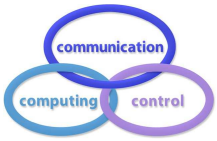
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 16, Issue: 5, Month: October, Year: 2021
Article Number: 4256, https://doi.org/10.15837/ijccc.2021.5.4256



CCC Publications

Deep Learning and Uniform LBP Histograms for Position Recognition of Elderly People with Privacy Preservation

M. Hamdi, H. Bouhamed, A. AlGarni, H. Elmannai, S. Meshoul

Monia Hamdi
Information Technology, College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University, Saudi Arabia
mshamdi@pnu.edu.sa

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

Abeer AlGarni
Information Technology, College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University, Saudi Arabia
adalqarni@pnu.edu.sa

Hela Elmannai
Information Technology, College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University, Saudi Arabia
hselmannai@pnu.edu.sa

Souham Meshoul
Information Technology, College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University, Saudi Arabia
sbmeshoul@pnu.edu.sa

Abstract

For the elderly population, falls are a vital health problem especially in the current context of home care for COVID-19 patients. Given the saturation of health structures, patients are quarantined, in order to prevent the spread of the disease. Therefore, it is highly desirable to have a dedicated monitoring system to adequately improve their independent living and significantly reduce assistance costs. A fall event is considered as a specific and brutal change of pose. Thus, human poses should be first identified in order to detect abnormal events. Prompted by the great results achieved by the deep neural networks, we proposed a new architecture for image classification based on local binary pattern (LBP) histograms for feature extraction. These features were then saved, instead of saving the whole image in the series of identified poses. We aimed to preserve privacy, which is highly recommended in health informatics. The novelty of this study lies in the recognition of individuals' positions in video images avoiding the convolution neural networks (CNNs) exorbitant

computational cost and Minimizing the number of necessary inputs when learning a recognition model. The obtained numerical results of our approach application are very promising compared to the results of using other complex architectures like the deep CNNs.

Keywords: Fall detection, Deep Learning, Classification, Deep Feed Forward Neural Network, Local Binary Pattern Histogram.

1 Introduction

Falling may be defined as an involuntary or unexpected body position change from a standing or sitting or lying position to an inferior inclined position [1]. A fall is an incident that causes a person to come to stay on the ground or other inferior levels involuntarily [2]. Overall, falls are among the greatest dangers to older adults with significant physical, financial, and emotional damage. Falls represent a heavy economic burden to society, with fall-related costs ranging from 0.85% to 1.5% of health care total expenditures in the United States, the United Kingdom, and Australia [3]. Furthermore, the falls rate is increasing in the elderly especially in those over 65 years old. Although fall incidents cannot be totally prevented, monitoring systems may save lives, when they can accurately recognize a fall incident and generate an imminent alert. This system must be capable of discerning between fall events and regular activities. This is a very problematic task because some everyday activities like lying down or sitting abruptly on the couch or switching from a standing position to lying down have very high similarities to falls incidents. Generally, a fall ends with an inactivity period on the floor in a lying down pose. Therefore, in this article, we tried to recognize five different activities that are sitting, standing, lying down, crawling, and bending.

There is a variety of interesting studies comparing deep neural networks application for the recognition of human action [4, 5]. Among these, CNNs and long-term memory networks (LSTMs) are the two most commonly used networks for this application where temporal and spatial information are very important [6]. Recurrent neural networks (RNNs) (based on LSTM, Gated recurrent unit (GRU), and other architectures) are dominant tools for the understanding of information changes over time, while CNNs are very well known for their impressive ability to extract spatial information. The merger of these two networks, either in parallel [11, 12] or in series [7, 8, 9, 10], has been the subject of some studies allowing quite interesting results in video analysis. In addition to the extracted temporal and spatial information, motion features, generally optical flow features, are usually used to improve performance. A small number of studies, all in a preliminary stage, present the merging of CNN and LSTM networks into a unified network, where temporal and spatial data are supported at the same time [13, 14]. The first proposal to use the LSTM to take advantage of the benefits of the CNN layers was introduced in paper [13] and this merger was further discussed and developed in [15, 16].

Human posture recognition is a key factor in this context of fall detection with deep learning since it allows the recognition of a fall from the sequencing frames of actions [17]. Knowledge of poses series is very important to detecting events of fall as specific "change of pose". To this end, we considered an arbitrary artificial neural network R and an arbitrary task T to be solved by R. The problem that arises here is whether there is a solution that adequately solves T in the space of all the R parameters (its weights and structure). Such a problem is clearly NP-hard. The algorithmic complexity of learning a deep neural network is very high even with the classic Deep Feed Forward Neural Network (DFFNN) architecture, especially where the number of inputs is very large. The image content classification is a perfect illustration of this case since with the currently used cameras, the number of pixels to be taken into account for a single image is very high (Full HD for example (1080p or 1080i) = 1920×1080, or approximately 2 million pixels per frame). Despite their undisputed effectiveness in image recognition, many studies have mentioned the difficulty of implementing the CNN architecture [18, 19, 20, 21]. Moreover, many users of this formalism prefer to rely on an already existing and validated architecture, depending on the context, with a slight adaptation possibility.

Regarding privacy preservation, Asif *et al.* [22] considered the encoding of the human posture in the form of a skeleton representation. The authors approach in this study is based on a CNN structure.

However, this structure is known for its complexity, which discouraged us from using it. Iazz *et al.* [23] proposed a three –stage approach, In the first step, the human silhouette is extracted from the image background. Then, global and local features such as horizontal and vertical angles are extracted. In the last step, a pre-trained Support-vector machine (SVM) classifier is used. This approach, however, is sensitive to image occlusions and Ricciuti *et al.* [24], focused on the top view (roof view), instead of the front one, in order to overcome the occlusion issue. Both cited works suffer from the traditional machine learning limitations.

