INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 18, Issue: 1, Month: February, Year: 2023
Article Number: 4981, https://doi.org/10.15837/ijccc.2023.1.4981

CCC Publications

Machine learning and uLBP histograms for posture recognition of dependent people via Big Data Hadoop and Spark platform

F. AlFayez, H. Bouhamed

Fayez AlFayez
Department of Computer Science and Information
College of Science, Majmaah University
Al-Majmaah, 11952, Saudi Arabia.
F.alfayez@mu.edu.sa

Heni Bouhamed
Advanced Technologies for Image and Signal Processing
Unit (ATISP)
Sfax University, Tunisia
heni.bouhamed@fsegs.usf.tn

Abstract

For dependent population, falls accident are a serious health issue, particularly in a situation of pandemic saturation of health structures. It is, therefore, highly desirable to quarantine patients at home, in order to avoid the spread of contagious diseases. A dedicated surveillance system at home may become an urgent need in order to improve the patients' living autonomy and significantly reduce assistance costs while preserving their privacy and intimacy.
The domestic fall accident is regarded as an abrupt pose transition. Accordingly, normal human postures have to be recognized first. To this end, we proposed a novel big data scalable method for posture recognition using uniform local binary pattern (uLBP) histograms for pattern extraction. Instead of saving the pixels of the entire image, only the patterns were kept for the identification of human postures. By doing so, we tried to preserve people's intimacy, which is very important in e-health. To our knowledge, our work is the first to use this approach in a big data platform context for fall event detection while using Random Forest instead of complex deep learning methods. Application results of our conduct are very interesting in comparison to complex architectures such as convolutional deep neural networks (CNN) and feedforward deep neural networks (DFFNN).

Keywords: Local Binary Pattern, Hadoop, Spark, Random Forest, surveillance system at home.

1 Introduction

a standing, sitting, or laying posture to a lower reclined one [24]. It is an incidence where someone falls unintentionally to the ground or a lower level [14]. It is generally believed that falls are the greatest risk to the aged and dependent, resulting in serious physical and psychological harm. When monitoring systems can correctly identify a fall incidence and produce an impending alert, they can save lives. A fall typically results in lying motionless on the ground for some time. Thus, this research work attempted to identify five distinct types namely bending, lying down, standing, crawling and sitting.

A number of enticing researchers looked into the use of deep neural networks (of various architectures) for the human postures and actions classification [9, 12, 26]. In brief, Convolutional Neural Networks (CNN) and Long short-term memory (LSTM) are the most tested models in this context in which spatial and temporal features are crucial [21]. Recurrent Neural Networks (RNN) (constructed on LSTMs, GRUs, and other architectures) are among the most used models in times series modeling, whereas CNN Are reputable for their good performance in finding spatial patterns. Some studies have been conducted on the fusion of these two models, either in series [2, 22, 25] or in parallel [8**?**] and results in video analysis have been quite interesting. Human posture recognition is important in fall detection because it is possible to recognize a fall by sequencing action frames [19]. It is critical to understand pose series in order to detect fall events as specific "pose changes." Consider a randomly artificial neural network N and a chosen task Ts that N must solve. It is Non-deterministic Polynomial time Hardness (NP-hard) to determine whether there is a possible solution to solve Ts by testing all the possibilities of N settings (structure and weight). Even with the simplest of its architectures, the Deep FeedForward Neural Networks (DFFNN), we can see the extremely high complexity of deep neural network learning. Given the CNN architecture, the complexity increases even further. This accounts for the formalism's user community tendency to refer to ready architectures based on the field of application, especially when the inputs number is very elevated (for example, the recognition of the content of Full HD images having approximately two million pixels per frame). Although it is undeniably efficient in the field of image recognition, many researches have acknowledged that the CNN architecture is difficult to implement [10, 17, 23, 29].

