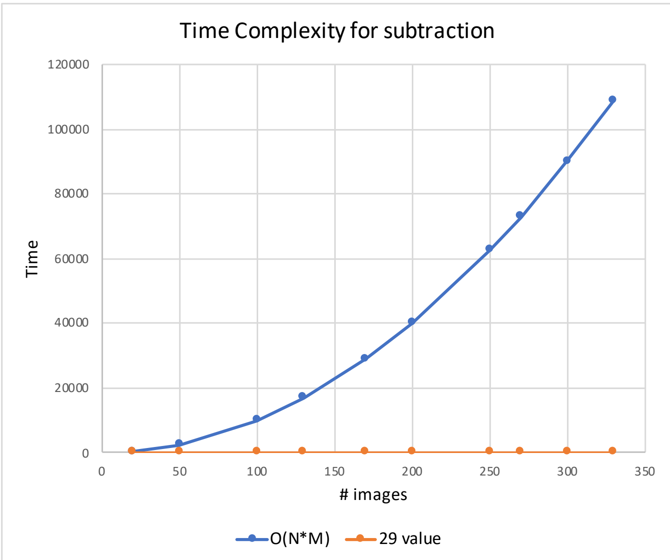

Figure 8. Time complexity for our subtraction method against subtraction in KNN

Figure 9 compares the accuracy of the results by using three different color spaces, CIELab, HSV, and RGB. It is clear that the curves are almost parallel to each other, and the results are almost the same with a privilege to the CIE-Lab color space. We think that the results are close and parallel because the original images are captured in the RGB color space. The privilege of Lab and HSV comes from the fact that these color spaces are closer to human perception. The CIE-Lab color space has a bigger gamut and it gives better results in the k-means algorithm [30]. This proves that our method in using two dimensions out of the *Lab* color space (*ab*) improved the results.

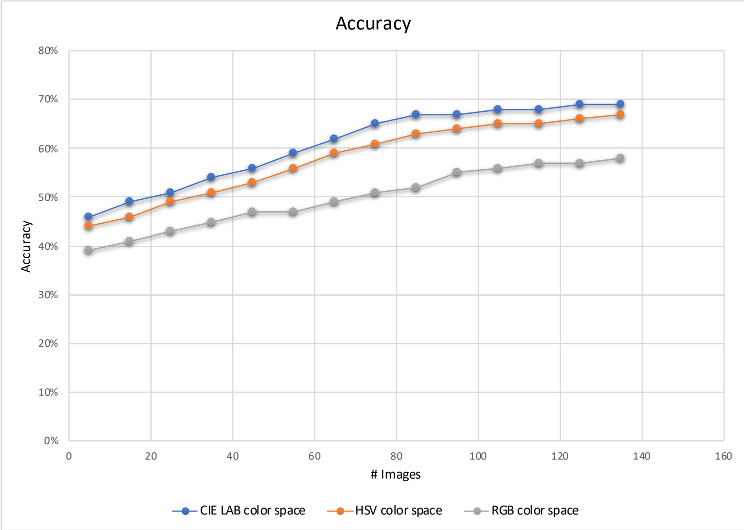

Figure 9. Accuracy using CIELab, RGB and HSV color spaces

Figure 10 compares the classification accuracy of the proposed technique with five of the most related algorithms and techniques from previous research, namely hill climbing, gray multilevel thresholding (GMT), SRM, texture features, and k-means color clustering. The classification accuracy is measured relative to the human classification. It is noticed that the closest algorithms are the texture and K-means color clustering. They depend on the number of clusters to give a finite number of segments. On the other hand, they used the L-layer of the Lab color space in their solution, while the proposed algorithm concentrated on pure color to mimic human perception.

