ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 12, Issue, 11, pp. 60271-60276, November, 2022
https://doi.org/10.37118/ijdr.25751.11.2022

RESEARCH ARTICLE

OPEN ACCESS

COMPARATIVE STUDY OF BUILDING RECOGNITION RATES IN URBAN ENVIRONMENTS USING VECTOR QUANTIZATION AND DEEP LEARNING

***[1]Eduardo Silva Vasconcelos, [2]Débora Vasconcelos Melo, [3]Leandro A. da Silva, [4]Tiago A. B. Oliveira, [5]Gilberto Arantes Carrijo and [6]Edilberto Pereira Teixeira**

[1]Eduardo Silva Vasconcelos is with the Faculdade de Engenharia Elétrica - Universidade Federal de Uberlândia; [2]Débora Vasconcelos Melo is with the Faculdade de Engenharia – Universidade Federal de Catalão; [3]Leandro A. da Silva is with the Universidade de Uberaba; [4]Tiago A. B. Oliveira is with the Faculdade Talentos Humanos [5]Gilberto Arantes Carrijo is with the Faculdade de Engenharia Elétrica - Universidade Federal de Uberlândia; [6]Edilberto Pereira Teixeira is with the Universidade de Uberaba

ARTICLE INFO

Article History:
Received 17th August, 2022
Received in revised form
20th September, 2022
Accepted 29th October, 2022
Published online 30th November, 2022

Key Words:

Image Recognition. Vector Quantization. Deep Learning, Artificial Intelligence.

Corresponding author:
Eduardo Silva Vasconcelos

ABSTRACT

Building recognition is essential for a variety of applications such as automatic target detection, 3D city reconstruction, digital navigation, etc. This work aims to comparatively analyze the recognition rates of building images, using the Vector Quantization technique for image compression using the Linde-Buzo-Gray algorithm, with the results obtained by the Deep Learning method. Fourty classes have been analyzed including 10, 20, and 30 images per class, separately, in the RGB color scale, varying the number of centroids in 16, 32, 64, 128, and 256 for the vector quantization technique, and also varying the percentage of the number of images for training in 40%, 50%, and 60%, with their respective percentages of the number of images for recognition, in both methods. To verify the differences, ANOVA was performed, with Tukey's post-hoc at 5% significance. High recognition rates could be obtained from both methods. In the inferential analysis of the results obtained by Vector Quantization, significant recognition rates were found from 32 centroids. No significant difference has been found, by comparing the results obtained from the application of both techniques.

Citation: ***Eduardo Silva Vasconcelos, Débora Vasconcelos Melo, Leandro A. da Silva, Tiago A. B. Oliveira, Gilberto ArantesCarrijo and Edilberto Pereira Teixeira.*** "Comparative Study of Building Recognition Rates in Urban Environments using Vector Quantization and Deep Learning*", International Journal of Development Research*, 12, (11), 60271-60276.

INTRODUCTION

One of the current challenges in pattern recognition is the large-scale recognition of building images. Building recognition can be used in several applications such as automatic detection and target tracking, surveillance, architectural design, mobile device navigation, robot localization, and georeferencing, among others (Li, 1967; Hutchings, 2005; Zhang, 2013; Bruckner, 2012; Park, 2011; Ullah, 2008; Liu, 2008). The great difficulty for this recognition is the enormous variability of parameters found when obtaining these images. These variabilities can be: obtaining images at different angles, various lighting conditions, or partial obstructions by trees, vehicles, or even other buildings (Li, 2013). Dealing with these challenges is a research problem that has been going on for some time and some building recognition systems have been proposed in recent years (Li, 2013; Zhang, 2013), (Ullah, 2008), (Liu, 2008), (Li, 2009), (Li, 2002), (Zhang, 2005), (Duan, 2010), (Chung, 2009).

