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

IJDR

ISSN: 2230-9926

International Journal of Development Research
Vol. 10, Issue, 07, pp. 38241-38246, July, 2020
https://doi.org/10.37118/ijdr.19529.07.2020

RESEARCH ARTICLE

OPEN ACCESS

IMAGE RECOGNITION WITH MACHINE LEARNING: AN ADAPTIVE APPROACH

***Alex Torres Carvalho, W. Muller, Ernande F. Melo and Manoel S.S. Azevedo**

Amazonas State University

ARTICLE INFO

Article History:
Received 14th April, 2020
Received in revised form
11th May, 2020
Accepted 08th June, 2020
Published online 30th July, 2020

Key Words:
Image recognition, Artificial Intelligence, Machine Learning, KNN and Cluster.

***Corresponding author: Alex Torres Carvalho,**

ABSTRACT

Image recognition is a recurring theme in the field of computer vision and it currently borrows some elements from Artificial Intelligence also. However, images are often dealt with in a specific manner, with the algorithms exploring some feature of the said image (be it during pre-processing or during image recognition). Such as is the case when, for instance, some specific algorithms are used for recognition purposes (ranging from biometric recognition to medical images recognition and also the recognition of some defects in electrical / electronic circuit boards), to name just a few. In this article, we propose instead an adaptive algorithm for image recognition, in such a way that an image is treated as if it were composed of a set of pixels - without any other specific parameter. Simple Artificial Intelligence elements are employed in this approach, such as the learning algorithm and the cluster concept.

Citation: ***Alex Torres Carvalho, W. Muller, Ernande F. Melo and Manoel S.S. Azevedo et al.*** "Image recognition with machine learning: an adaptive approach*", International Journal of Development Research*, 10, (07), 38241-38246.

INTRODUCTION

Currently, image recognition algorithms follow a pre-processing sequence that frequently involves the use filters (for noise removal, normalization, feature extraction). It occasionally involves also the application of Principal Component Analysis (PCA) and its variants [1][2][3][8]. Next, comes the recognition phase using geometric distance metrics: Euclidean distances [4] [5] and also Mahalanobis statistical distance [4] [5]. Artificial Intelligence (AI) techniques and more specifically Machine Learning (ML) techniques have been incorporated into the recognition process, thus complementing or replacing existing techniques, such as PCA and its variants [1] [2] [3]. The image databases that are frequently employed to evaluate the performance of such algorithms, are: the AT & T base [6] for face recognition; the FVC2006 DBn-A, DBn-B bases [7] for fingerprint recognition [8]; and the Caltech base [9] for recognizing different images, especially faces against different backgrounds - to name just a few. This article thus proposes an algorithm that does not extract specific features from an imagem, but rather treats an image as an image. The advantage of such approach is that it enables one to take advantage of the same code for different types of recognition problems, be they in biometrics, medical images, defects detection (in circuit boards), or voice recognition, to name just a few.

Furthermore, unlike other, more recent approaches to AI [10] [11] [12], and to Deep Learning - in deep neural networks (DNN) [13] [14], this work employs only the cluster concept [15] [16] and Machine Learning [17] [18] [19] more specifically the KNN algorithm [20] [21] [22]. In other words, simpler and well-known tools for classification problems. Section II describes the Materials and methods. A description of the components and concepts (that make up the environment necessary for the development and implementation of the recognition system) is also given. Section III, in its turn, presents an analysis of the implementation results. Next, comes Section IV Discussion and Section V acknowledgments.

MATERIALS AND METHODS

This section shall describe the components and concepts that make up the system's development environment, which consists basically of a Learning Machine, an image base, and the cluster concept, as well as the concepts of recognition and performance metrics.

Machine Learning: Figure 1 shows the scheme of a Machine Learning (ML), which basically consists of a knowledge base and a learning algorithm. The data is structured as attributes and, in the case of supervised learning, there is a class for each set of attributes [17] [18] [19]. In the case of image data, each set of attributes (pixels) classifies an image. Part of such data used to "train" the learning algorithm, whereas another part is used to test and to evaluate the ML performance. The more data usedto "train" the ML, the better its performance or hit rate in the recognition process. Among learning algorithms, the most popular are: KNN, DT3, SVM and RNA [17] [18]. In this work we chose the KNN learning algorithm [20] [21] [22], for its simplicity and to implement a smaller Euclidean metric, which is the most popular metric used in the image recognition process. [4] [5].