Our work addressed the recognition of individuals' positions in video images avoiding the CNNs exorbitant computational cost and the use of all image pixels when learning a recognition model. The results generated by the classification would be used as time sequences for training another LSTM model or other times series for fall detection. Our contributions are:

- Bypass the use of CNNs given their difficulty in both configuration and implementation
- Minimize the number of necessary inputs for learning the classic DFFNN architecture
- Preserve privacy. Our approach is founded on the use of the LBP histograms to characterize each frame, which limits the number of inputs to a few hundred elements at most.

The remainder of this paper is organized as follows. Section 2 was dedicated to present our methodology and theoretical background. Datasets, model parametrization and performance evaluation were described in Section 3 before presenting our conclusion and perspectives in the last section.

2 Methodologies and theorical background

CNNs have powerful characteristic extraction capability, and have been used to extract characteristics from the hyperspectral image. The LBP and uniform LBP (uLBP) are simple but powerful descriptors for spatial features, which can reduce the workload of CNNs and improve the classification accuracy.

2.1 LBP and uniform LBP (uLBP) methods

LBP labels each pixel in an image with a decimal number, called LBP code, which quantifies the local structure at the neighborhood of each pixel [25]. This operation is shown in figure (1a): each pixel is compared to its eight neighbors; the resulting positive values are coded with 1, and the other ones with 0. A binary number is obtained by concatenating all these binary values clockwise with respect to each given pixel. The starting point is the first upper left neighbor. The decimal value that represents the generated binary number is subsequently used to label the given pixel. The LBP label histogram is the frequency of occurrence of each value. It is computed over a region or an image and can be used as a texture descriptor [26].

The neighbors of the central pixel can be the direct neighbors with a radius equal to one or in other cases the neighbors with two distant units which represents a radius equal to two or even to three distant units of neighboring pixels which represents a radius equal to three. The number of neighbors may also vary; it can be eight at most with a radius equal to one and more with higher radii. For our work, we chose to use a number of neighbors equal to eight with the first three possible radii; radius = 1, radius = 2, and radius = 3). As a future perspective, we are planning to test other combinations especially with a larger number of neighbors. Several studies [27, 28, 29, 30, 31, 32] have presented the effectiveness of using LBP histograms (Figure 1) to learn graph-based classification models. Figure 2 shows the cases where the pattern can be uniform with only one change of the binary digits.

The LBP is a forceful non-parametric tool for the interpretation of particular image characteristics and has been settled to be invariant with rotation. Considering a central pixel (x_c, y_c), an ordered binary set presented as LBP is procured by comparing the value of gray of the center pixel (x_c, y_c) with the its eight neighbors pixels. Hence, the LBP code is communicated as the decimalized form an

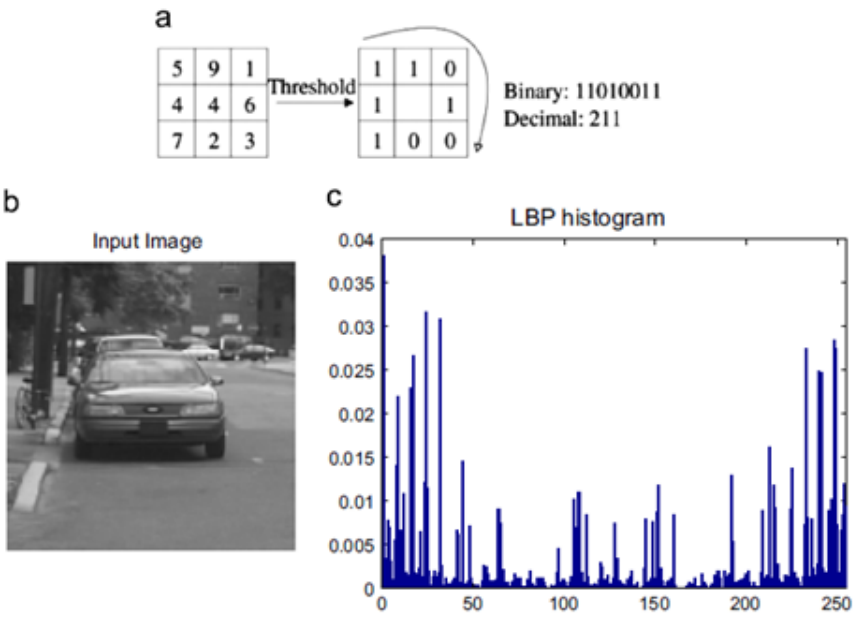


Figure 1: From pixels to the LBP histogram

octet binary number:

$$\mathrm{LBP}\,(x_c, y_c) = \sum_{i=0}^{7} S\,(i_n - i_c)\,2^n.$$

To note that i_c represents the value of gray of the center pixel (x_c, y_c), and i_n represents the gray value of its eight neighbors pixels. The code of LBP has been settled to be changeless to any monotonous transformation of gray level, and the local neighborhood binary code remains unmodified after transformation.

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

Thus, the binary value set can be obtained with the corresponding binary code, and the LBP code is a weighted coefficient set of binary codes. The LBP code only depends on the difference values between the central pixel and the neighboring ones. This code has 256 patterns to reflect a variety of texture information. In our approach and in order to further reduce dimensionality, a uLBP with only 59 patterns was considered. It is worth reminding here that the uLBP can be defined as follows:

$$\mathrm{uLBP}\,(x_c, y_c) = \sum_{\substack{i=0 \\ \sharp\Gamma(\mathrm{LBP}_{(x_c,y_c)})\leq 2}}^{7} S\,(i_n - i_c)\,2^n,$$

where $\sharp\Gamma\,(\mathrm{LBP}\,(x_c, y_c)) = \left\{i \in [0,6] : \mathrm{LBP}\,(x_c, y_c)_i \neq \mathrm{LBP}\,(x_c, y_c)_{i+1}\right\} + 1$ denotes the number of blocks formed by zeros and ones. In addition, three possible radii ($r = 1, 2$, and 3) with a number of neighborhoods equal to 8 are used, so the total number of patterns for any image (n×n pixels) is equal to 177. Algorithm 1 and Figure 3 introduce our proposed method for position recognition (in Figure 3, DFFNN is presented with an arbitrary architecture and the resulting table first line represents the detected postures codes and the second line represents the images index in the video sequence).