One of the major problems encountered in these building recognition systems is the large number of features to be extracted and processed for image representation. According to Li and Allinson (Li, 2009), there are several robust feature types for object recognition, such as shape, color, texture, intensity, motion, etc., such that the more feature types are considered, the higher the accuracy for classification. Some works related to this subject stand out, among them, the work of Zhang, Pan, and Zhang (Hutchings) who developed a building recognition system for facade defect detection using Deep Learning. This system can be used as a guide for the sustainable development of cities. It automatically detects building facade elements from images using prior engineering knowledge, e.g., windows, doors, walls, and so on. Furthermore, Li and Allinson (Li, 2009) developed a building recognition system that, after extracting several features, produces a feature map for each image. After that, a saliency model is built by dividing each feature of the map into several sub-regions and each sub-region is described by a saliency feature. Then, for dimensionality reduction, these saliency features, are submitted to one of these reduction algorithms. These are principal component analysis (PCA), linear discriminant analysis (LDA), semi-supervised discriminant analysis (SDA), and locality preserving projections (LPP). Finally, the result is applied to a classifier that in the work of these authors, was the k nearest neighbors. In their research, Li and Shapiro (Li, 2009) used color, orientation, and spatial information for each line segment in the image. These segments were integrated and grouped into a mid-level feature type giving rise to the consistent line clusters. Thus, intracluster and intercluster relationships were used to recognize different buildings. Zhang and Kosecká (Li, 2002) proposed hierarchical building recognition (HBR). HBR is based on leak point detection and localized color histograms. In this system, the detected line segments are grouped into dominant vanishing directions and the vanishing points are estimated by the maximum expectancy (MS) algorithm. Thus, a threshold of the image pixel gradient magnitude is used and the result is one of the groups called: left, right and vertical. If the difference between its gradient direction and the directions of the main leak points is less than a threshold, then a histogram of colors located only in these pixels is calculated and the result obtained is indexed on one of the vectors defined as left, right, and vertical. Finally, the histograms are fitted by chi-square distance, and the recognition results are extracted from the ZuBud database, which contains 201 random buildings in Zurich, Switzerland, with 5 images per building, using the Scale-Invariant Feature descriptor (SIFT). Afterward, a simple probabilistic model is applied to these results to integrate the evidence of the individual fit. Li et al. (2006) stated that the system developed by Zhang and Koseckà has limitations. These limitations are: (1) they assumed that in each image to be recognized there is only one building. However, in real scenes this is not always the case; (2) although they used a fast indexing method, the processing time is still long for extracting many features from color images; and (3) the recognition performance of the algorithm is satisfactory only when the image background is simple. However, the algorithm is inadequate for navigation systems that require real-time processing. Thus, unlike what is found in the literature, this paper proposes an innovation to statistically compare the rates obtained by the Vector Quantization (VQ) technique, through the Linde-Buzo-Gray (LBG) algorithm, with the Deep Learning technique, for the recognition of images of buildings from the SBID database (Madeiro, 2012). This paper is structured as follows: section II describes the Vector Quantization algorithm - Linde-Buzo-Gray and Deep Learning. Section III brings the methodology and the proposed method used in this paper. The results are shown in section IV. And finally, in section V the conclusions are presented.

TECHNIQUES USED

Vector Quantization: Silva et al. (2008) point out that the target of image processing is the digitalization of sample values from an image by compressing and grouping its data, i.e., mapping values from a larger set to a smaller set. It is a clusteringprocess of grouping a set of data into classes or groups, also called clusters so that samples from the same group have high similarity between them. To this end, there is the possibility of improving the performance of this compression using Vector Quantization (VQ) techniques, a relevant technique in mapping and imaging systems (2008). VQ aims to represent data distributions using a reference number of patterns significantly smaller than the number of data (15). Thus, the compression of a data set seeks to decrease the amount of information, both for transmission and storage (16), since only the reference vectors or centroids need to be stored, instead of the entire database. The centroids serve as a representative element of the clusters obtained, and their quantity is chosen based on Equation 1 (Manjunath, 2001).

$$C = 2^m \quad (1)$$

where: "C" is the number of centroids and "m" is a pre-set input value.

The set of all centroids is known as the codebook, which is a previously computed list, and the set of all coding regions is called the partition space. The similarity of the processed image to the original image depends directly on the codebook. The distance between the vectors and the centroids can be measured in several ways, the most widely used being the Euclidean Squared Distance (Equation 2).

$$d(x, \hat{x}) = \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^2 \quad (2)$$

It is noteworthy that, in addition to image compression, there is a wide spectrum of applications for VQ, such as steganography (Jain, 1996), digital watermarking, voice identification, and classification of speech signals with pathologies (Lowe, 2004). In VQ the input image for training is analyzed to determine the codebook, however, the computational complexity in the phase of encoding the vectors to be quantized poses some challenges, such as the design of codebook creation and the sensitivity of the technique to transmission errors. Therefore, some algorithms can be employed. These algorithms when incorporated lead to a reduction in execution time in image processing, as well as finding an optimal codebook for the vector representation of the processed dataset, to minimize the inherent error when used to compress images. Among these image compression algorithms is Linde-Buzo-Gray (LBG). LBG is a technique that infers the way the data is organized and related in each group. It divides the image into centroids, creating a new codebook that is iteratively updated until it reaches the smallest distance, that is, the optimal codebook (Manjunath, 2001).