As far as privacy is concerned, Asif et al. [1] presented the encoding of the human position in the skeleton form, they relied on the CNN structure, which we tried to avoid because it is overly complex. Iazz et al. [15] suggested a three-step procedure: The human shape is selected from the background of an image in the first step; then, both local and global patterns, such as vertical and/or horizontal angles, are selected; finally, a pre-trained Support Vector Machine (SVM) model is used. However, this method, is susceptible to bring out occlusions of images. Authors of [27] employ the same methodology, but focus on the view from the top of the room rather than a side view to surmount the problem of occlusions. However, the traditional machine learning limitations remain the greatest shortcoming facing the works mentioned above. Hamdi et al. used Local Binary Pattern (LBP) histograms to extract features without removing the silhouette, yielding very promising results. Nonetheless, the efficacy of deep learning, even with its classic DFFNN architecture and NP-hard learning complexity, have to be evaluated. This work could be enhanced by calculating the Local Binary Pattern (LBP) histograms before applying the silhouette extraction.

Our research work focused on the position classification of persons in video (images) while bypassing the CNN/DFFNN high computational cost, on the one hand, and using all the image pixels during the learning of a recognition pattern, on the other hand. The foundation of our application would therefore be a scalable big data platform allowing real-time data streaming via spark streaming or/and Apache Kafka. To our knowledge, our conduct is the first to use this approach in a big data platform context, especially with Hadoop-Spark frameworks, for fall event detection while using Random Forest instead of complex deep learning formalisms. We will see that application results of our conduct are very interesting.

The remainder of this paper is organized as follows. Section 2 is dedicated to present our methodology and theoretical background. In Section 3, Images data bases, model selection and performance evaluation are described before presenting our conclusion and perspectives in the fourth and last section.

2 Methodologies, theoretical background and data base

CNNs have strong feature extraction capability that has been adapted to select features from images hyperspectral. Both of the Local Binary Model (LBP) and Local Uniform Binary Model (uLBP) are elementary and robust descriptors for spatial patterns extraction, which enables us to avoid us the use of CNNs or even replace them with another much simpler classification formalism.

2.1 LBP and uLBP

The LBP method assigns a decimal number to each pixel in an image, known as the LBP code, which measures the local structure texture in the vicinity of all pixel one by one [13]. Figure (1a) depicts this process: each pixel is measured relative to its 8 neighbors. Negative values are encoded with 0 while the positive ones are encoded with 1. Binary numbers could be constructed by fusionning all of these binary values from the first pixel in the top left to the right making a full spin. The first upper left neighbor is the starting point. A decimal value is generated from the binary number and then used to label the given pixel. The occurrence frequency of each value is represented by the LBP labels histogram. It can be then exploited as a texture descriptor [3]. The central pixel's direct neighbors may be located at various distances. In fact, the distance separating this central pixel from its direct or indirect neighbors can be a radius of one, or it can be 2 distant units, representing a radius (r) of 2, or it can be three distant units, representing a radius of three (see fig. 1 a). Neighbors number can also change reaching up to 8 with r equal to one or more with a larger radius. For our studies, we have chosen to use a total of 8 neighbors with r = 1, r = 2, and r = 3. In a future research, we intend to test other possibilities with a particularly higher number of neighbors. Several researchers [4, 5, 6, 18, 31**?**] looked into the efficacy of exploiting the LBP histograms to conduct graph-based recognition networks. Pattern can be classified as uniform in case of having a single binary digit change (2).

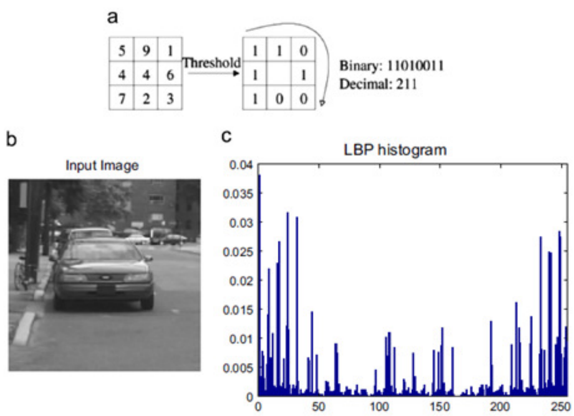

Figure 1: LBP histogram from images pixels.

