High-Level Descriptors for Fall Event Detection Supported by a Multi-Stream Network

Original Scientific Paper

Sarah Almeida Carneiro
Institute of Computing, University of Campinas
Campinas, SP, 13083-852, Brazil
sarah.alcar@gmail.com

Silvio Jamil Ferzoli Guimarães
Computer Science Department, Pontifical Catholic University of Minas Gerais (PUC Minas)
Belo Horizonte, MG, 30535-065, Brazil
sjamil@pucminas.br

Hélio Pedrini
Institute of Computing, University of Campinas
Campinas, SP, 13083-852, Brazil
helio@ic.unicamp.br

Abstract *– The need for assertive video classification has been increasingly in demand. Especially for detecting endangering situations, it is crucial to have a quick response to avoid triggering more serious problems. During this work, we target video classification concerning falls. Our study focuses on the use of high-level descriptors able to correctly characterize the event. These descriptor results will serve as inputs to a multi-stream architecture of VGG-16 networks. Therefore, our proposal is based on the analysis of the best combination of high-level extracted features for the binary classification of videos. This approach was tested on three known datasets, and has proven to yield similar results as other more consuming methods found in the literature.*

Keywords *– Video Classification, Multi-stream Network, Fall Detection, High Level Features, Convolutional Neural Network*

1. INTRODUCTION

Abnormal situations can be characterized as actions that fall outside the scope of a given context. Therefore, an example of such, can be the identification of endangering situations. A condition that can be considered a risk, especially for older people, are falls. Due to the weakening of the body's physical structures, an elderly person's fall can lead to various other problems that can contribute to more severe outcomes. Accordingly, it is critical the fast and correct identification of these actions to avoid further complications.

A way of determining some of these abnormal actions is through the analysis of videos from surveillance cameras. Thus, the identification of these cases in videos is being facilitated by the use of computational resources. The sole employment of a surveillance camera operator is not always as assertive as an on-going surveillance algorithm. This can be observed since people can get easily distracted, unlike a machine.

The use of neural network approaches has been playing an important role in action identification in videos. There are many branches considering these studies, one of them is video classification. Video classification is the verification of the existence of a given action, in a group of actions, in an analyzed input video.

This study, an extension of Carneiro et al. [1], still influenced by the identification of falls, focuses on the binary classification of videos. Since many works use information that is directly linked to the RGB information, such as the video frame itself, our proposal is based on the use of data only generated from a high-level descriptor extraction. It is a concern of ours that the RGB can influence the classification behavior of the network when dealing with the generalization of cases.

The use of a specific dataset, that is influenced by camera noise, actors, furniture and others, can make the network not as assertive as expected if the frame information is solely used. In addition, since high-level descriptors might not be as assertive as an RGB information for a given dataset, we also focus on the merge of three of these descriptors: (i) optical flow; (ii) visual rhythm; and (iii) pose estimation. Although some of these features have already been used in other works, their combina-

tion is a novel approach. This combination can provide us with complementary temporal, spatial and rhythmic (temporal-spatial) information of the video without having to rely on the raw frame information itself.

This descriptor ensemble was thought of as a three-stream VGG-16 [2] architecture known as a multi-stream. With this multi-stream, we were able to verify the best complementary stream combination for a video classification concerning falls. Furthermore, by avoiding the use of RGB, we are able to conceal the identity of the people in the analyzed videos as well as to observe that the combination of lesser information descriptors can provide as good results as the features commonly used in the literature.

This text is organized as follows. In Section 2, we discuss some of the recent works associated with fall detection in videos. In Section 3, concepts used in this work are clarified. In Section 4, the proposed methodology is explained. In Section 5, we describe the experiments performed and compare the achieved results to other published methods. Finally, some concluding notes and suggestions for future work are presented in Section 6.

2. RELATED WORKS

Given the aging process, it is possible to notice a weakening of various body structures. Associated with this, it is observable that as well as reflexes, balance can also be affected. Therefore, these issues allied to a number of other factors can be responsible for the occurrence of falls. In addition, since recovery for this portion of the population might not be as fast, falling situations might lead to aggravated injuries. Thus, it is imperative to have access as quickly as possible to aid. Hence, several researches associated with computational resources, related to wearable sensors, video processing and machine learning, have been conducted to try to identify these falls faster.

A study associated with video information was conducted by Lin et al. [3] regarding the extraction of motion vectors and DC+2AC images. This information is used as input to a Global Motion Estimator (GME) to cluster global and local motion. Based on this clustering, falls can be identified by the analysis of the person's centroid, the vertical projection histogram value, and the event duration. There is also research related to identifying falls based on silhouette recognition. These studies use this data as input to statistical models, such as Hidden Markov Model (HMMs), to determine the occurrence of falls [4][5].