Deep Learning: The transformation of large amounts of data into recognition is becoming increasingly important and used in various fields (Haghighat, 2013). To extract information from these images Deep Learning, which is a branch of machine learning, has been a widely used methodology. Deep Learning is a technique for training deep architectures that process data in a fast way. Among these architectures are convolutional neural networks (CNNs), LSTMs, Bayesian networks, fully connected convolutional networks, and others (Silva, 2022). CNN, on the other hand, looks for patterns in large amounts of data, in other words, it trains the computer to learn by itself through pattern recognition. CNN consists of a deep-layer neural network with performance characteristics in computer vision problems that stand out, being widely applied in problems of image classification, object detection, etc. (Madeiro, 2012). CNN consists of a few layers, the main ones being: convolution layers, clustering layers, and dense layers (Haghighat, 2013). The convolution layer is responsible for transforming the input image to extract features that allow a correct distinction. To extract this information, convolution filters are used, by means of matrices called masks or kernels, and in some applications, there is the possibility of using more than one convolution kernel in the same image, to extract as many features as possible. The pooling layer is responsible for reducing the image size, keeping only the most important features, and removing the remaining area of the image, which in practice reduces the computational cost. In the pooling layer, the size of the pooling matrix is what will determine the rate of image reduction, for example, a 2x2 matrix will reduce the image size by 50% (Vieira, 2012). The dense layer is executed after the neural network is fully connected, it is where the size of the output image and the activation function are informed. For image recognition, the dense layer is usually built with the rectified linear unit (ReLU) activation function, which is applied to convoluted values to increase their non-linearity. Then another dense layer is built, with dimension equal to the number of classes, to be characterized by the "softmax" activation function.

METHODOLOGY

An ideal building recognition technique should investigate the following aspects: (1) visual models that can accurately describe buildings and be sensitive to small changes in the image, and (2) fast compression to improve the efficiency of an algorithm while decreasing data storage space and computational complexity. In turn, this recognition consists of three parts: (i) feature representation, (ii) feature matching, and (iii) feature classification. After this representation feature matching is conducted to find similarities between the images, i.e., the query image and a reference image in the database. Finally, classification is conducted to determine the best match, where classifiers over statistical models combine the outputs of the global or local feature vectors of similar appearance to maximize the quality of the output on a training set (Li, 2013). For data processing to obtain the results in VQ, the Matlab program version R2015a was used, on HP ProBook hardware with Intel Inside processor, core i7 vPro, and, in Deep Learning, the Google Colab cloud platform, which is an interactive environment that allows writing and executing hosted computational codes, and requires no configuration for its use, besides having free access. The database used was the same used by Li et al. (2014). It is composed of 40 classes, each class consisting of a different building. Each class contains different images of the same building, varying the angle, brightness, rotation, and scale, the challenge is to verify if the proposed recognition methods identify if it is the same building in different images analyzed within the same class. Due to this difference in the number of images per class, the analysis was done by standardizing the extraction of information with the amount of 10, 20, and 30 images per class, in the RGB color scale.

This bank of images makes the task of building recognition challenging, as it combines different lighting conditions and viewpoints. The SBID images have rotation, scaling, different lighting conditions, viewpoint changes, occlusions, and vibration. The database consists of 3,192 images, in JPEG format, of 40 buildings that include churches and a variety of modern buildings such as exhibition halls and office buildings. The size of the images is fixed at 160 x 120 pixels to ensure computational efficiency. However, for each class the number of images is different. These images were taken of buildings at the University of Sheffield and at Sheffield city center on different days and times in the year 2008, which makes the recognition process more challenging (Li, 2014). Fig. 1 shows some of these buildings.