The LBP is acknowledged to be a powerful nonparametric operator for the description of local features of an image and has been proven to be robustness to translation and rotation. We consider a pixel (x_c, y_c), an arranged binary assembly defined as LBP. It is obtained by measuring the gray value of a pixel (x_c, y_c) with respect of its 8 neighbors' pixels. As a result, the LBP value is expressed as a decimal number:

$$LBP\left(x_c, y_c\right) = \sum_{i=0}^{7} S\left(i_n - i_c\right) 2^n \tag{1}$$

where i_c denotes the gray value of the studied pixel (x_c, y_c). The LBP value was proven to be invariant to any monotonic gray level transformation, and the binary value of the local neighborhood remains unaltered after transformation.

$$S\left(i_n - i_c\right) = \begin{cases} 1, & i_n - i_c \geq 0 \\ 0, & i_n - i_c < 0 \end{cases}$$

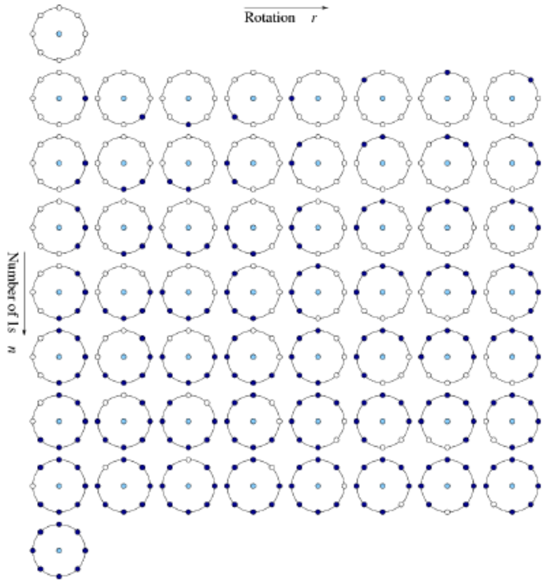

Figure 2: Uniform LBP pattern.

Table 1: Algorithm 1. Our new method for human posture recognition

Algorithm: PostureRecognition
Begin
Transform video images to gray level
For each frame (image) do
Extract Foreground from Background via pixels extraction (Background Image – Image)
Compute LBP for each pixel of extracted silhouette image
Compute the histogram of all uLBP codes
Consider the participant position in the image as a label (class)
Fit a Random Forest model using the image dataset and testing it with test and validation images
End

The LBP value is a group of binary code weighted coefficients. This value depends on the subtraction results between the studied pixel and neighboring ones, each of which has two hundred fifty-six features to describe different texture patterns. To further reduce dimensionality, we considered the uLBP, which has only 59 patterns. It is worthwhile to recall that the uniform LBP may be expressed as follows:

$$uLBP(x_c, y_c) = \sum_{\substack{i=0 \\ \#\Gamma\left(LBP(x_c,y_c)\right) \leq 2}}^{7} S(i_n - i_c)2^n \tag{2}$$

where $\#\Gamma\left(LBP(x_c,y_c)\right) = \left\{LBP(x_c,y_c)_i \neq LBP(x_c,y_c)_{i+1} : i \in [0,6]\right\} + 1$ is the number of blocks consisting of ones and zeros. Furthermore, 3 possible r (r=1,2, and 3) with a number of neighborhoods equal to eight are exploited, resulting in a global number of features of 177 ($n \times n$ pixels) for any image. Our proposed method for human posture recognition is further explained in 3 and Algorithm 1.

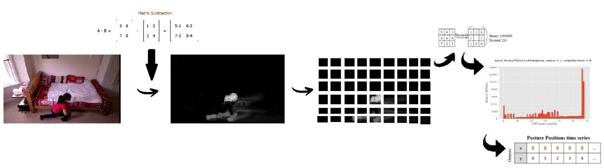

Figure 3: RGB image transformation into uLBP histograms through silhouette extraction.

2.2 Temporal algorithmic complexities

It is obvious from section 2.1. that the processing time complexity of the uLBP is a constant, i.e.:

$$\mathcal{C}_{uLBP} = O(1)$$

Note that the learning Random Forests time complexity cost is not considered in the above formula.

Since the comparison between the CNN complexity and that of the DFFNN has already been presented in Hamdi et al., we focused on the comparison of the latter with the Random Forest having the uLBP histograms as inputs.