2.2 Time complexities

It is clear from the previous sub-section that the time complexity of the processing LBP method is a constant, i.e.

$$\mathcal{C}_{\mathrm{LBP}} = \mathcal{O}(1).$$

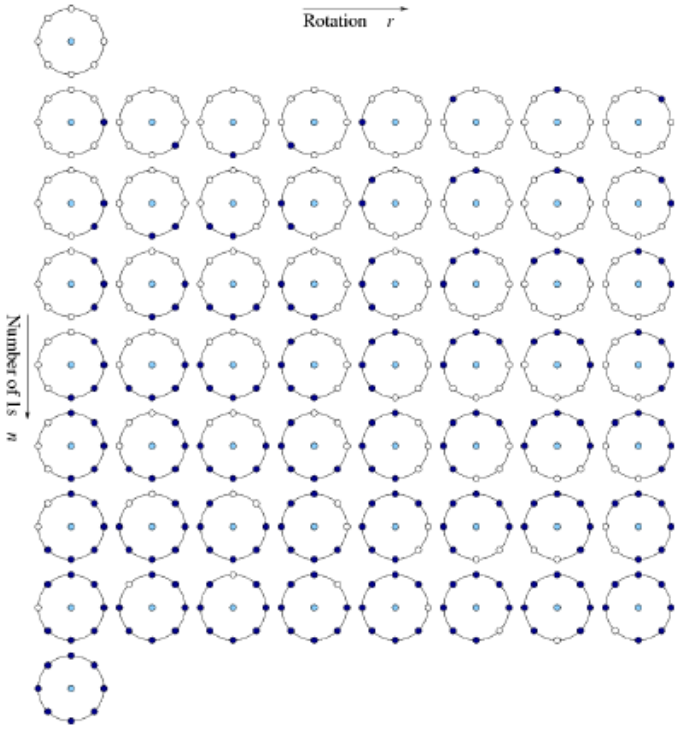


Figure 2: Uniform LBP pattern

Algorithm 1 Our proposed method for Position-Recognition

Transform video frames to gray level
for each image **do**
  Compute LBP for each pixel
  Compute the frequency histogram of all LBP values
  Add the position label of the participant in the image as a class
  Learn a DFFNN using the training image dataset and testing it with test and validation images
end for

It should be noted that the time complexity cost of feedforward neural network is not involved in the above formulation.

On the other hand and in order to compare the complexity of our method with the most used methods, the total complexity of all convolutional layers in CNN architecture are presented as:

$$\mathcal{C}_{\text{CNN}} = \mathcal{O}\left(\sum_{\ell=1}^{d} n_{\ell-1} s_l^2 n_\ell m_\ell^2\right),$$

where

- ℓ is the index of a convolutional layer.
- d is the depth (number of convolutional layers).
- n_ℓ is the number of filters (or features detectors) in the ℓ-th layer.
- $n_{\ell-1}$ is also known as the number of input channels of the ℓ-th layer.
- s_ℓ is the spatial size (length) of the filter.
- m_ℓ is the spatial size of the output feature map.

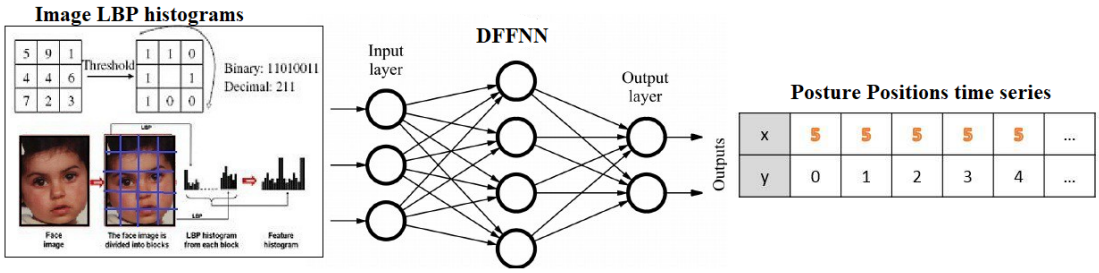


Figure 3: LBP histograms with the DFFNN for posture recognition

According to the previous expression, the CNN complexity is polynomial, which requires high computational capacity. In general, the notations (s, n) are adopted in such a way that they respectively represent the filter size and the number of filters. In this case, we can say that the complexity is a 4th degree polynomial or quadratic with respect to each hyper-parameter. The time complexity cost of fully connected layers and pooling layers is not involved in the above formulation.

At this point, we can conclude that according to these two theoretical formulas our approach will be faster in its execution time and thus remains very competitive compared to CNN method.

The complexity of a Deep Feed Forward Neural Network (DFFNN) can be summarized as follows: In general, a DFFNN has n inputs and M hidden layers. The ith hidden layer contains m_i hidden neurons, and k output neurons. These latters will perform a certain number of multiplications (excluding activation functions) and can be written as

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

If we have just one hidden layer, the number of multiplications becomes

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

In general, when analyzing the time complexity of an algorithm, we do it with respect to the size of the input. However, in this case, the time complexity (more precisely, the number of multiplications involved in the linear combinations) also depends on the number of layers and the size of each layer. The time complexity of a forward pass of a trained DFFNN is thus architecture-dependent. However, in our case there will be a difference between using an uLBP with 177 inputs and n^2 inputs without using any image processing methods.