Fig. 1. Example images from the SBID dataset

The proposed method comprises the processing of the building images in RGB color scale. The recognition task, presented in this paper, uses VQ and Deep Learning to build the representative patterns. The design of the vector quantizer codebooks is performed by the traditional LBG algorithm. The results were analyzed based on the variables: centroids, for VQ, the number of images, and training percentage, for both techniques. The information was extracted separately in three steps, the first step was from the first 10 images per class, the second was from the first 20 images, and the third from the first 30 images per class, for each of the 40 classes of the database. The analysis has been stipulated with the first images from each class because their distribution is random. Furthermore, up to 30 images per class have been established in the analysis, because they have different amounts of images, one of them with 35 images. In each stage, the training was run, separately, with 40%, 50%, and 60% of the first images of the classes, and the rest of these images were used for recognition, and these are not the same ones worked on in the training (Table I). The null hypothesis of the study was that there is no significant difference between the two image recognition methods analyzed. The data were entered and processed using the SPSS® 20.0 and Excel® databases and analyzed using descriptive and inferential statistics. For descriptive statistics, the results were analyzed utilizing percentage frequency. For inferential statistics, some analysis of variance (ANOVA) was performed, with Tukey's post hoc test, which is a parametric test that checks the hypothesis and helps to evaluate the importance of one or more factors, comparing the means of the variables in different groups. Significant results have been considered for $p \leq 0.05$.

Vector Quantization: The whole procedure was processed and analyzed for 16, 32, 64, 128, and 256 centroids. The results are presented in tables I to IV. For VQ, the recognition method divides the image into square blocks of 8 x 8 pixels, resulting 300 blocks per image.

Then the Discrete Cosine Transform (DCT) is applied to convert the spatial amplitude data into spatial frequency coefficients. Next, the LBG algorithm is used to perform the training on the selected set of images to generate each of the codebooks (Fig. 2).

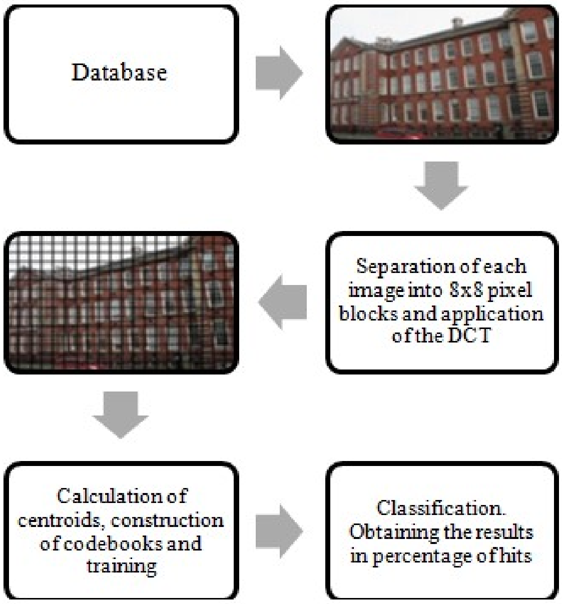

Fig. 2. Extraction of 8x8 pixel blocks per image

Each block is a vector with a dimension of 64 (Equation 3). where: Xm: red color scale ordinates, Ym: green color scale ordinates, and Zm: blue color scale ordinates. The LBG algorithm is executed, starting the recognition process and the classification of the image by class.

$$X_K = \{X_m, Y_m, Z_m; m = 1,2,\dots,64\}; K = 1,2,\dots,300 \quad (3)$$

Theresult is the percentage of correct recognition (Equation 4).

$$Percentage = \frac{Qa}{Qc \times (Qic - Q\ \ )} \times 100 \quad (4)$$

where: Qa: hit quantity, Qc: quantity of classes, Qic: quantity of images in each class, and Qit: quantity of images for training.

Deep Learning: The "max-pooling" strategy was used, which consists of checking whether a given feature is found anywhere in a region of the image and then eliminating the exact positional information. For the CNN network to have sufficient accuracy, the size of the sampled dataset is considered. In building the network, the Google Colab platform was used, due to it supports GPU-enabled hardware, which aids in performance during network training.

In the implementation, the following order was followed

- Data download and upload: the data was hosted on Google Drive, and utilizing the command "gdown", it was possible to download and import them.
- Network configuration: a configuration setting was introduced comprising the width (W) and height (H) of the images (160 x

120), the size of the filter matrix (3 x 3), the size of the downscaling, the max pooling (2 x 2) and the relative path for the files.

- Reading the images: the images are read, using a technique of generating new images from the existing ones (applying random transformations: distortion, rotation, zoom, etc.), to compensate for the relatively small quantity of images used from the database.