KNN learning algorithm: KNN (K-Nearest Neighbors) is an algorithm for supervised learning [20] [21] [22] that classifies inputs through a prediction method that uses the distance between the current entry and its closest k neighbors, in the training set, to define what the result of its prediction shall be and its classification.

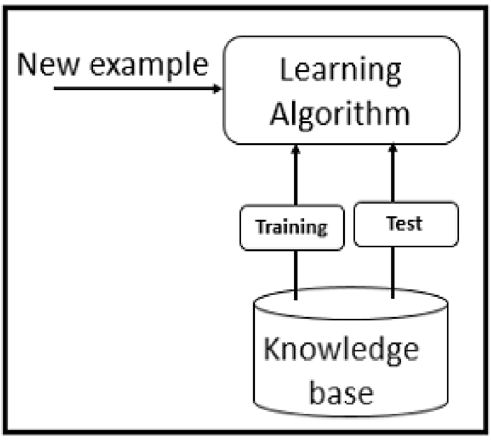


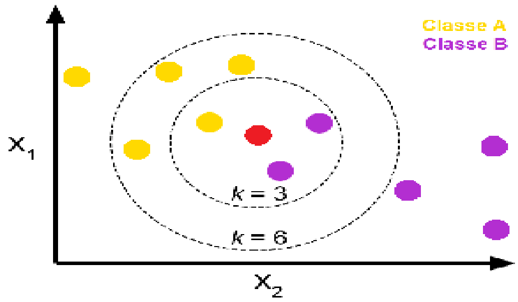
Figure 1. Schematic representation of an ML that consists of a knowledge base (with one data set - for training) and another set (for tests), as well as a learning algorithm

These distance measures can be: Euclidean distance, Manhattan distance, and also Minkowski distance [4] [5]. Given two vectors X = (X1, X2 „ X3, ..., Xn) and Y = (Y1, Y2, Y3, ..Yn) the Minkowski distance is defined as:

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \qquad \text{Eq.(1)}$$

If p = 1, the distance is Euclidean; if p = 2 then it is Manhattan. After the number of closest neighbors is chosen, KNN calculates the distance of the new instance, with the training data, and gathers the k nearest instances. If k = 3, for example, the 3 closest instances are selected. Once the closest instances are selected, classification is made in such a way that the class with more representatives in the neighborhood is the class of the new instance, as shown in Figure 2.

Cluster: Clustering is the unsupervised classification of data, thereby forming clusters that share some features [15] [16]. It is one of the main stages regarding the data analysis processes, in unsupervised learning). In other words, there is no previous classification for the training and testing instances. Applying clustering to the data will generate clusters or classes that, (after some affinity analysis, will, in their turn, make the data set classifiable or not. Figure 2 also illustrates a representation of clustered data, grouped by some similarity criteria (in this case organized by colors). There are several tools for clustering. Among them we mention the KMeans (sklearn library) and also C-means (skfuzzy library) - these being the most popular clustering tools. The problem with the clustering process is that there is a possibility of error in the classification of instances, which could cause problems in recognition. It is necessary to ensure the correct classification of training instances.



Above: “Class A” and “Class B”

Figure 2. The selection of the class amongst two possible ones - for 3 or 6 closest neighbors

Structure of the image bases: Structured image bases are employed to evaluate the performance of the classification algorithms. Of these, Att_faces was the one employed in this work. It contains a set of face images, taken between April 1992 and April 1994 at the Olivetti Research Laboratory (ORL) in Cambridge (United Kingdom). There are 10 different images of the same individual together with 40 different individuals, totaling 400 images. For individuals, the images were taken at different times (under varying lighting conditions and with slightly variation regarding facial expressions - eyes open / closed, smiling / not smiling, as well as and facial details - with glasses / without glasses). All images were taken against a dark and homogeneous background with the subjects in the right front position (allowing for some lateral movement). All files are in PGM format. The size of each image is 92x112 (8-bit gray levels). The images are organized in 40 folders, one for each individual. The DB1-A fingerprint base from FVC2006 was also used in this work, with 12 fingerprint images of the same individual, in a total of 11 individuals, resulting in 132 images of 96 x 96 levels of gray in BMP format. The images are organized in 11 folders, each containing 12 images. A subset of the Frontal face dataset from the California Institute of Technology (Caltech) was also employed, with 150 images of faces (896 x 592 pixels, JPG format) of 10 different individuals under different lighting conditions /with varying expressions and background. Figure 3 (a), (b) and (c) show an example of image taken from these bases.