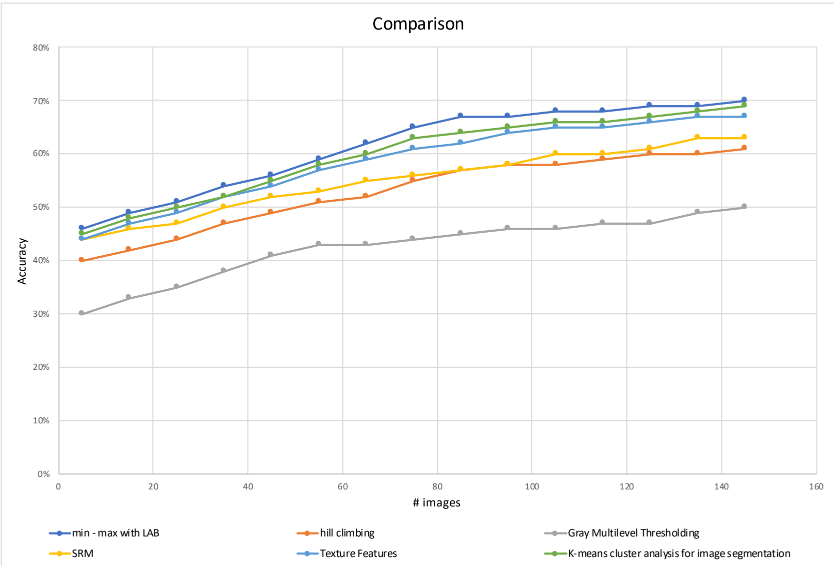

Figure 10. Comparing our algorithm (min-max with Lab) to five other well-known most related algorithms

Hill climbing algorithm produces less accurate results, and that is because it depends on max value, where the number of clusters is small, and it removes too much features in the image. Therefore, the number of segments will be less than our proposed algorithm. The number of segments in our algorithm is about double. The gray thresholding uses one layer while our algorithm uses two layers from a perceptual color space, which uses more useful information. Finally, as shown in Figure 10, our technique gives the best result in cultural heritage features among all the compared techniques.

4.2. Results from k-nearest neighbor, logistic regression, and decision tree

In this section, the classification results for each classifier will be depicted. The results will be represented through the precision, recall, and f-measure metrics. Furthermore, each class label of the images will be examined in terms of precision, recall, and f-measure along with the weighted average as shown in Figure 11.

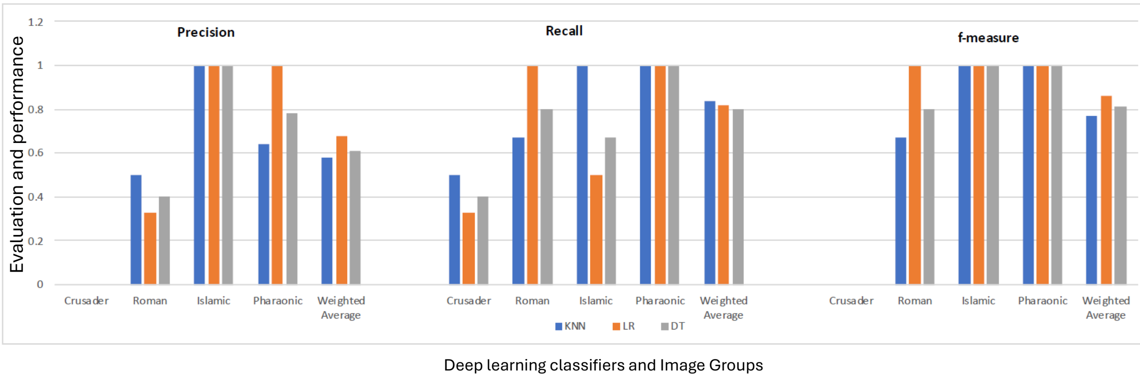

Figure 11. Comparing the results of KNN, LR and DT

Table 1 shows that KNN had no ability to classify Crusader images in which all precision, recall, and f-measure were zero. However, this is not the case with Roman images in which the KNN had the ability to acquire a precision of 0.5, a recall of 0.33, and an f-measure of 0.4. This has followed by a magnificent performance of classifying Islamic images in which the KNN achieves 1.0 for all metrics. In addition, a high performance of classifying Pharaonic images has also depicted where KNN acquires a precision of 0.64, a recall of 1.0, and an f-measure of 0.78. Lastly, KNN has obtained a weighted average precision of 0.58, recall of 0.68, and f-measure of 0.61.