According to Hamdi et al., (2021) the complexity of CNN is polynomial whereas that of the DFFNN deep neural network is represented as follows:
Let us consider a DFFNN with M hidden layers and n inputs, where the i^{th} hidden layer is composed of mi neurons, and k output neurons will carry out the following multiplications number (activation functions are excluded):

$$\mathcal{C}_{MLP} = \mathcal{O}(nm_1 + m_M k + \sum_{i=1}^{M-1} m_i m_{i+1}$$

And with only one hidden layer, the number of multiplications becomes:

$$\mathcal{C}_{MLP} = \mathcal{O}(nm_1 + m_1 k)$$

In general, we mainly consider the size of the input when we study the temporal complexity of an algorithm. Nevertheless, the number of multiplications performed in linear combinations (the complexity of time) also depends on the number of hidden layers and the number of neurons in each layer according to the current case. The time complexity of a forward transition of an adjusted DFFNN therefore depends on the architecture. Nonetheless, with our suggested method, there would be a wide variation between using the uLBP (with 177 entries) and x entries representing all the image pixels, which yields:

$$\mathcal{C}_{MLP} = O\left(177i1m_1 + m_M k + \sum_{i=1}^{M-1} m_i m_{i+1}\right) \tag{3}$$

The complexity of the Random Forest (model which would be retained after the tests) depends closely on the complexity of the decision trees which are of the order of $\mathcal{C}_{DT} = \mathcal{O}(n * \log(n) * d)$

Where: n is the training inputs dimension and d is the data.
Hence the number of the Random Forest multiplications is as follows:

$$\mathcal{C}_{DT} = \mathcal{O}(n * \log(n) * d * k)$$

where k is the number of the learned decision trees.
Using uLBPs with 177 inputs instead of using the n inputs would yield a number of multiplications of the order of:

$$\mathcal{C}_{DT} = \mathcal{O}(177 * \log(177) * d * k) \tag{4}$$

It can, therefore, be concluded that in accordance with to the 2 theoretical formulations (3 and 4), our method with the uLBP histograms - Random Forest combination will have a faster execution time than the uLBP histograms + DFFNN. We just need to compare their performance results for posture classification.

2.3 Experimental Data base

We used dataset from Kripesh et al., 2017 [32] (free downloadable from http://www.falldataset.com/). Kinect sensors are used to capture these images (at the quick frame rate of 30 per second so as to detect all of the postures in each image). Nevertheless, furniture might have been changed or removed, or another person might have accessed the frame; these are scenarios we are planning to take into account in our future research. For training, we have five labeled postures. This data was collected using a data set of raw RGB images (640 × 480 in size) recorded by only one Kinect sensor, placed at a 2.4-meter-high roof. The images dataset contains 21499 images, of which 73% (15800) are adopted for training, 15% (3199) for testing, and 12% (2500) for final validation. The dataset images were captured with five different persons in five different decorations and from 8 different capturing angles. 2 are men, aged 32 and 50, and the other three are women, aged 19, 28, and 40. The postures of the participants were restricted to five different poses: standing, sitting, crawling, bending and lying (Figures 4, 5, 6, 7, 8, and 9). We also showed the silhouette extraction image created by subtracting pixels from the original image (at gray level for both). The corresponding uLBP histogram characterizing each image was then plotted as a scalar vector. The images of 2 participants, a 32-year-old man and a 28-year-old woman making a total of 15,800 images were used for the training. The dataset for testing included all the 3199 images of the 32-year-old person taken in different decorations. We used all the 2500 images of the three persons for validation. These photographs were also taken in a different decoration and were not adapted for the model fitting or testing.

Figure 4: Bending position.

Figure 5: Crawling position.

Figure 6: Lying position.

Figure 7: Sitting position.

Figure 8: Standing position.

Figure 9: Empty scene.