Figure 3 (a). A sample from the Att_faces image base (from Olivetti Research Laboratory in Cambridge, United Kingdom)

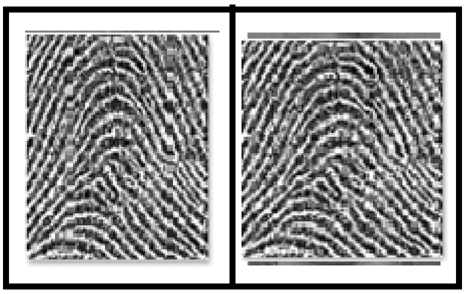


Figure 3 (b). A sample from the FVC2006 DB1-A fingerprint image base

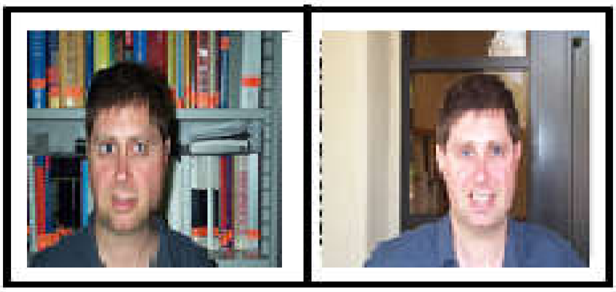


Figure 3 (c). A sample from the image base of the California Institute of Technology (Caltech)

METHODS

This section describes the methods applied for the development of the recognition algorithm.

Setting up the ML knowledge base:

1) Each set of images of the same individual or object is allocated in a numbered folder.
2) A manual cluster is created where each folder represents a class and each image in the folder receives the folder classification.
3) Such organizational structure of classification by folder and image in the folder, allows the application of learning algorithms with supervised training.
4) The folder structure with the images is transformed into a dataframe that classifies the images with the sequential number of the folder containing the images. Table 1 shows the representation of the Att_faces base with 3 folders and 5 images in each folder. In this case, in folder 1 each image is labeled 1 and so on,
5) The ML algorithm trains, predicts and then its performance is measured in relation to the errors and successes in the prediction. The complete algorithm is shown in section II.7

Table I- Dataframe of an att_faces knowledge base. representation of the Att_faces base with 3 folders and 5 images in each folder. In this case, in folder 1 each image is labeled 1 and so on, Each line represents the attributes (pixels) of a vectorized image with their classification given by the number of the folder containing the image;

Performance measure: In order to measure ML performance, we employed as metrics the confusion_matrix {23] [24] (from the sklearn library). Figure 4 shows an outline of the _matrix confusion, in the context of this article. It is observed that the main diagonal elements represent the correct answers in the classification. True positive (tp) is at the base (and the prediction is correct) whereas if true negative (tn) is not at the base the prediction is correct. The other elements of the matrix, in the turn, show the amount of false positive (fp) and false negative (fn) classification errors. Performance is calculated by P=(tp+tn)/(tp+tn+tp+fn).

Table I. Dataframe of an att_faces knowledge base. representation of the Att_faces base with 3 folders and 5 images in each folder. In this case, in folder 1 each image is labeled 1 and so on. Each line represents the attributes (pixels) of a vectorized image with their classification given by the number of the folder containing the image

Image	px0	px1	px2	px3	px4	...	px10300	px10301	px10302	px10303	kclass
0	47.0	49.0	46.0	45.0	47.0	...	46.0	46.0	46.0	46.0	1.0
1	60.0	62.0	61.0	58.0	54.0	...	34.0	33.0	34.0	33.0	1.0
2	40.0	43.0	50.0	49.0	42.0	...	29.0	28.0	27.0	29.0	1.0
3	62.0	54.0	42.0	35.0	36.0	...	75.0	27.0	13.0	25.0	1.0
4	63.0	71.0	75.0	68.0	52.0	...	36.0	36.0	37.0	40.0	1.0
5	35.0	36.0	37.0	37.0	36.0	...	122.0	128.0	128.0	124.0	2.0
6	37.0	35.0	35.0	36.0	37.0	...	26.0	27.0	28.0	28.0	2.0
7	30.0	36.0	35.0	34.0	34.0	...	26.0	26.0	28.0	31.0	2.0
8	34.0	35.0	35.0	36.0	36.0	...	28.0	28.0	25.0	25.0	2.0
9	35.0	32.0	34.0	34.0	33.0	...	25.0	26.0	28.0	29.0	2.0
10	103.0	105.0	105.0	105.0	107.0	...	42.0	42.0	39.0	43.0	3.0
11	101.0	99.0	104.0	105.0	101.0	...	44.0	43.0	43.0	44.0	3.0
12	101.0	104.0	103.0	103.0	104.0	...	43.0	47.0	48.0	45.0	3.0
13	104.0	103.0	105.0	106.0	104.0	...	42.0	38.0	36.0	38.0	3.0
14	101.0	103.0	104.0	103.0	102.0	...	44.0	44.0	45.0	43.0	3.0