It has also become important for these studies to conceal the subject's identity. Accordingly, privacy can be achieved by blurring, silhouetting, covering the object with graphical shapes, among other strategies. A common element that is also associated with action detection is background subtraction. Thus, a subtraction method in conjunction to head tracking algorithms was proposed by Yu et al. [6] to identify falls. As observed, head tracking is also useful in this scenario, Yu et al. [7] was able to correlate this data to density calculation methods, with a mixture Gaussian model for fall detection. Similarly, with a Gaussian model, Rougier et al. [8] was able to use a feature of human body deformity to cope with falling. Attempts for detection were also made by using a subject's bounding box surrounding angulations to train a K-Nearest Neighbor (KNN) [9].

Depth information also started to be considered among detection studies. The RGB-D (RGB image and analogous depth image) were used to extract features, such as Histogram of Oriented Gradients (HOG), Optical Flow (OF) and target skeletons. These features were served to a Support Vector Machine (SVM) to be able to classify fall events [10]. It has also been a concern to determine whether a learning algorithm has a good performance associated with action detection. Thus, a comparative research was conducted to verify distinct learning structures associating them to falls [11].

Kwolek and Kepski [12] demonstrated that, in given circumstances, the KNN algorithm could provide better results compared to the SVM algorithm. The optical flow is regularly associated with a VGG for classification scenarios. By using it, Nunez-Marcos et al. [13] classified falls with a three-stage transfer learning method. Naive-Bayes has also been used as an alternative for detection, employing as data Bag-of-Words from silhouette oriented volumes (SOV) strategy [14]. Alternative studies used a modified CNN AlexNet architecture allied to transfer learning applied to fall detection for surveillance videos [15]. The use of curvelet transforms and an SVM for the identification of human postures are also used with a hidden Markov model (HMM) to classify the existence of falls in videos [16].

3. THEORETICAL BACKGROUND

In this section, we will clarify some of the basic knowledge necessary for understanding this work.

3.1. Optical Flow

The optical flow is a high level feature descriptor able to outline an approximation of a possible movement between video frames (Figure 1). This identified movement can, for example, be caused by an object displacement or a camera shift. Therefore, the algorithm's objective is to highlight frame information that was relocated from a previous frame to its following. Hence, we are able to obtain only the moving object information in the image, thus, being able to discard every other extra static information in the frame.

Let $I(x,y,t)$ be a pixel in a video's first frame $f1$ and d_T the elapsed time between the first frame and its next compared frame $f2$. Considering that the pixel $I(x,y,t)$ was relocated by a distance (d_x, d_y), based on the condition that the pixels of $f1$ and $f2$ have equivalent static intensities, we are able to use Equation 1, where x, y reference the coordinate of a pixel at the t positioned frame. Then, after applying a Taylor series approxima-

tion, it is possible to calculate the optical flow with Equation 2 through 6. There are two possible ways to calculate the optical flow of a set of frames, by a sparse or a dense methodology.

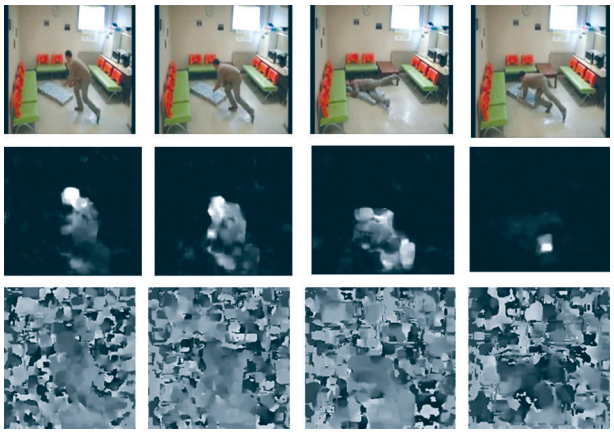

Fig. 1. Frame examples from the FDD dataset and their optical flow extraction example in y and x components, respectively.

A sparse application will generate the optical flow information for selected pixels that can be considered relevant for an object, for instance edges. The dense method, on the other hand, calculates the optical flow per pixel. Despite being a more expensive approach, our study used the Farnebäck [17] algorithm to generate the video's optical flow filtered frames, which is a dense optical flow calculation. This means that the flow vectors are calculated for each pixel. However, this choice was made once we

$$I(x,y,t) = I(x+dx, y+dy, t+dt) \quad (1)$$

$$\mathrm{I}(x+dx, y+dy, t+dt) \approx \mathrm{I}(\mathrm{x},\mathrm{y},\mathrm{t}) + \frac{\partial \mathrm{I}}{\partial \mathrm{x}}\mathrm{dx} + \frac{\partial \mathrm{I}}{\partial \mathrm{y}}\mathrm{dy} + \frac{\partial \mathrm{I}}{\partial \mathrm{t}}\mathrm{dt} \quad (2)$$