3 Machine learning formalism selection, application platform and performance evaluation

During the conventional use of different machine learning formalisms in image classification, the exploited input is generally made up of the images all pixels. The number of input characteristics in this case may be very high, for example, an image having size 1000/600 would be exploited with 600000 input, which can blow up the number of parameters to be calculated throughout training, especially when we deal with big data. Some studies [4, 5, 6, 9, 16, 31] have presented the efficiency of exploiting uniform LBP histograms for fitting graph-based recognition networks. This allows reducing the inputs number considerably by exploiting only the histograms of the uLBP features (whatever of the size of the image). Thus, the only remaining task was to choose the machine learning formalism which will be the most efficient in our context.

3.1 Hadoop and Spark platform

Apache Hadoop is an open source framework used to store and process large data sets. It allows data to be analyzed in parallel on a cluster of multiple computers, rather than on a single machine. This allows a significant speed gain. Four main modules make up Hadoop:

- The HDFS (Hadoop Distributed File System) which is a distributed file system that can be run on standard or low-end hardware. It offers better performance and increases error tolerance compared to traditional file systems,
- YARN (Yet Another Resource Negotiator) which manages and monitors cluster nodes and resource usage. It is also used to plan tasks and jobs,
- The MapReduce framework which helps programs to perform parallel calculations on data,

- Hadoop Common which provides common Java libraries that can be used with many other software in the Hadoop ecosystem.

Thanks to Hadoop, it is easier to use all the storage and processing capacity of clustered servers and perform distributed processing on large volumes of data. This framework provides the building blocks on which applications and services are dependent. Over the years, the Hadoop ecosystem has grown and now includes many tools and applications dedicated to Big Data. Among these we can cite the analytical interface Hive, the non-relational database HBase, the interactive notebook Zeppelin, or the distributed processing system Apache Spark.

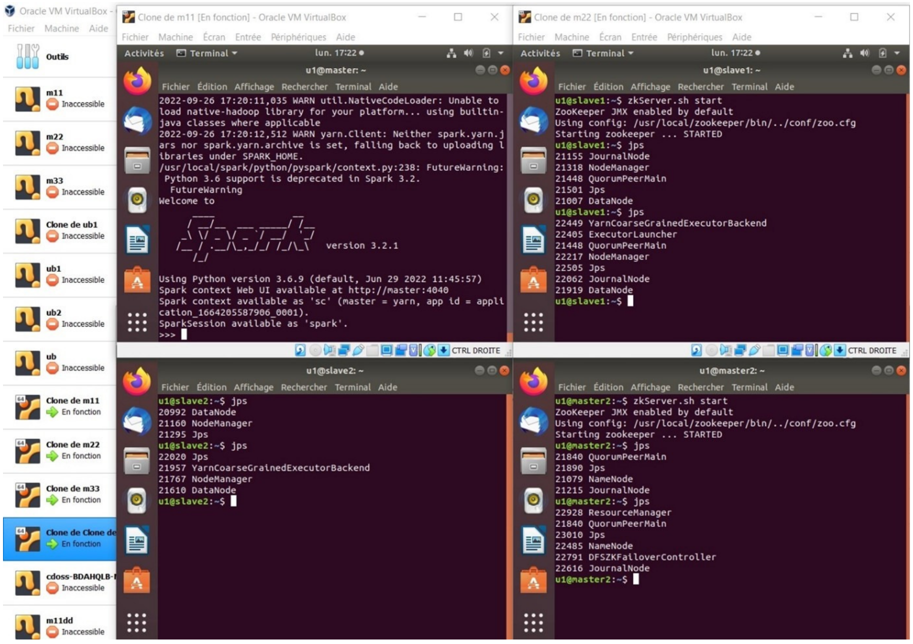

Figure 10: The four virtual machines of our Hadoop-Spark cluster.