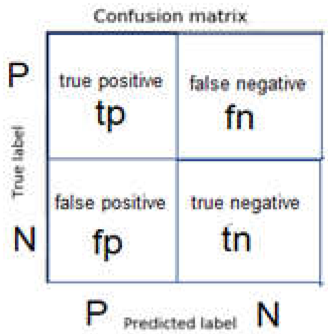


Figure 4. Structure of a confusion_matrix. The elements in the main diagonal represent the correct answers in classification (true positive) while the other elements of the matrix provide the amount of classification errors (false). The performance (P) is P=(tp+tn)/(tp+tn+tp+fn)

Experiments: The experiments consist of applying the KNN algorithm to image recognitionin the image bases described in section II.1. The purpose of the experiments is to answer the following problem: given a base of images, distributed in folders and, a test image, to determine whether the test image is in the base and more specifically, in which folder. To answer these questions, some scenarios and strategies are evaluated for each base. Starting with the Att_faces base.

Experiment I: KNN applied to the Att_faces base: Strategy: Forming the base with the first 20 folders with 8 images of each folder in such a way that the base contains 160 images. The other 2 images (together with the 20 folders with 10 images per folder), go into the predict mode. It is expected that ML recognizes the 2x20 images of the folders registered in the database and does not recognize the 20 x 10 images that are not in the database ,are not registered and therefore do not have similar or equivalent images in the database. Figure 5 shows the performance for this strategy. Overall performance is 197/400=49.2%. Such is a very poor performance: This result say, in summary, that if the image is registered in the database, then KNN can correctly predict the folder in which there is a test image similar or equivalent to any of the images in the database with a high performance (197/200 = 98.5%). However, if the test image is not registered in the database and is not similar or equivalent to any registered image, KNN still generates false positives with 200/200=100% of erros. In other words, the image is not in the database but KNN predicts that it is, thus resulting in a very poor overall performance.

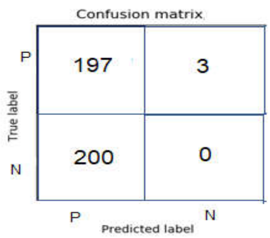


Figure 5. Confusion_matrix, shows the performance of strategyfor the Att_faces base. The false positive rate (200/200 = 100%) is too high. The performance is 49.2%

This is due to the very nature of the model, which always seeks the closest neighbors, with no distance limit. Therefore, it is necessary to explore the excellent results obtained by KNN in strategies 1 and 2, with some solution that circumvents the poor performance of strategy 3.

Minimizing KNN errors: From the KNN prediction, the test image is then considered to be in the folders identified by the KNN classification. The problem comes down to: given a folder with n images and the same individual plus a test image, the question is whether the test image is in the folder or if it there is a similar or equivalent image.For this new problem, several approaches can be taken such as statistics, AI, or brute force, among others. The problem was whether a test image is in a base of size number_folders x number of images is now reduced to the problem of whether a test image is in a folder or base with 1 x number of images. For this reason, in this version of the article, we opted for brute force, which consists of calculating and comparing different attributes of the images in the folder with the test image, such as the difference between pixels, and the Euclidean distance, among others - observing variations and establishing thresholds[8]. Figure 6 (a) and 6 (b) show the use of thresholds and Figure 7 shows the new performance for this strategy. A relevant performance improvement is observed when the threshold is applied.