Table 1. Results from KNN, LR, and DT classifiers

Class	Precision			Recall			f-measure		
	KNN	LR	DT	KNN	LR	DT	KNN	LR	DT
Crusader	0.0	0.5	0.0	0.0	0.33	0.0	0.0	0.4	0.0
Roman	0.5	0.67	0.67	0.33	1.0	1.0	0.4	0.8	0.8
Islamic	1.0	1.0	1.0	1.0	0.5	1.0	1.0	0.67	1.0
Pharaonic	0.64	1.0	1.0	1.0	1.0	1.0	0.78	1.0	1.0
Weighted Average	0.58	0.84	0.77	0.68	0.82	0.86	0.61	0.8	0.81

LR had the ability to classify Crusader images with 0.5, 0.33, and 0.4 for precision, recall, and f-measure, respectively. On the other hand, LR has also classified Roman images with 0.67, 1.0, and 0.8 precision, recall, and f-measure, respectively. This has also been depicted for Islamic images, where precision was 1.0, recall was 0.5, and f-measure was 0.67. Noticeably, for the Pharaonic images, LR had the ability to classify all of them correctly with 1.0 of all metrics. Lastly, LR has achieved a weighted average precision of 0.84, recall of 0.82, and f-measure of 0.80.

Similar to KNN, DT was unable to classify any Crusader image with 0.0 for all metrics. However, for other classes, DT had a substantial performance in which it has a precision of 0.67, a recall of 1.0, and an f-measure of 0.8 for Roman images. Both Islamic and Pharaonic images have been correctly classified entirely by DT with 1.0 of all metrics. Finally, DT showed a weighted average precision of 0.77, recall of 0.86, and f-measure of 0.81.

4.3. Performance

In order to identify the outperformance of the classification results between the three classifiers, it is essential to perform a comparison for each class label based on the precision, recall, and f-measure. As shown in Figure 12, it is obvious that both KNN and DT were not capable of classifying any of the Crusader images. The reason behind this lies in the great similarity of the Crusader domes with the Islamic domes. However, LR was able to correctly classify about half of them. Both LR and DT have obtained identical results of precision, recall, and f-measure for classifying Roman images. In general, they have obtained better results than the KNN, which achieved lower precision, recall, and f-measure.

Both KNN and DT have got identical results where they have capable of classifying all Islamic images correctly with 1.0 for all metrics. However, LR has obtained lower results of recall and f-measure in which it had a precision of 1.0. For pharaonic images, the three classifiers have the ability to classify all images correctly with 1.0 of all metrics. The reason for substantial performance lies in the uniqueness of the pharaonic monuments, which facilitates the process of identifying them by the three classifiers.

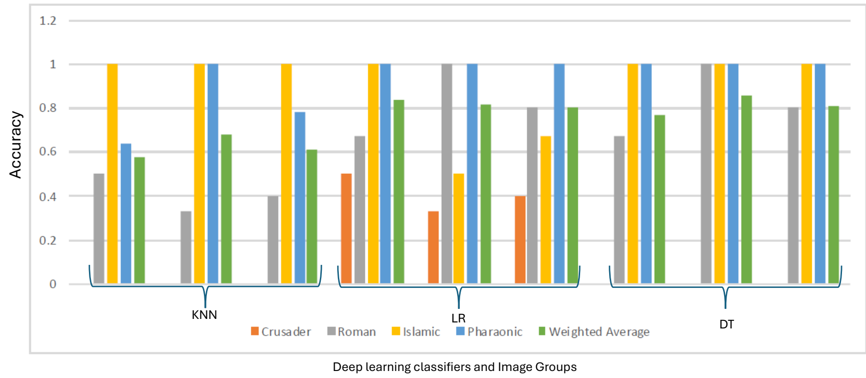

Figure 12. Precision, recall and f-measure of all class label

Figure 13 shows the weighted average results of the three classifiers. Both LR and DT have obtained better results than KNN by achieving higher rates of weighted precision, recall, and f-measure. However, between LR and DT, there are some differences. In particular, LR has higher weighted average precision (i.e., 84) than the DT (i.e., 76). However, DT has achieved a higher weighted average recall (i.e. 86) than LR (i.e. 82). This is also represented by the weighted average f-measure where DT has obtained 0.81 compared to 0.8 achieved by the LR. Generally speaking, DT has outperformed other classifiers in terms of classifying heritage images.

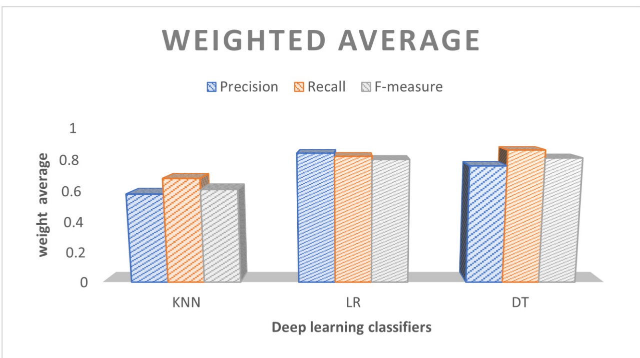

Figure 13. Results of weighted average precision recall and f-measure of the three classifiers