Apache Spark is a distributed processing system used for big data workloads. It uses in-memory caching and helps optimize query execution to enable fast queries on data of any size. Simply put, it is a fast engine for big data processing. It offers better performance than previous Big Data tools such as MapReduce. Its runs on a cluster RAM memory, providing faster processing than on hard drives. This general engine can be used for distributed SQL queries, building data pipelines, ingesting data into a database, running Machine Learning algorithms, or working with data streams and graphics. Nowadays, Spark is included with most Hadoop distributions. It has become the main Big Data processing framework, thanks to its speed and very easy-to-use API for developers. The experimental study of our work was carried out on a Hadoop cluster (version 3.3.2) configured based on Ubuntu 18.04 virtual machines via the Oracle VirtualBox 6.1.28 virtualization software. Our cluster (Figure 10) consists of a master machine, a standby machine (backup of the master's metadata) and two slaves (or workers). We installed the Apache Spark 3.2.1 as the distributed engine of our Hadoop cluster. Everything was mounted on a PC with an Octacore CPU (with 16 threads) and 24 Giga bytes of RAM. All the Python-Spark codes developed as part of our work (transformation of images into uLBP histograms, learning and testing of machine learning models) can be downloaded via https://github.com/henibouhamed/Fall2.

3.2 Random Forest selection and configuration

We decided to test the best-known machine learning formalisms on our database of images transformed into a matrix of 178 columns (177 uLBP plus the class variable) on 21499 lines representing all the images (Training, test and validation). We evaluated our conduct in two steps: first, K-fold (k=10) cross-validation was used to evaluate the models with a test dataset (3199 images with new rooms) and then those constructed were tested with a validation dataset (2500 images with unknown

Table 2: Test dataset Results

Method	Accuracy	Precision	Recall
uLBP + Logistic Regression	72,5%	72,4%	72,1%
uLBP + Decision Tree	87,1%	87,1%	87,1%
uLBP + Naïve Bayes	71,2%	71,2%	71,2%
uLBP + Neural Network	94,4%	94,4%	94,3%
uLBP + Random Forest	98,4%	98,4%	98,4%
[9] DFFNN	93,2%	93,14%	91,45%
[18] CNN	81%	-	-

Table 3: Validation dataset results

Method	Accuracy	precision	Recall
uLBP + Logistic Regression	62,4%	62,4%	62,4%
uLBP + Decision Tree	81,7%	81,5%	81,4%
uLBP + Naïve Bayes	61,6%	61,6%	61,6%
uLBP + Neural Network	87,3	87,3%	87,3%
uLBP + Random Forest	92,6%	92,6%	92,6%
[9] DFFNN	86,3%	86,3%	85,7%
[18] CNN	74%	-	-

rooms and unknown people). Tables 1 and 2 show the results obtained by investigating the formalism models we tested on the data of the test and validation images, respectively. The Random Forest was revealed to be the formalism that gave the best results.

3.3 Discussion

Tables 2 and 3 display the recognition results of our approach with Random Forests on the test and validation sets, respectively. It should be noted that we have changed the number of trees introduced as a parameter to the random forest (100 trees adopted at the end of the tests) until reaching a stabilization of the results. To evaluate our method, we opted for comparing our results with previous ones using the same database. The first group of studies [18] used a CNN for the posture classification phase before going on to fall detection. The second set of studies [9] referred to a DFFNN+uLBP without applying a silhouette extraction before the learning and classification phase. Our objective in this work was to show that the use of a simple machine learning formalism, adapted to the problem considered in a big data context may be more advantageous than the use of very demanding architectures in terms of computational cost. First, we managed to avoid the known complexity of implementing and fitting a CNN or even a DFFNN; second, we have succeeded in improving the recognition results for the test dataset from 81% in [32] and 93.2% in [9] to 98.4% and from 74% in [32] and from 86.3% in [9] to 92.6% for data from validation image in terms of Accuracy (percentage of good classification). Moreover, the use of the Random Forests and uLBP histograms allowed us to bypass using all pixels as input to the classifier. Using the uLBP histograms with 3 r (r = 1, 2 and 3) and with eight neighborhoods has given us only 177 entries (59 * 3) which greatly decreased the training complexity of our classifier.

Finally, acquiring only the uLBP histograms instead of saving all the image also allows us to protect the privacy of the dependent persons at home. Besides, the recording will not have to stop even while these people go to the toilets or take a bath without any breach of privacy. Nevertheless, a diminution in accuracy was detected when validating our model on images of new persons in new places with new decors. This can be justified by the difference in body volume of persons in this experiment. On the other hand, the improvement of the classification results with a simpler method (Random Forest) than deep learning architectures (CNN/DFFNN) might be due to the difficulty of configuring the latter.