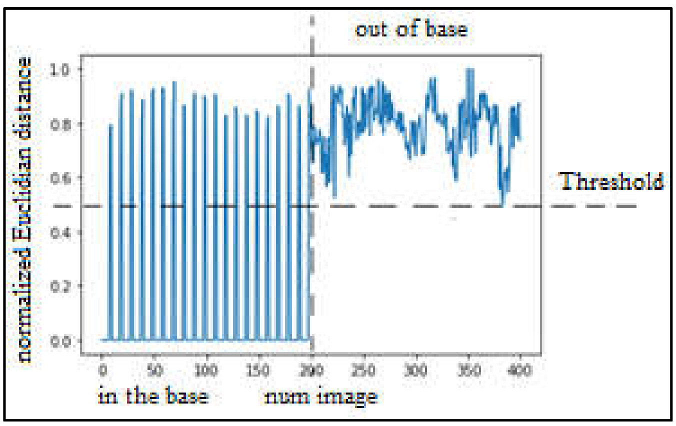


Figure 6 (a). Separation threshold between the face images that are in the Att_faces base (or that are not in the base but belong to the set of face images of the same individual). In this case, the normalized Euclidean distance was used as a criterion for defining the threshold

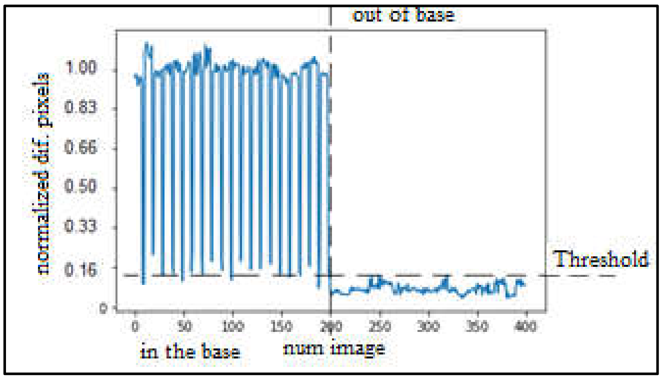


Figure 6 (b). Separation threshold between the face images that are in the base and the images that are not in the base (and have no relation with the images that are in the base). In this case, the difference between normalized pixels was used as a criterion for defining the threshold

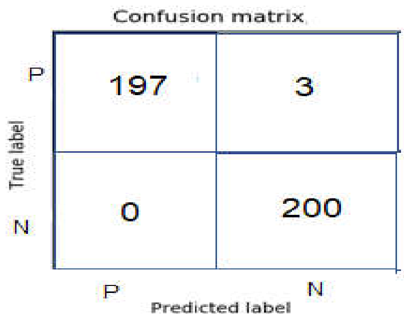


Figure 7. Confusion_matrix shows the performance of strategy 3 for the Att_faces base, with the application of thresholds. The performance is very good: (197 + 200) / 400 = 99.2% while the previous one was 49.20%, (see Figure 5)

Experiment II: KNN applied to DB1-A fingerprint base (from FVC2006)

Figure 3 (b). presents a sample from the DB1-Abase. The same strategies used for the Att_faces base are applied to the DB1-A fingerprint base (observing the proportionality of the base size and the images dimensions). However, as in the case of the Att_faces base, no pre-processing was applied as filters for noise reduction or feature extraction. The performance for this base is shown in Figures 8.

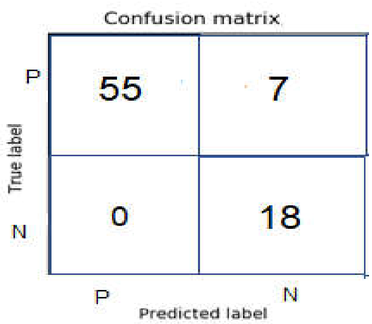


Figura 8. Confusion_matrix, shows the performance of strategy for the DB1-A fingerprint base. The rate of tp = 55, of tn = 18 and fn = 7, resulting in a performance of (55 + 18) / 80 = 91.25%

Experiment III: KNN applied to the Caltech image base: Figure 3 (c) shows a sample from the Caltech base. The same strategies used for the Att_faces database is applied for this database. As in the case of the previous bases, no pre-processing was applied as filters for noise reduction or feature extraction. The performance for this base is shown in Figure 9.

Development Environment: The experiments were developed in the following environment:

Hardware: Note Book SONY VAIO Notebook, 4 GB, 2.5 GHz, 64-bit Framework Spyder3, Python 3.7.3, libraries numpy, pandas 1.16.4, scipy.org sklearn-0.21.2, , scikit-learn.org,OpenCV2.