$$fx = \frac{\partial \mathrm{I}}{\partial \mathrm{x}} \quad fy = \frac{\partial \mathrm{I}}{\partial y} \quad ft = \frac{\partial \mathrm{I}}{\partial \mathrm{t}} \quad (3)$$

$$fx\,dx + fy\,dy + ft\,dt = 0 \quad (4)$$

$$u = \frac{dx}{dt} \qquad v = \frac{dy}{dt} \quad (5)$$

$$fx\ u + fy\ v + \ ft\ = 0 \quad (6)$$

3.2. Visual Rhythm

The visual rhythm of a frame-set is a descriptor capable of providing spatio-temporal information about the video [18-20]. Although not visually intuitive, this descriptor, by gathering information of each individual video frame, is able to concatenate this information to generate an outcome that is composed of a single image capable of summarizing the video used as input.

This descriptor can be built based on a variety of methods. Therefore, different outcomes can be generated depending on the approach used.

A first approach can be by using histograms. Consider a video $V=(f_t)_{t\in[0,T-1]}$ in the $2D + t$ domain, $D = \{0,...,M-1\}$ x $\{0,...,N-1\}$, where M and N are the width and height of the frame. The variable f_t is a progression of frames and T is the total number of frames in V. A visual rhythm ϑ can be produced, for example, by assembling a sequence of histograms $(H_{ft})_{t\in[0,T-1]}$ calculated based on all the frames of a video. ϑ is defined in Equation 7, where $Z\in[0,L-1]$ and $t\in[0,T-1]$, such that T is the number of frames and L the number of histogram bins. Therefore, the result is a 2D representation of the combination of all frame histograms, where each column of ϑ represents a frame histogram. This process is exemplified in Figure 3.

$$\vartheta(t,z) = H_{ft}(z) \quad (7)$$

Another technique that can be used to generate a visual rhythm is by sub-sampling [21][22]. Hence, the visual rhythm, in domain 1D+T, is a rendition of the video $V=(f_t)_{t\in[0,T-1]}$ in which each frame f_t has their pixels sampled into a column of ϑ (Equation 8). Accordingly, $Z\in[0,....,h_{\vartheta-1}]$ and $k\in[0,....,w_{\vartheta-1}]$, where z, k, r_x and r_y are the height and the width of ϑ, and the ratios of pixel sampled from f_t respectively. In addition, A and B relate to the shifts on each frame, from where the algorithm can initiate the pixel sampling (Figure 3).

$$\vartheta(k,z) = f_t(r_x \times z + a, r_y \times z + b) \quad (8)$$

Although the output of these algorithms is a single image for a video, in the attempt of gathering more spatio-temporal information, we modified the algorithm so that it could produce a visual rhythm for each frame. Thus, we are able to define how many frames we intend to use to create a visual rhythm that is associated with f_t. Consequently, considering a window of 5 frames, we use f_t plus its 5 following frames to calculate a visual rhythm and associate it with f_t, and so on until the last frame.

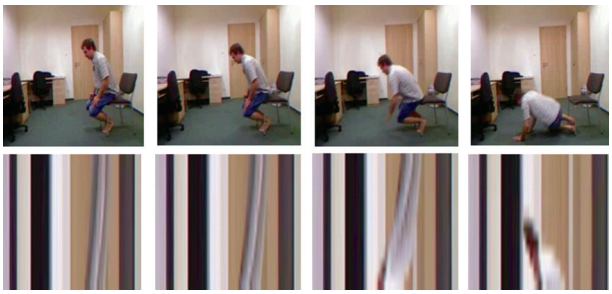

Fig. 2. First row: Frame examples from the URFD dataset Second row: The frame visual rhythm extraction examples.

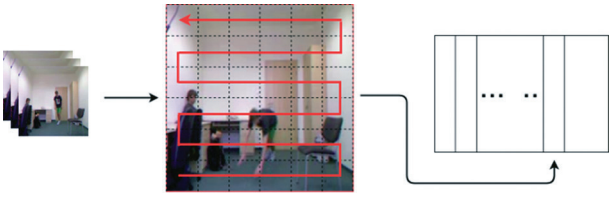

Fig. 3. Visual rhythm algorithm example. The red arrow is demonstrating a possible way of pixel sampling of a frame. The information gathered is placed as a column of the visual rhythm output image.

3.3. Pose Estimation

The pose estimator used during this work is a descriptor able to detect a humanoid and determine its main joints. Similar to the optical flow, the pose estimation will conceal unnecessary information for a classification study (Figure 4).