In the end, the list of classes to be characterized is obtained. To check the network accuracy according to the variation in the amount of training data, the "validation_split" command was used, which assumes the following values:

- 0.6: for using 40% of the data for training.
- 0.5: for using 50% of the data for training.
- 0.4: for 60% training data.

Creation of the CNN: a CNN composed of two convolution layers was created, where 32 filters were applied to the first layer and 64 filters to the second layer. At the end of the filter sweep, the activation function "ReLU" was applied, followed by "max pooling". The ReLU activation function was adopted, because it does not activate all neurons at the same time, which means that when the input is negative, it will be converted to zero and the neuron will not be activated, consequently, only some neurons are activated, which makes the network more efficient and understandable (20). Then the "flatten" was performed, which is the layer used in the division of the two parts of the CNN for the extraction of features from the neural network. The input vector for the neural network was generated, having a hidden layer of 128 neurons activated again by a "ReLU", and that uses the "Adam Optimizer" to apply the "stochastic gradient descent", and the "categorical cross entropy", as an error function. The output layer consists of 40 neurons, each corresponding to one of the 40 possible classifications, activated by a "softmax" (Fig. 3). The network parameters are derived empirically from tests performed on the network (trial and error). Due to the use of an unbalanced database, the prediction of these values becomes more complex. After satisfactory parameters were determined, the training and testing processes were started, and these parameters can be changed in the future to further improvement of the results.