3 Datasets, parametrization and performance evaluation

3.1 Datasets

We used images from [32] available for download from the link `http://www.falldataset.com/`, these images are recorded with the Kinect sensor with a frame rate of 30 frames per second. As a result, the images are quickly captured in order to detect all postures for each frame. However, it is possible for furniture to be changed or moved or even for another person to enter the frame; we are planning to consider these scenarios in a future work. We have five labeled poses available for training. A raw RGB image dataset (with a size of 640x480) that was recorded by a single uncalibrated "Kinect" sensor that was used to collect this data. This sensor was used with a roof height of 2.4 meters. The dataset has 21499 images, of which 15800 images were used for training, 3199 images for testing, and 2500 images for final validation. The images in the dataset are of five different participants and were recorded in five different rooms, with eight different viewing angles. There are two male participants with an age range from 32 to 50 years and three female participants aged respectively 19, 28, and 40 years. The participants' activities were limited to five different poses: sitting, standing, bending, lying,

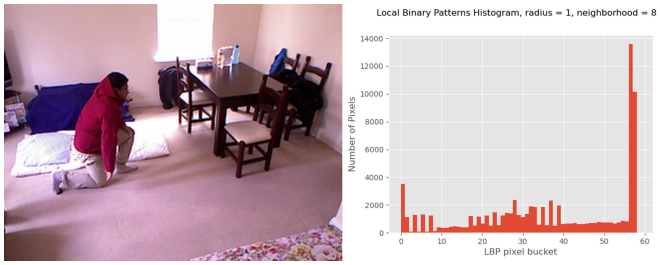
and crawling (Figure 4). We also plotted the corresponding LBP diagram representing each image as a scalar vector. We used the images of two participants; the 28-year-old female and the 32-year-old male, which represents a total of 15800 images for learning. The test dataset has 3199 images of the 32-year-old participant but was in a different room from the others. The images of the 3 participants including a 50-year old man and the 2 women aged 19, and 40 years were used for the validation set. These images were also recorded in a different room and were not used during the learning or test processes. We did not perform any segmentation for the person extraction and we did not address the possibility of the presence of two or more people in the scene, these scenarios are to be considered in the future work. LBP histograms act according to the change of textures in the images and since only the person moves while the decoration of the room remains the same, we preferred to simplify the application of our approach by avoiding the segmentation and extraction of the person.

When using different deep learning architectures in image recognition, the often used input is the individual pixels forming the images. The number of input features can be quite large, e.g., an image of size 1000/500 will be considered as an input with 500,000 entries, which might cause the number of parameters to be computed during learning to increase exponentially, especially with the presence of massive data. Several works [27, 28, 29, 30, 31, 32] have proven and defended the effectiveness of using uLBP histograms for learning graph-based classification models. This conduct reduces the number of inputs to 59 when using only uniform LBP patterns (regardless of the image size).

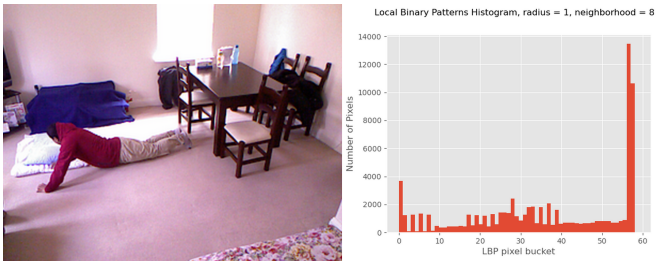
3.2 DFFNN parameterization

For the DFFNN parameterization, we were inspired by the work of Darren Cook [33] to try to find the best possible architecture of our DFFNN for the studied database. We tested the following combinations (which gave good results with the famous MINIST digit classification database): 200-200 (Figure 5), 512-512 (Figure 6), 1024-1024 (Figure 7), 2048-2048 (Figure 8), 400-800-800 (Figure 9), 1024-1024-2048 (Figure 10), 200-200-200-200 (Figure 11), 300-400-500-600 (Figure 12), 1024-1024-1024-2048 (Figure 13) and 2048-2048-2048-4096 (Figure 14). To note that the use of uniform LBP histograms with three radii ($r = 1, 2$, and 3) and 8 neighborhoods gave us 177 inputs in the input layer. Nevertheless, as participants' activities were limited to five different poses: sitting, standing, bending, lying, and crawling with an empty scene (Figure 4), the output layer presents 6 neurons.

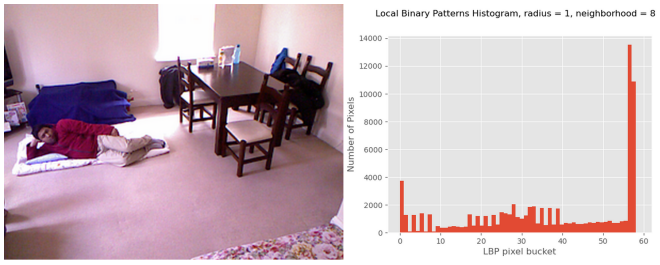
Our methodology consists in starting from two hidden layers with a reduced number of neurons (200-200). By progressively increasing the number of hidden layers and the number of neurons, we evaluated the gain in terms of classification accuracy. According to our tests, the results are satisfactory from