5. CONCLUSION

In this paper, a novel automated supervised image classification methodology is introduced to categorize images within the domain of cultural heritage. The proposed approach effectively categorizes images based on specific attributes such as date, cultural origin, demographics, and historical context. This methodology comprises two primary stages: feature extraction utilizing unsupervised segmentation techniques and subsequent classification utilizing supervised learning algorithms. During the feature extraction stage, common features are extracted from the images and their histograms are utilized as input features for classification. The classification stage employs three distinct classifiers: KNN, LR, and DT. Upon application of this methodology to a dataset comprising images from sources of cultural heritage, notable reductions in complexity and improvements in classification accuracy were observed. The DT classifier demonstrated superior performance, achieving f-measure of 0.81, which outperformed the state-of-the-art studies. For future directions, the use of deep learning classifier might yield promising results in terms of classification accuracy.

ACKNOWLEDGMENTS

The author/s would like to thank the Palestine Technical University - Kadoorie for their financial support to conduct this research.

REFERENCES

[1] G. Paschos, "A color space performance comparison in the processing of color textured images: RGB vs. L*a*b*," in *PICS 1999: Image Processing, Image Quality, Image Capture, Systems Conference*, 1999, pp. 200–203.
[2] Y. Delignon, A. Marzouki, and W. Pieczynski, "Estimation of generalized mixtures and its application in image segmentation," *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1364–1375, 1997, doi: 10.1109/83.624951.
[3] H. Wang and D. Suter, "Color image segmentation using global information and local homogeneity," in *Digital Image Computing: Techniques and Applications*, Melbourne, Australia: Csiro Publishing, 2003, pp. 10–12.
[4] T. Ohashi, Z. Aghbari, and A. Makinouchi, "Hill-climbing algorithm for efficient color-based image segmentation," in *Proceedings of the IASTED International Conference on Signal Processing, Pattern Reconition, and Applications*, 2003, pp. 17–22.
[5] N. Kulkarni, "Color thresholding method for image segmentation of natural images," *International Journal of Image, Graphics and Signal Processing*, vol. 4, no. 1, pp. 28–34, 2012, doi: 10.5815/ijigsp.2012.01.04.
[6] S. M. A. Burney and H. Tariq, "K-means cluster analysis for image segmentation," *International Journal of Computer Applications*, vol. 96, no. 4, pp. 1–8, 2014.
[7] D. S. Guru, Y. H. Sharath, and S. Manjunath, "Texture features and KNN in classification of flower images," *IJCA Special Issue on "Recent Trends in Image Processing and Pattern Recognition,"* vol. 1, pp. 21–29, 2010.
[8] M. Sabha, P. Peers, and P. Dutré, "Texture synthesis using exact neighborhood matching," *Computer Graphics Forum*, vol. 26, no. 2, pp. 131–142, 2007, doi: 10.1111/j.1467-8659.2007.00942.x.
[9] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color- and texture-based image segmentation using EM and its application to content-based image retrieval," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 675–682, doi: 10.1109/ICCV.1998.710790.
[10] A. Vadivel, M. Mohan, S. Sural, and A. K. Majumdar, "Segmentation using saturation thresholding and its application in content-based retrieval of images," in *Image Analysis and Recognition*, 2004, vol. 3211, pp. 33–40, doi: 10.1007/978-3-540-30125-7_5.
[11] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004, doi: 10.1109/TPAMI.2004.110.
[12] Y. Deng, B. S. Manjunath, and H. Shin, "Color image segmentation," in *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999, vol. 2, pp. 446–451, doi: 10.1109/CVPR.1999.784719.
[13] D. P. Singh and A. Khare, "Automatic image segmentation using threshold based parameter and genetic algorithm," *International Journal of Research & Reviews in Computer Science*, vol. 3, no. 3, pp. 1672–1677, 2012.
[14] G. Sun, A. Zhang, and Z. Wang, "Grayscale image segmentation using multilevel thresholding and nature-inspired algorithms," in *Hybrid Soft Computing for Image Segmentation*, Cham: Springer International Publishing, 2016, pp. 23–52, doi: 10.1007/978-3-319-47223-2_2.
[15] F. -E. -Basit, M. Y. Javed, and U. Qayyum, "Face recognition using processed histogram and phase-only correlation (POC)," in *2007 International Conference on Emerging Technologies*, 2007, pp. 238–242, doi: 10.1109/ICET.2007.4516350.
[16] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 2, pp. 1150–1157, doi: 10.1109/ICCV.1999.790410.
[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
[18] L. Wang, J. Shi, G. Song, and I. Shen, "Object detection combining recognition and segmentation," in *Asian Conference on Computer Vision*, Berlin, Heidelberg: Springer, 2007, pp. 189–199, doi: 10.1007/978-3-540-76386-4_17.
[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. London, England: MIT Press, 2016.
[21] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.
[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learnings," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[23] M. Cosovic and R. Jankovic, "CNN classification of the cultural heritage images," in *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2020, pp. 1–6, doi: 10.1109/INFOTEH48170.2020.9066300.
[24] A. Belhi, A. Bouras, and S. Foufou, "Towards a hierarchical multitask classification framework for cultural heritage," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–7, doi: 10.1109/AICCSA.2018.8612815.
[25] R. A. Kambau, Z. A. Hasibuan, and M. O. Pratama, "Classification for multiformat object of cultural heritage using deep learning," in *2018 Third International Conference on Informatics and Computing (ICIC)*, 2018, pp. 1–7, doi: 10.1109/IAC.2018.8780557.
[26] U. Kulkarni, S. M. Meena, S. V. Gurlahosur, and U. Mudengudi, "Classification of cultural heritage sites using transfer learning," in *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, 2019, pp. 391–397, doi: 10.1109/BigMM.2019.00020.
[27] R. Janković, "Machine learning models for cultural heritage image classification: Comparison based on attribute selection," *Information*, vol. 11, no. 1, 2020, doi: 10.3390/info11010012.
[28] A. Belhi, A. Bouras, and S. Foufou, "Leveraging known data for missing label prediction in cultural heritage context," *Applied Sciences*, vol. 8, no. 10, 2018, doi: 10.3390/app8101768.
[29] M. Awad, L. Wang, Y. Chin, L. Khan, G. Chen, and F. Chebil, "A framework for image classification," in *2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, 2006, pp. 134–138, doi: 10.1109/SSIAI.2006.1633737.
[30] M. Sabha, "A novel approach for k-means optimization in image segmentation," in *11th Edition of International Conference on New Technologies, Artificial Intelligence and Smart Data*, 2023, pp. 51–58.