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_18 (Conv2D)           (None, 158, 118, 32)      896
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 79, 59, 32)        0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 77, 57, 64)        18496
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 38, 28, 64)        0
_________________________________________________________________
flatten_9 (Flatten)          (None, 68096)             0
_________________________________________________________________
dense_18 (Dense)             (None, 128)               8716416
_________________________________________________________________
dense_19 (Dense)             (None, 40)                5160
=================================================================
Total params: 8,740,968
Trainable params: 8,740,968
Non-trainable params: 0
```

Fig. 3. Summary of the network created in the project using a validation_split = 0.4.

Net Training

Finally, the net training process is executed, over the "training set" and "test set", varying the number of epochs and steps per epoch, to obtain the best accuracy for the model.

To facilitate this, the trained net is saved for future use. Fig. 4 shows an illustrative representation of the network implementation process.

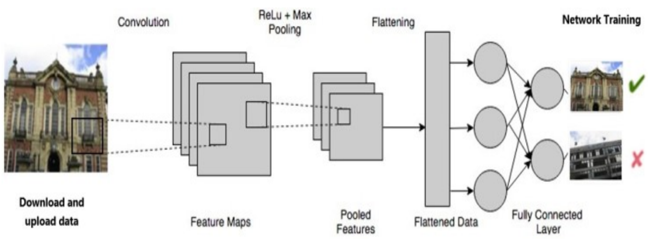

Fig. 4. Illustrative process of the network implementation

RESULTS

The data presented in this section is from a sample of 400 images in the analysis with the first 10 images of the 40 classes, 800 images with the first 20 images, and 1,200 images processed in the analysis with the first 30 images of the classes in both VQ and Deep Learning.

Results with Vector Quantization: When 400 images were analyzed a 100% recognition rate was found, except for the 50% and 60% training with 16 centroids. When analyzing 800 images, a recognition rate of 100% could be obtained from 64 centroids in any training, and, when processing 1,200 images from the database, 100% recognition was found only with 128, and 256 centroids, regardless of chosen training process (Fig. 5). A recognition rate of 100% could be obtained of the buildings in all image classifications for 128 and 256 centroids. Therefore, a distribution of the results into quartiles was made using the box and narrow box diagram, which highlights the recognition averages to indicate the variability of the sample (Fig. 6). This analysis showed the variability of recognition rates for each training session with centroids joining different amounts of processed images per class. It is noteworthy that the recognition for 16 centroids has the largest distribution of data in quartiles, and for 128, and 256 centroids there was no variation in the recognition rate. Because of these variabilities found in the presented rates, for the inference of the data, to verify if there are significant differences, and to determine the effect of two independent variables in a dependent variable, the two-way ANOVA was calculated, by Tukey's post-hoc, with the significance level of $p \leq 0.05$, between the training and centroid variables, for the different amounts of images (Table II).

For the discussion of data, it was characterized that equal letters in the columns of the tables represent that there is no statistically significant difference between them, and on the other hand different letters represent that there is a significant difference. That is, the data in the columns that have the same letter, either "a" or "b", means that there is no significance between them, and in turn, when it presents different letters, "a" and "b", it means that a significance level of $p \leq 0.05$ was found. The analyses were done separately by amounts of images per class. No significant difference was found between the training runs, and a significant difference was found only between centroids, in each training run. For 10 images per class, no statistically significant difference was found between the training sessions and between the centroids. For 20 and 30 images, a statistically significant difference was found only between the recognition rates processed with 16 centroids and the other centroids. Therefore, for the VQ technique, no significant difference was found among the training runs, except for the centroids. It is observed that images processed from 32 centroids show a significant recognition rate.

Deep Learning: The Deep Learning technique also achieved high recognition rates of the buildings, where for 10 images per class, at 40% and 60% training, 100% recognition was achieved (Table III).

Comparison of techniques: For the comparison with the results obtained by Deep Learning processing, we adopted the recognition achieved with 32 centroids, because a significant recognition rate was found from this number of centroids, in the images processed by the

Table I. Quantity of images used for training and recognition

Quantity of images per class	Training	Total quantity of images used for training	Total quantity of images used for recognition	Total processed images