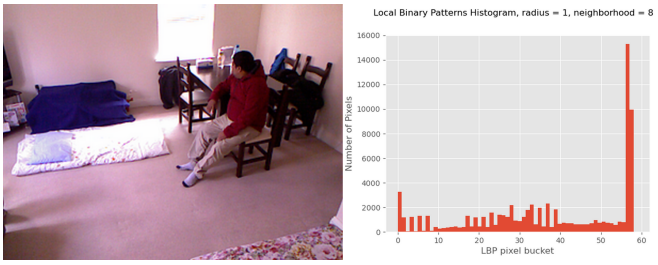
(a) Bending position



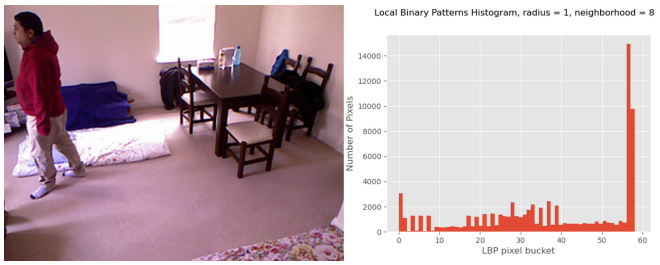
(b) Crawling position



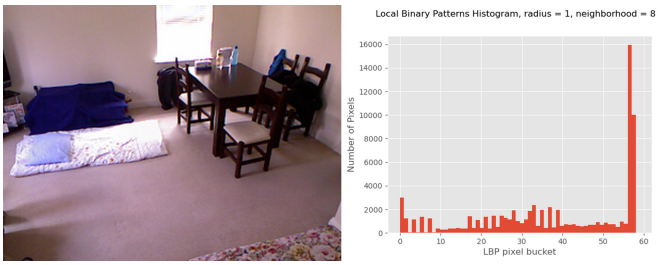
(c) Lying position



(d) Sitting position



(e) Standing position



(f) Empty scene

Figure 3: Our five labeled poses and empty scene

the configuration in the two hidden layers of 2048 each. The subsequent increase in terms of layers did not really improve the results noticeably by testing the different combinations of neurons with three (Figures 9 and 10) and four layers (Figures 10-14). The performance indicators in terms of accuracy (number of correct predictions divided by number of total predictions), precision and recall (see Figure 15 for more details concerning precision and recall formulas) suggest that it would be better to keep two layers only. The 2048-2048 architecture model was adopted to predict the test and validation database. This model includes two layers with 2048 neurons each. It should be noted here that the ReLu activation function was adopted for the hidden layers since the results deteriorated when the Sigmoid and Tanh functions were tested. Large neural nets trained on relatively small datasets can overfit the training data and in this context, our dataset includes an important number of images (21499) but with a limited number of scenarios that involve only: five different rooms, eight different viewing angles and five different participants. Consequently, overtraining may occur. Among the most used methods to avoid this phenomenon, we may cite « Dropout » which is a method to regularize approximation of training a huge number of neural networks in parallel with different architectures. During training, a certain number of output layers are haphazardly omited or « dropped out ». By doing so, we temporarily remove it along with all its incoming and outgoing connections from the network. We have chosen to dropout 20% of nodes from each layer. This is justified by the fact that this figure is a common value of dropout used in practice to avoid overfitting without really disturbing the general classification accuracy [34, 35].

One epoch denotes that each sample in the training dataset has had the possibility to upgrade the internal parameters of model. Commonly, the number of epochs is large, usually hundreds or even thousands, authorizing the learning algorithm to run till the error from the model has been enough minimized. We have found examples of epochs number in the literature and in tutorials set to 10, 100, 500, 1000, and more [36]. We have chosen to test our model with a large number of epochs (1000) in order to diagnose whether the model has over learned, under learned, or is suitably fit to the training dataset. Annotated samples of our python codes (developed in python 3.8 with Tensor flow Backend) are available at: `https://github.com/henibouhamed/Fall`.

The suggested method in this paper was evaluated in two steps: firstly, with labeled test data (3199 images with known persons and new rooms) and secondly with labeled validation data (2500 images with unknown people and unknown rooms). Tables 1 and 2 illustrate accuracy, precision, and recall obtained by inferring the DFFNN model on test and validation images data, respectively.