BIOGRAPHIES OF AUTHORS

Muath Sabha holds a Ph.D. degree in Computer Engineering from the University of Leuven in Belgium, and a master's and bachelor's degree from the Arab Academy for Science and Technology and Maritime Transport in Alexandria, Egypt. He works as an assistant professor at the Arab American University, Palestine, where he held several positions such as Vice President for Planning and Development and Academic Quality, Advisor to the University President for University Ranking, Head of the University Image Unit, Dean of the College of Engineering and Information Technology, and Head of the Department of Multimedia Technology and Computer Science. He has research activities in the field of computer vision and machine learning. He can be contacted at email: muath.sabha@aaup.edu.

Muhammed Saffarini holds a Bachelor of Engineering (B.Eng.) in Computer and Electronics Engineering, Master of Science (M.Sc.) in Electrical and Electronics Engineering, Master of Engineering (M.Eng.) in Computer Engineering, and he is a Ph.D. in Engineering of Information Technology. He is currently lecturing with the Department of Computer Science at the Palestine Technical University-Kadoorie at Tulkarm – Palestine. He is a member of the Palestinian Society of Engineers and the Institute of Electrical and Electronics Engineers (IEEE). He can be contacted at email: muhammed.saffarini@ptuk.edu.ps.

Rami Yousef is an associate professor at the Department of Computer Engineering at the Palestine Technical University-Kadoorie at Tulkarm – Palestine. He obtained his Ph.D. from the University Kebangsaan Malaysia (UKM) – Malaysia in Computer Sceince – Artificial Intelligence in 2019, whereas his master was from Birzeit University in Sceintific Computing in 2005. His Bachelor degree was in Electrical Engineering from An Najah National University in 1999. He is now the Dean of Palestine Technical College. He has developed many professional disciplines that keep up with the local market and its need for professionals specialized in all technological and industrial fields. His primary research interests are in the fields of "artificial intelligence applications," remote control, expert systems, neural network, machine learning, and deep learning. He can be contacted at email: r.yousuf@ptuk.edu.ps.