10	40%	160	240	400
	50%	200	200	
	60%	240	160	
20	40%	320	480	800
	50%	400	400	
	60%	480	320	
30	40%	480	720	1.200
	50%	600	600	
	60%	720	480	

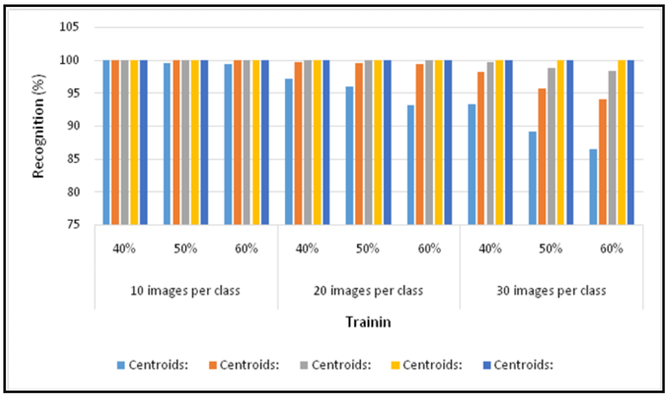

Fig. 5. Descriptive result of the recognition processing of the building images in the VQ technique, with the variables: centroids, number of images, and training

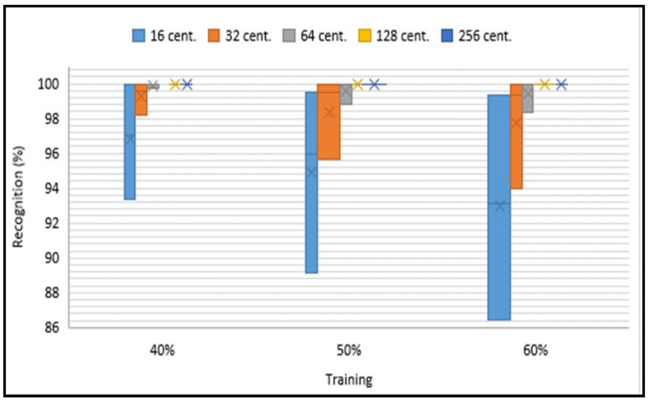

Fig. 6. Box-and-whisker diagram for the VQ analysis

Table II. Descriptive and inferential analysis of the recognition rate of building images using the vq technique

Training	10 images per class			20 images per class			30 images per class		
	40%	50%	60%	40%	50%	60%	40%	50%	60%
16 centroids	100a	99,5a	99,375a	97,0833b	96b	93,125b	93,3333b	89,1667b	86,4583b
32 centroids	100a	100a	100a	99,5833a	99,5a	99,375a	98,1944a	95,6667a	93,9583a
64 centroids	100a	100a	100a	100a	100a	100a	99,7222a	98,8333a	98,3333a
128 centroids	100a	100a	100a	100a	100a	100a	100a	100a	100a
256 centroids	100a	100a	100a	100a	100a	100a	100a	100a	100a

Table III. Descriptive result of the recognition processing of the images of the buildings in the deep learning technique, with the variables quantity of images and training

	10 images per class			20 images per class			30 images per class		
Training	40%	50%	60%	40%	50%	60%	40%	50%	60%
D L (%)	100	97,78	100	99,49	92,37	94,43	93	91,06	95,81

Table IV. Descriptive and inferential analysis of the recognition rates of the building images using the deep learning and vq technique for 32 centroids

Images per class	10			20			30		
Training	40%	50%	60%	40%	50%	60%	40%	50%	60%
D L (%)	100a	97,78a	100a	99,49a	92,37a	94,43a	93a	91,06a	95,81a
QV–32 centroides (%)	100a	100a	100a	99,6a	99,5a	99,4a	98,2a	95,7a	94a

Table V. Results were obtained using the proposed method and the results found in the literature.

Related works	Database	Recognition Rate
Proposed Method – QV	SBID	98,47%
Proposed Method – DL	SBID	95,99%
Li e Allinson [1]	SBID	94,6%
Li e Allinson [7]	SBID	85,25%
Groeneweg. et al [12]	ZuBud	91%
Li e Shapiro [8]	977 color image	94,2%
Zhang e Kosecká [9]	ZuBud	95%
Chung, Han e He [11]	ZuBud	81%

VQ technique. The distribution of these results was made using the box and narrow box diagram, indicating the variability of the sample (Fig. 7). Furthermore, there is a greater variation in recognition rates for the Deep Learning technique between training sessions, when compared to VQ. At 40% training, there was a large difference in variability between these techniques, and at 60% training, both had similar variability in building recognition rates. Given the differences in the percentages between the recognition rates of the building images presented by the two methods under study, data inference was performed to verify if there are significant differences between these techniques (Table IV). No significant difference between the techniques could be observe in this study, when varying the training and the recognition techniques, for each set of images per class. The average recognition rate obtained among the images per class and among all the training sessions, for the Deep Learning technique, is 95.99%, and the average for the VQ technique, with 32 centroids, is 98.47%. However, for QL, when working with 128 and 256 centroids, an average of 100% recognition was obtained.

Comparison with other works: For comparison purposes of the results obtained in this work, Table V shows the list of some papers found in the literature as a function of the recognition rate.

CONCLUSION

Advances in the computational field have promoted new and improved techniques for building recognition. However, we highlight that the VQ technique, using the LBG algorithm, had an excellent performance for building recognition, since, for 128 and 256 centroids, recognition rates of 100% could be obtained, regardless of the used number of images per class and the applied number of training sessions. It was also observed that the images processed by the VQ method presented statistically significant results from 32 centroids. Then, we adopted these recognition rates to compare the results obtained by VQ method with the results found by the Deep Learning method. In the analysis with 32 centroids it was obtained, by the VQ method, an average of 98.47% of buildings recognition, and for the Deep Learning technique, the average was 95.99%. However, when comparing the results obtained by the analysis of all variables using the parametric ANOVA test, with Tukey post-hoc, no significant difference was found between the two methods, which proves the null hypothesis presented in this work. It is concluded that the two techniques when used for building recognition in 40 classes of the SBID database, varying 10, 20, and 30 images per class, have no significant difference between the recognition rates. With the use of statistical tests, the value was generated for the research, since no work was found that statistically compares the processing techniques used in this work, since this is little explored in the literature, but it deserves to be highlighted. Therefore, the innovation of this work is in the proposal of the originality of the comparison, using the parametric ANOVA test, with Tukey post-hoc, between the techniques used for building recognition.