4 Conclusion

Our study focused on the phase preceding the detection of falls, which requires us to recognize the persons' positions in video frames (images) in a first place. The recognition of various human positions can, then, be exploited to train an LSTM model using a time sequence. Our main objective was to bypass the use of CNN/DFFNN due to their configuration and implementation difficulties. Furthermore, by using only the uLBP histograms, we were able to reduce the number of inputs required to train the Random Forest. Finally, people's intimacy and privacy were protected by saving only the uLBP histogram characteristics rather than the entire information included in the image. However, our conduct has not been tested on real big data in order to show the scalable capabilities of the used Big data frameworks in term of processing speed.
The proposed method increased the recognition accuracy results for the test dataset from 93.2% to 98.4% and the validation dataset from 86.3% to 92.6%. In the future, we trying to find out a novel solution that allows us to detect only the person in question among a group people by extracting the silhouettes of all the people within the scene before calculating LBPs. We also envisage to predict the sequential evolution of person's behavior based on their posture.

Acknowledgment

The author extends their appreciation to the deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IFP-2020-83).

References

[1] Asif U.; Mashford B.; Von Cavallar S.; Yohanandan S.; Subhrajit R.; Jianbin T. & Stefan Harrer S. (2020). Privacy Preserving Human Fall Detection using Video Data. *Proceedings of the Machine Learning for Health NeurIPS Workshop, PMLR*, 116:39-51, 2020.

[2] Delgado-Escaño R.; Castro F. M.; Cózar J. R.; Marín-Jiménez M. J.; Guil N. & Casilari E. (2020). A cross-dataset deep learning-based classifier for people fall detection and identification. *Computer Methods and Programs in Biomedicine*, 184 (2020) 105265.

[3] Dornaika, F.; Bosaghzadeh, A.; Salmane, H.; & Ruichek, Y. (2014). A graph construction method using LBP self-representativeness for outdoor object categorization, *Engineering Applications of Artificial Intelligence*, 36, 294-302,2014.

[4] Dornaika, F.; Bosaghzadeh, A.; Salmane, H. & Ruichek, Y. (2014). Graph-based semi-supervised learning with Local Binary Patterns for holistic object categorization, *Expert Systems with Applications*, 41(17): 7744-7753,2014.

[5] Dornaika, F.; Bosaghzadeh, A.(2015). Adaptive graph construction using data self-representativeness for pattern classification, *Information Sciences*, 325: 118-139,2015.

[6] Dornaika, F.; Moujahid, A.; El Merabet, Y. & Ruichek, Y.(2016). Building detection from orthophotos using a machine learning approach: An empirical study on image segmentation and descriptors. *Expert Systems with Applications*, 58: 130-142, 2016.

[7] Ercolano, G. & Rossi S. (2021). Combining CNN and LSTM for activity of daily living recognition with a 3D matrix skeleton representation. *Intelligent Service Robotics*, 2021, p. 1-11.

[8] Feichtenhofer C.; Pinz A. & Zisserman A. (2016). Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1933–1941.

[9] Hamdi M.; Bouhamed H.; AlGarni A.; Elmannai H. & Meshoul S. (2021). Deep Learning and Uniform LBP Histograms for Position Recognition of Elderly People with Privacy Preservation, *International Journal of Computers, Communications & Control*, 16 (5), 2021, page 1-15, 2021.

[10] He K. & SUN J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 5353-5360, 2015.

[11] Heinrich, S., et al. (2010). Cost of falls in old age: a systematic review, *Osteoporosis international*, 21.6 (2010), 891-902.

[12] Herath S.; Harandi M. & Porikli F. (2017). Going deeper into action recognition: a survey. *Image Vis Comput*, 60, 4-21.

[13] Huang, D.; Shan, C.; Ardabilian, M.; Wang, Y. & Chen, L. (2011). Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6), 765-781, 2011.

[14] Hyndman D.; Ann A. & Stack E.. "Fall events among people with stroke living in the community: circumstances of falls and characteristics of fallers. *Archives of physical medicine and rehabilitation Volume*, 83, Issue 2, 2002, pp 165-170. https://doi.org/10.1053/apmr.2002.28030.