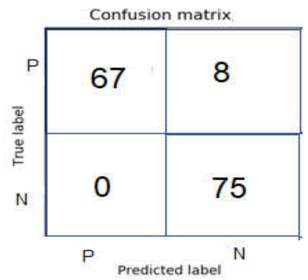


Figure 9 Confusion_matrix. Strategy for the Caltech base with a performance of 94.65%

Graphical representation of the Algorithm: Figure 10 shows a graphical representation of the sequence of steps of the proposed algorithm, briefly described in the section II.5.1

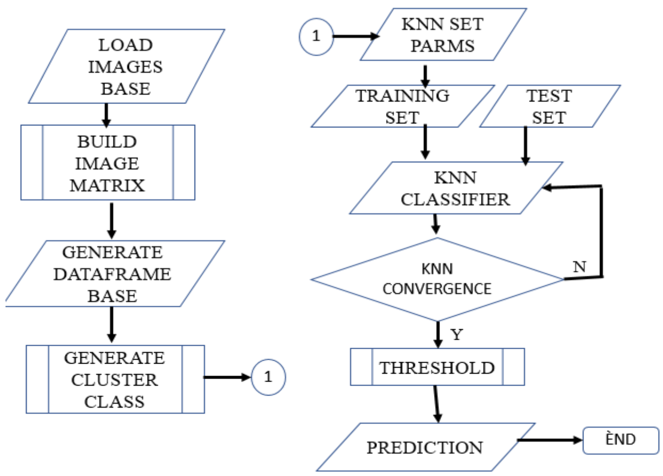


Figure 10. Graphical representation of the proposed algorithm sequence of steps

RESULTS

Table 2 shows a summary of experiments for each base. The experimentsare complete, showing the performance of the algorithm in the classification of images that are in the base, images that are not in the base but that are part of some set of the images that are in the base and the images that are not and have nothing to do with the images that are in the base. The performance for the three bases is very good when applying the thresholds. In the case of the Att_faces base, the performance of 99.2% is very good when compared to some other methods [25]. The DB1-A and Caltech bases are bases with a higher degree of difficulty for recognition, either due to image quality problems in the case of DB1-A or due to the variation in the background of the Caltech base. Despite these difficulties and the lack of pre-processing of the images, the results are very promising, with room for further performance improvements.

Table 2. Summary of performance for each base and experiments, by applying KNN. The Att_faces base shows better results, although the performance for the other bases is close

Base	Base size n_pastas/n_imagens	Performance (%)
Att_faces	20/8	99.2
Db1-A	8/8	91.25
Caltech	5/12	94.65

In general, if the image is not in the base but has some characteristics or similarities with any of the images in the base, KNN can identify if the image is registered, with a performance of 98.5% for the Att_faces base, 82% for the DB1-A base and 86.62% for the Caltech base. These results are extracted from experiment 2 for each base. This KNN feature enables us to reduce the problem (of finding an image on a M_folder x N_image basis) to a problem of finding an image on a 1_folder x N_image basis. It is thus possible to obtain a much higher performance for bases with better quality and quantity of images also (Att_faces), pre-processing with application of filters for the elimination of noise. Or, for a well-behaved base, one could let the user decide visually. That is, if the image is not in the base, the algorithm displays two images anyway. If they are different (as seen by the user himself or herself), it is then certain that the image is not registered in the base. That way, there would be no need to apply the threshold. However, this will depend on the application.Another issue is that due to the KNN features, there is a lot of memory consumption, which makes recognition very slow. But the dimensional reduction of the images or dimensional reduction of the base through the application of the PCA or application of edge detection [8] (that binarizes the images) can minimize these problems. In the same manner, the problem of KNN classification errors can be circumvented by the repetition of the same image classification process.

DISCUSSIONS

The purpose of this article was to describe an adaptive image recognition system, that is, one that recognizes images regardless of their type. Be it ifaces, fingerprints, or medical images, among others, the structure of the recognition system remains the same, with the base organized in folders and images in the folders - and then the manual clustering of the images, the repeated application of the KNN algorithm on the loaded base, the classification, and then the reduction of the problem to determining whether an image belongs to a set of images in a folder and, if necessary, the application of a threshold. Such is the structure of the proposed system. Although the literature is full of criticisms of the use of KNN (for the recognition problem), it actually proves itself to be the most suitable for this activity, due to its simplicity and transparency. There is a lot of room for improvement in this approach, though.