REFERENCES

Bruckner, D. C. Picus, R. Velik,W. Herzner, and G. Zucker, “Hierarchical semantic processing architecture for smart sensors in surveillance networks,” IEEE Trans. Ind. Inf., vol. 8, no. 2, pp. 291–301, May 2012.

Chang, C. C. Lin and Y. Hsieh. “Data Hiding dor Vector Quantization images using mixed-base notation and dissimilar patterns without loss of fidelity, information scienes”, Information Sciences, 201, 70-79, 2012.

Chung, Y. C. T. X. Han, and Z. He, “Building recognition using sketchbased representations and spectral graph matching,”. IEEE Int. Conf. Comput. Vis., pp. 2014–2020, 2009.

Data Science Academy. Deep Learning Book. Disponívelem: https://www.deeplearningbook.com.br/funcao-de-ativacao/.

Duan W. and N. M. Allinson, “Vanishing Points Detection and Line Grouping for Complex Building Façade Identification,” WSCG, pp. 213-220, 2010.

Groeneweg, N. B. Groot, A. Halma, B. Quiroga, M. Tromp, and F. Groen, “A fast offline building recognition application on a mobile telephone,” Adv. Concepts Intell. Vis. Syst., pp. 1122–1132, 2006.

Haghighat, M. S. Zonouz and M. Abdel-Mottaleb, “Identification Using Encrypted Biometrics,” Computer Analysis of Images and Patterns, vol.8048, pp.440–448, 2013.

Hutchings R. and W. Mayol-Cuevas, “Building recognition for mobile devices: Incorporating positional information with visual features,” Comput. Sci., 2005.

Jain A. and A. Vailaya, “Image Retrieval Using Color and Shape”. Pattern Recogn., vol. 29, no. 8, pp. 1233–1244, 1996.

Li and Allinson, N. M. “Building Recognition Using Local Oriented Features,” IEEE Transactions on Industrial Informatics, vol. 9, no. 3, pp. 1697-1704, Aug. 2013.

Li and L.G. Shapiro, “Consistent line clusters for building recognition in CBIR”. IEEE Int. Conf. Pattern Recognition, vol. 3, 952–956, 2002.

Li J. and N. M. Allinson, “Subspace learning-based dimensionality reduction in building recognition,” Neuro computing, vol. 73, no. 1–3, pp. 324–330, 2009.

Li, J. W. Huang, L. Shao and N. M. Allinson, “Building recognition in urban environments: A survey of state-of-the-art and future challenges”, Information Sciences, vol. 277, pp. 406-420, 2014.

Linde, Y. A. Buzo and R. M. Gray. “An algorithm for Vector Quantizerdesing”, IEEE Transactions on Communications, v. 28, n. 1, 84-95, 1980.

Liu, H. “A fuzzy qualitative framework for connecting robot qualitative and quantitative representations,” IEEE Trans. Fuzzy Syst., vol. 16, no. 6, pp. 1522–1530, 2008.

Lowe, D. G. “Distinctive image features from scale-invariant keypoints”, Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.

Madeiro, F. R. R. A. Galvão, F. A. B. S. Ferreira and D. C. Cunha. “Uma alternativa de aceleração do algoritmo fuzzy k-means aplicado à QuantizaçãoVetorial”, TEMA, 13, n. 2, 193-206, 2012.

Manjunath, B. J. Ohm, V. Vasudevan, and A. Yamada, “Color and texture descriptors,” IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 6, pp. 703–715, Jun. 2001.

Ottoni A. L. C. and M. S. Novo. “A Deep Learning Approach to Vegetation Images Recognition in Buildings: A Hyperparameter Tuning Case Study”. IEEE Latin America Transactions, vol. 19, no. 12. 2021.

Park J. and J. Lee, “A beacon color code scheduling for the localization of multiple robots,” IEEE Trans. Ind. Inf., vol. 7, no. 3, pp. 467–475, Aug. 2011.

Silva, J. V. S. L. N. Matos, F. Santos, H. O. M. Cerqueira, H. T. Macedo, B. O. P. Prado, G. J. F. Silva and K. A. Bispo. “Combining Deep Learning Model Compression Techniques”. IEEE Latin America Transactions, Vol. 20, n. 3. 2022.

Srinivasan. A. “Speaker identification and verification using Vector Quantization and mel frequency cepstral coefficients”, Engineering and Technology, 4:1, 33-40, 2012.

The University of Sheffield. Departament of Eletronic and Electrical Engineering. Disponívelem: http://www.sheffield.ac.uk/eee/research.

Trinh, H.-H.D.-N. Kim, and K.-H. Jo, “Facet-based multiple building analysis for robot intelligence,” J. Appl. Math. Computation, vol. 205, no. 2, pp. 537–549, 2008.

Ullah, M. M. A. Pronobis, B. Caputo, J. Luo, R. Jensfelt, and H. I. Christensen, “Towards robust place recognition for robot localization,” IEEE Int. Conf. Robot. Autom., pp. 530–537, 2008.

Vieira, R. N. Brunet, S. Costa, S. Correia, B. Neto and J. Fechine. “Combining entropy measurements and cepstral analysis for pathological voice assessment”, Journal of Medical and Biological Engineering, 2012.

Zhang W. and J. Kosecka, “Localization based on building recognition,” IEEE Workshop Appl. pp. 21–28,2005.

Zhang, Q. L. Lapierre, and X. Xiang, “Distributed control of coordinated path tracking for networked nonholonomic mobile vehicles,” IEEE Trans. Ind. Inf., vol. 9, no. 1, pp. 472–484, 2013

****