[15] Iazzi, A.; Rziza, M. & Oulad Haj Thami, R. (2021). Fall Detection System-Based Posture-Recognition for Indoor Environments, *J. Imaging* 7, no. 3: 42, 2021. https://doi.org/10.3390/jimaging7030042

[16] Jebara T.; Wang J. & Chang SF. (2009). Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, 441-448. ACM, 2009.

[17] Justus, D.; Brennan , J.; onner, S., et al. (2018). Predicting the computational cost of deep learning models. In*2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018. p. 3873-3882, 2018.

[18] Kripesh A.; Bouchachia H.; Nait-CharifH. (207). Activity Recognition for Indoor Fall Detection Using Convolutional Neural Network. *Fifteenth IAPR International Conference on Machine Vision Applications (MVA) Nagoya University*, Nagoya, Japan, May 8-12, 2017.

[19] Kourtzi Z. & Kanwisher N. (2000). Activation in human mt/mst by static images with implied motion. *J Cogn Neurosci*, 12(1), 2000, 48–55.

[20] Li Z.; Gavrilyuk K.; Gavves E.; Jain M. &, Snoek CG. (2018). Video lstm convolves, attends and flows for action recognition. *Comput Vis Image Underst*, 166, 2018, 41-50.

[21] Mahshid M. & Reza S. (2019). A motion-aware ConvLSTM network for action recognition. *Applied Intelligence*, (2019) 49:2515–2521.

[22] Na L.; Yidan W.; Li F. & Jinbo S. (2019). Deep Learning for Fall Detection: Three-Dimensional CNN Combined With LSTM on Video Kinematic Data. *IEEE Journal Of Biomedical and Health Informatics*, VOL. 23, NO. 1, 2019.

[23] Neshatpour K.; Homayoun H. & Sasan A. (2019). Icnn: The iterative convolutional neural network. *ACM Transactions on Embedded Computing Systems (TECS)*, 2019, vol. 18, no 6, p. 1-27.

[24] Noury N.; Fleury A.; Rumeau P.; Bourke A.; OĹaighin G.; Rialle V. & Lundy J. E. (2007). Fall detection-principles and methods. *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France*, 2007, pp. 1663-1666. doi: 10.1109/IEMBS.2007.4352627

[25] Ordonez F.J. & Roggen D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16(1) 2016, 115.

[26] Poppe R. (2010). A survey on vision-based human action recognition. *Image Vis Comput* 28(6), 976-990.

[27] Ricciuti M.; Spinsante S.& Gambi E. (2018). Accurate Fall Detection in a Top View Privacy Preserving Configuration. *Sensors (Basel)*. 2018 May 29;18(6):1754. doi: 10.3390/s18061754. PMID: 29844298; PMCID: PMC6021973.

[28] Simonyan K. & Zisserman A. (2014). Two-stream convolutional networks for action recognition in videos. In: *Advances in neural information processing systems*, 2014, pp 568–576.

[29] Singh P.; Verma V. K.; Rai P. & Namboodiri V. P. (2019). Hetconv: Heterogeneous kernel-based convolutions for deep cnns. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019. p. 4835-4844.

[30] Srivastava N.; Mansimov E. & Salakhudinov R. (2015). Unsupervised learning of video representations using lstms. In: *International conference on machine learning*, pp 843–852.

[31] Wright J.; Yang AY.; Ganesh A.; Sastry SS. & Ma Y. (2009). Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*. 31(2): 210-227.

[32] Wu J. M.; Li Zh., Herencsar N., et al. (2021). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems*, 2021, p. 1-20.

[33] Xingjian S.; Chen Z.; Wang H.; Yeung D-Y.; Wong W-K. & Woo W-C. (2015). Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: *Advances in neural information processing systems*, 2015, pp 802–810.

Cite this paper as:

AlFayez, F.; Bouhamed, H. (2023). Machine learning and uLBP histograms for posture recognition of dependent people via Big Data Hadoop and Spark platform, *International Journal of Computers Communications & Control*, 18(1), 4981, 2023.
https://doi.org/10.15837/ijccc.2023.1.4981