	Accuracy	Precision	Recall
Our approach	93.2%	93.14%	91.45%
[32] (CNN)	81%	-	-

Table 1: Accuracy, Precision and recall for 3199 test images

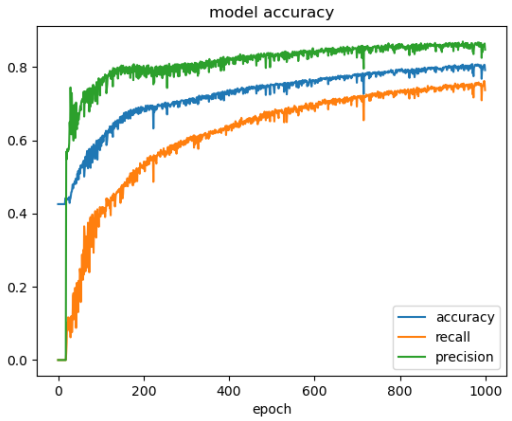


Figure 4: DFFNN training 200-200

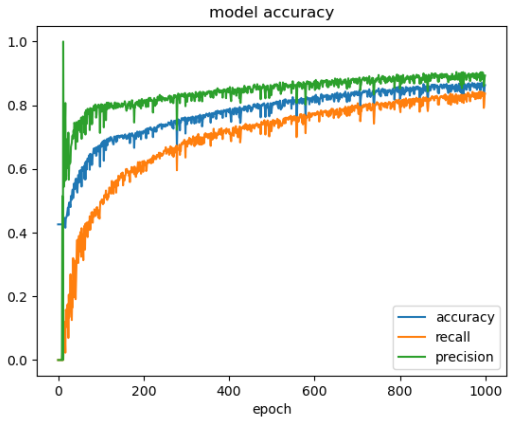


Figure 5: DFFNN training 512-512

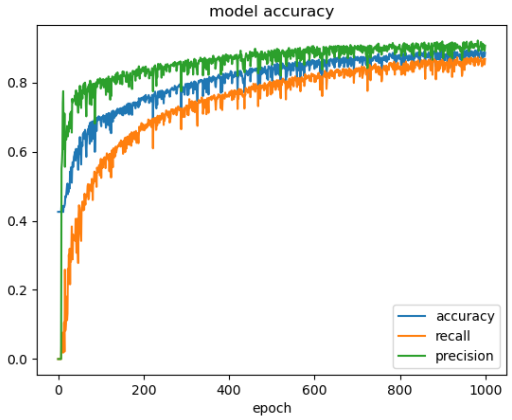


Figure 6: DFFNN training 1024-1024

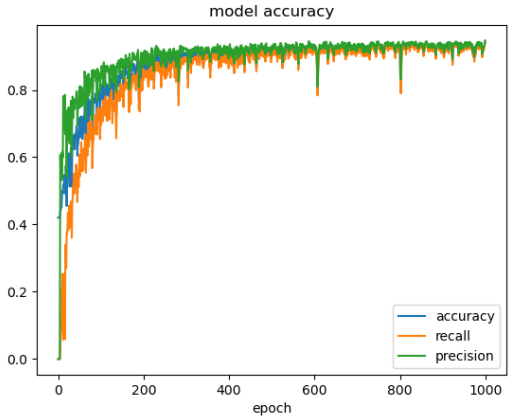


Figure 7: DFFNN training 2048-2048

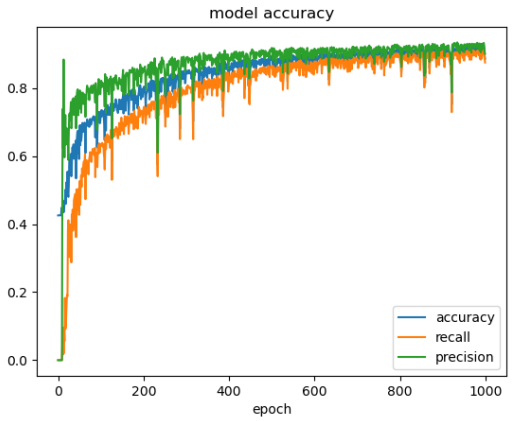


Figure 8: DFFNN training 400-800-800

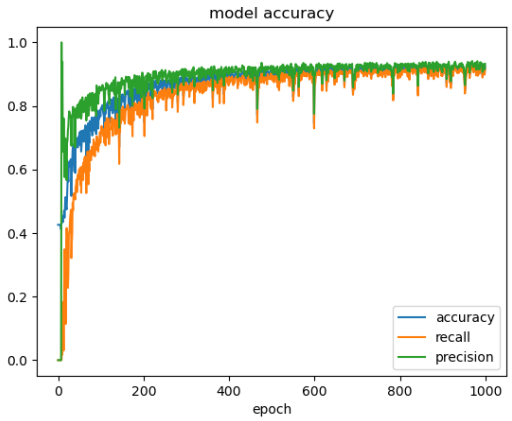


Figure 9: DFFNN training 1024-1024-2048

	Accuracy	Precision	Recall
Our approach	86.3%	86.3%	85.7%
[32] (CNN)	74%	-	-

Table 2: Accuracy, Precision and recall for 2500 validation images

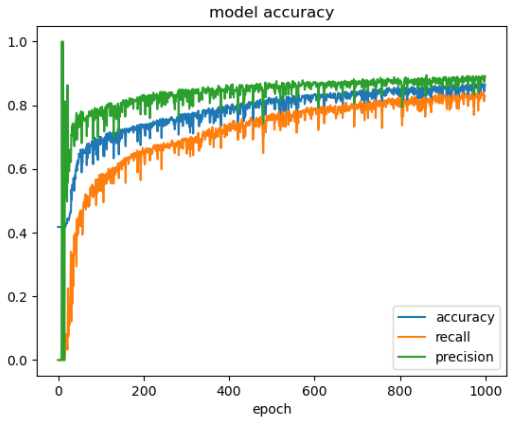


Figure 10: DFFNN training 200-200-200-200

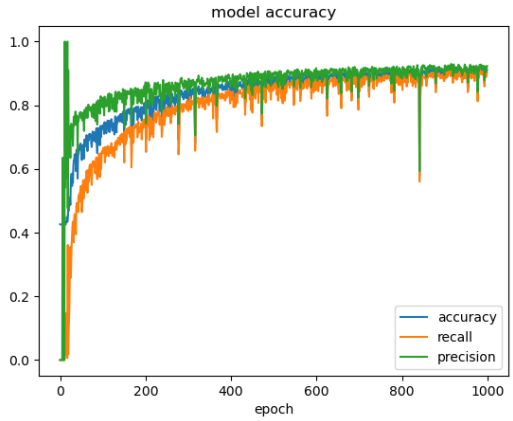


Figure 11: DFFNN training 300-400-500-600

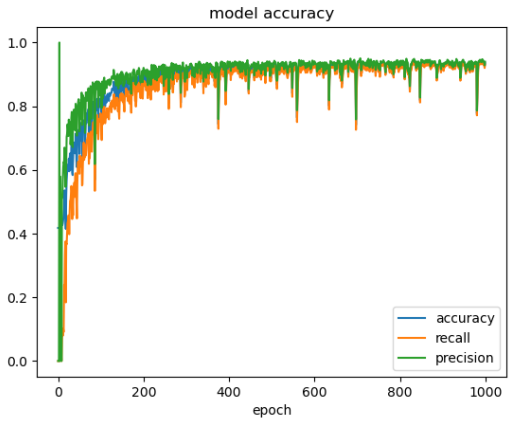


Figure 12: DFFNN training 1024-1024-1024-2048

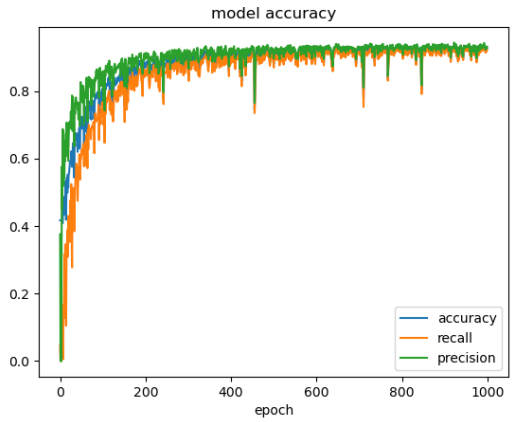
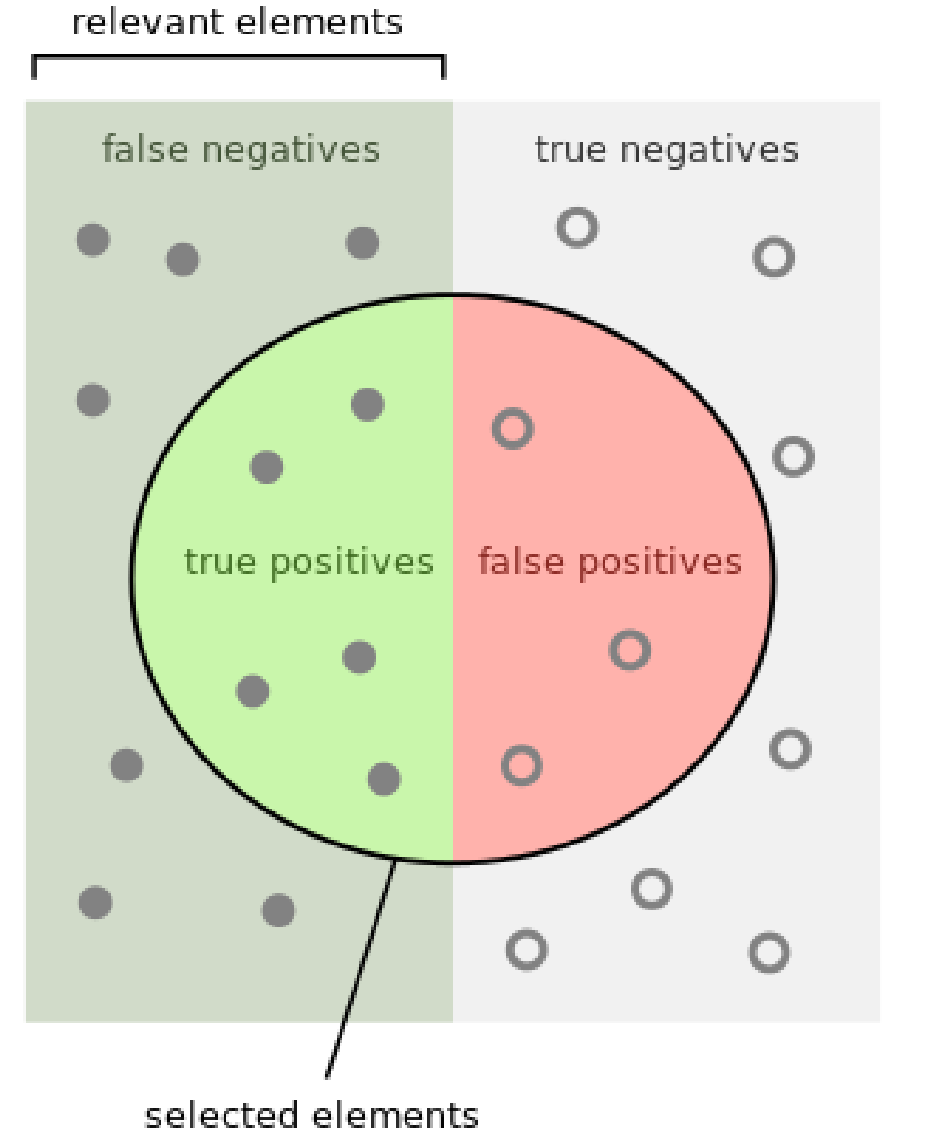


Figure 13: DFFNN training 2048-2048-2048-4096



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Figure 14: Precision, recall and f1-measure formulas [37]

3.3 Performance evaluation

Tables 1 and 2 illustrate the classification accuracy, precision and recall achieved by our model on respective tests and validation sets. We chose to test our approach on the same image database from a previous work where the authors used a CNN for the position recognition phase before moving on to fall detection [32]. Our goal in this work was to show that using a lightweight structure, adapted to the considered problem can be more advantageous than using high demanding architecture in terms of computational complexity. Firstly, we have avoided the known complexity of setting up and learning a CNN, secondly, we have managed to improve the classification accuracy results from 81% to 93.2% for the test image data and from 74% to 86.3% for the validation image data.