ACKNOWLEDGEMENTS

The results presented in this publication were obtained through Research and Development activities carried out in the Data Center Project with Internet of Things, with resources from Law no. 8.387 / 1991, and any advertising should refer to the aforementioned law according to Art. 48 of Decree 6.008 / 2006, having the company Samsung Electronics of the

Amazônia LTDA funded this Project under the terms of said law.

REFERENCES

ANALYTICS. Available Online at https://help.highbond.com/ helpdocs/analytics/141/user-guide/pt-br/Content/ analyzing_data/summarizing_data/clustering_data.htm.

AT & T. Laboratories Cmabridge. The Datábase of Faces. Available Online at http://camorl.co.uk/facedatabase.html.

BASSEVILLE, M, Distance Measures for Signal Processing and Pattern Recognition, Signal Processing, vol. 18, N.4, pp.349-369, 1989.

BORS, Adrian G. Face recognition using orthodiffusion bases. Conference: European Signal Processing Conference (EUSIPCO), At Bucharest, Romania, Volume: pp. 1578-1582 · August 2012.

CALTECH. Available Online at https://www.researchgate.net/ figure/Figura-4-Exemplo-de-objetos-das-classes-da-base-de-dados-Caltech-101_fig4_312491975

CHOLLET, François. Deep Learning with Python. Manning PUblications Co. 2018.

Documentação do Scikit Learn – KneighborsClassifier. Disponível em: https://scikit-learn.org/stable/modules/ generated/sklearn.neighbors.KNeighborsClassifier.html. Visitado em 06/06/2020.

FVC2006. Available Online at http://bias.csr.unibo.it/ fvc2006/download.asp

GÉRON, Aurélien. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc.", 2017

HU, Li-Yu et al. The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus, v. 5, n. 1, p. 1304, 2016.

HUANG, G.B.; Wang, D. H.; Lan, Y. Extreme learning machines: a survey. International Journal of Machine Learning and Cybernetics, Springer, volume 2, n.2, p. 107–122, 2011.

KIRBY, M. and Sirovich, L. Application of the Karhunen-Loève Procedure for the Characterization of Human Faces, IEEE PAMI, vol. 12, pp. 103-108, 1990.

KOTISIANTIS S.B. Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence. Applications in Computer Engineering. IOS Press Amsterdam, The Netherlands.2007.

KOTSIANTIS, Sotiris B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: A review of classification techniques. 2007.

MELO, Ernande F., Oliveira, Hélio M. A Fingerprint-based Access Control using Principal Component Analysis and Edge Detection. IEEE/SBrT International Telecommunication Symposium, ITS 2010. 2010

MICHALSKI, Ryszard S., Jaime G. Carbonell, and Tom M. Mitchell, eds. Machine learning: An artificial intelligence approach. Springer Science & Business Media, 2013.

PORTAL Data Science. Available Online at https://portaldatascience.com/introducao-a-clusterizacao-e-os-diferentes-metodos/

Portal Data Science. Disponível em: https://portaldatascience. com/o-algoritmo-k-nearest-neighbors-knn-em-machine-learning/. Visitado em: 06/06/2020.

PREETI, Kumar D. Effect of Various Distance Classifiers on the Performance of Bat and CS-Based Face Recognition System. Harmony Search and Nature Inspired Optimization Algorithms pp 1209-122. 2019.

REVSTAT: Statistical Journal.Metrics and scoring: quantifying the quality of predictions. Overview. Volume 12. Number 1. March, 1–20, 2014.

RUUSKA, Salla et alli. Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. Behavioural ProcessesVolume 148, March 2018, Pages 56-62. Elsevier.

SAMBROO, Thomas D. and Goslin, Jeremy.Principal components analysis of reward prediction errors in a reinforcement learning task. NeuroImagem, Voluume 124. Part A, 1 January 2016, Pages 276-286

TURK, M. and Pentland, A. Eigenfaces for Recognition, *Journal of Cognitive. Neuroscience,* vol.3, N.1, pp.71-86, 1991.

WILSON, C. Candela, G. Watson, C. Neural Network Fingerprint Classification, J. Artificial Neural Networks, vol.1, N.2, pp.1-25, 1993.

Y. LeCun, Y. Bengio, G. Hinton. Deep Learning. Nature 521(7), 436-444. 2015.

****