Moreover, using the DFFNN and LBP histograms enabled us to avoid the use of all the pixels as input to the model, which would have required 640 x 480 inputs (pixels) for each image processed during our study. The use of uniform LBP histograms with three radii ($r = 1, 2$, and 3) and 8 neighborhoods gave us 177 inputs only (59 x 3) which considerably reduced the learning complexity of our model.

Finally, the acquisition of the LBP histograms only instead of storing all the image information also allowed us to preserve the privacy of elderly people at home and the recording will be able to work even when taking a bath or being in the toilet without any privacy violation. However, a decrease in accuracy can be noticed when testing our model on images of new individuals in new rooms. This can explained by the non-segmentation of individuals from the backend decor. The achieved results in this case could be improved when segmenting people, which will allow handling the presence of more than one person in the same room.

4 Conclusion

Our research work addressed the fall detection issue, which requires in a first step, the recognition of the individuals' positions in video images. The classification of different human postures can be then used as time sequences for training an LSTM model. Our main motivation was to avoid the use of CNNs, given their configuration and implementation difficulties. Moreover, we succeeded in minimizing the number of necessary inputs for learning the classic DFFNN architecture through the use of uniform LBP histograms only, and finally, we took into account the respect of people's privacy and intimacy by only saving LBP histograms features and not all images information.

Our proposed architecture improved the classification accuracy results which increased from 81% to 93.2% for the test image data and from 74% to 86.3% for the validation image data. As a future perspective, we intend to apply individual segmentation before position recognition, as many people may be present in the scene. We also intend to take into consideration the sequential evolution of the individuals' behavior according to their posture position.

Funding

The authors would like to thank the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research through project number PNU-DRI-RI-20-026.

Author contributions

Methodology and software: Heni Bouhamed; formal analysis and validation: Monia Hamdi and Abeer AlGarni; writing: Hela Elmannai and Souham Meshoul; project administration: Abeer AlGarni; funding acquisition: Monia Hamdi.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Noury, N.; Fleury, A.; Rumeau, P.; Bourke, A. K.; Laighin, G. O.; Rialle, V.; Lundy, J. E. (2007, August). Fall detection-principles and methods, *In 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1663–1666, 2007.

[2] Hyndman, D.; Ashburn, A.; Stack, E. (2002). Fall events among people with stroke living in the community: circumstances of falls and characteristics of fallers, *Archives of physical medicine and rehabilitation*, 83(2), 165–170, 2002.

[3] Heinrich, S.; Rapp, K.; Rissmann, U.; Becker, C.; König, H. H. (2010). Cost of falls in old age: a systematic review, *Osteoporosis international*, 21(6), 891–902, 2010.

[4] Herath, S.; Harandi, M.; Porikli, F. (2017). Going deeper into action recognition: A survey, *Image and Vision Computing*, 60, 4–21, 2017.

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

[6] Majd, M.; Safabakhsh, R. (2019). A motion-aware ConvLSTM network for action recognitions, *Applied Intelligence*, 49(7), 2515–2521, 2019.

[7] Srivastava, N.; Mansimov, E.; Salakhudinov, R. (2015, June). Unsupervised learning of video representations using lstms, *In International conference on machine learning*, 843-852, 2015.

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

[9] 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, 105265, 2020.

[10] Lu, N.; Wu, Y.; Feng, L.; Song, J. (2018). Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data, *IEEE journal of biomedical and health informatics*, 23(1), 314–323, 2018.

[11] Simonyan, K.; Zisserman, A. (2014, December). Two-stream convolutional networks for action recognition in videos, *In Proceedings of the 27th International Conference on Neural Information Processing Systems*, 1, 568—576, 2014.

[12] 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*, 1933–1941, 2016.

[13] Xingjian, S.H.I.; 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*, 802–810, 2015.

[14] Li, Z.; Gavrilyuk, K.; Gavves, E.; Jain, M.; Snoek, C.G. (2018). Video lstm convolves, attends and flows for action recognition, *Computer Vision and Image Understanding*, 166, 41–50, 2018.

[15] Ercolano, G.; Rossi, S. (2021). Combining CNN and LSTM for activity of daily living recognition with a 3D matrix skeleton representation, *Intelligent Service Robotics*, 14(2), 175–185, 2021.

[16] Wu, J.M.T.; Li, Z.; Herencsar, N.; Vo, B.; Lin, J.C.W. (2021). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators, *Multimedia Systems*, 1–20, 2021.

[17] Kourtzi, Z.; Kanwisher, N. (2000). Activation in human MT/MST by static images with implied motion, *Journal of cognitive neuroscience*, 12(1), 48–55, 2000.

[18] Justus, D.; Brennan, J.; Bonner, S.; McGough, A.S. (2018, December). Predicting the computational cost of deep learning models, *In 2018 IEEE international conference on big data*, 3873–3882, 2018

[19] Neshatpour, K.; Homayoun, H.; Sasan, A. (2019). Icnn: The iterative convolutional neural network, *ACM Transactions on Embedded Computing Systems*, 18(6), 1–27, 2019.

[20] He, K.; Sun, J. (2015). Convolutional neural networks at constrained time cost, *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 5353–5360, 2015.

[21] 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*, 4835–4844, 2019.

[22] Asif, U.; Mashford, B.; Von Cavallar, S.; Yohanandan, S.; Roy, S.; Tang, J.; Harrer, S. (2020, April). Privacy preserving human fall detection using video data, *In Machine Learning for Health Workshop*, 39–51, 2020.

[23] Iazzi, A.; Rziza, M.; Thami, R.O.H. (2021). Fall Detection System-Based Posture-Recognition for Indoor Environments, *Journal of Imaging*, 7(3), 42, 2021.

[24] Ricciuti, M.; Spinsante, S.; Gambi, E. (2018). Accurate fall detection in a top view privacy preserving configuration, *Sensors*, 18(6), 1754, 2018.

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

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

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

[32] Adhikari, K.; Bouchachia, H.; Nait-Charif, H. (2017, May). Activity recognition for indoor fall detection using convolutional neural network, *In 2017 Fifteenth IAPR International Conference on Machine Vision Applications*, 81–84, 2017.

[33] Cook, D. (2016). *Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI*, O'Reilly Media, Inc., 2016.

[34] Kever, I.; Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, 15(1), 1929–1958, 2014.

[35] Brownlee, J. (2018). *Better deep learning: train faster, reduce overfitting, and make better predictions*, Machine Learning Mastery, 2018.

[36] Brownlee, J. (2016). *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*, Machine Learning Mastery, 2016.

[37] Bouhamed, H.; Ruichek, Y. (2018). Deep feedforward neural network learning using Local Binary Patterns histograms for outdoor object categorization, *Advances In Modelling And Analyses B*, 61(3), 158–162, 2018.

Copyright ©2021 by the authors. Licensee Agora University, Oradea, Romania.
This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.
Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

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), 4256, 2021.
https://doi.org/10.15837/ijccc.2021.5.4256