For this descriptor, we used the algorithm proposed by Cao et al. [23]. The approach receives a video frame that is then used on a two-branch CNN feed-forward network, that is, networks that do not return the information calculated by a layer to a previous one. Thus the information runs through the network until it reaches and becomes an output. An example is a multi-layer perceptron, that is able to predict sets of 2D confidence maps (S) of body member positioning (where might a body member be positioned in the frame) and 2D vector fields (L) (that calculates the degree of association between these members).

The set $S = (S_1, S_2, ..., S_J)$, where J is the number of body parts found, and $L = (L_1, L_2, ..., L_C)$, where C is the number of limbs, are parsed by bipartite matching greedy inference so they can be associated. Lastly, the algorithm produces the 2D key-points indicating the posture of all human components of the frame.

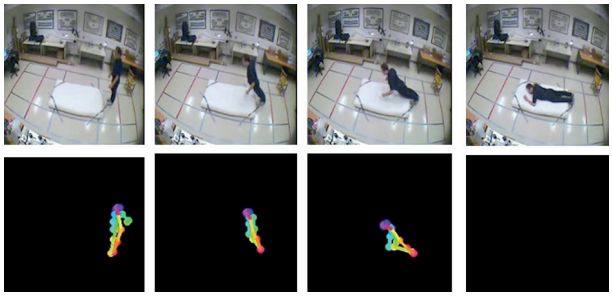

Fig. 4. Frame examples from the Multicam dataset and their pose estimation examples.

3.4. Multi-Stream Architecture

The multi-stream architecture [28-30] is a learning algorithm based on the ensemble of individual learners, each learner is considered a stream (Figure 5). The ensemble can be made based on some different approaches, such as an average of the individual results, the use of a support vector machine (SVM), and others [24]. The use of a multi-stream makes it possible for the network to learn each of the high-level descriptors without disregarding a descriptor as unimportant. Another feature is that, with multi-stream, we are able to evaluate the use of all descriptor combinations. In other words, considering the employment of three high-level descriptors, we can analyze the results of a combination one by one, two by two, and all three.

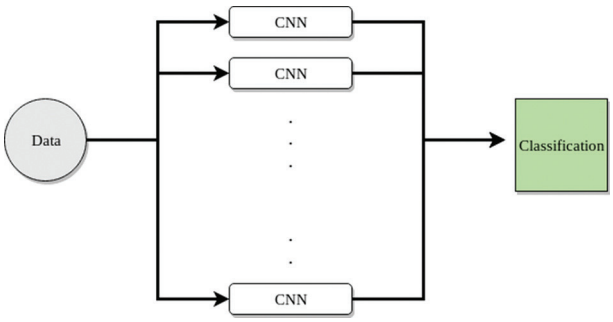

Fig. 5. Multi-stream architecture example.

4. MULTI-STREAM FALL DETECTION APPROACH

Since previous studies regarding abnormal action detection in videos have used standard descriptors for classification, our goal was to verify if the use of weaker feature descriptors could provide as satisfying classification results concerning fall detection. The purpose of this is based on the fact that, by discovering good descriptors, we are able to manage potentially faster or, even, more specialized descriptors for classifying each abnormal situation depending on its need.

The methodology proposed in this work is based on an ensemble of three VGG-16 streams. We acknowledge that the use of the VGG-16 might be outdated, however, it is a classical method and have yielded satisfying results regarding classification in past studies. In addition, since the goal was to verify the relevance of feature combination rather than faster learning algorithms we decided to keep this architecture for a primary result analysis. The features chosen for this study were the optical flow, the visual rhythm, and the subject's pose estimation. Even though some of these features have already been used in past literature works, the proposition of their combination is, to our knowledge, novel.

4.1. High-Level Descriptor Extraction

In this work, we focused on three high-level descriptors: (i) Optical Flow; (ii) Pose Estimation; (iii) Visual Rhythm. As an initial step of our method, the videos from each dataset needed to go through the investigated descriptor algorithms. This pre-processing step guarantees the original frame information filtering according to our need of temporal, spatial or spatial-temporal data. From each frame, these extractions generate other 2D image information (Figures 1, 2 and 4). These images will then serve as inputs to what we acknowledge as a high-level feature extraction.

4.2. High-Level Feature Extraction

After obtaining the images provided by the descriptors, each of these image groups (Optical Flow, Pose Estimation, Visual Rhythm) go through a modified VGG as it can be observed in Figure 6. It should be noticed that

this modified CNN is not a part of the stream learning structure. It is mainly used to avoid explicit feature engineering, since it is not intuitive or easy to determine which are the best features to extract and to ensure the method's independence. The extracted feature vectors acquired by the modified VGG serve as weights for the following step, fine-tuning.

4.3. Individual Streams

Our work was based on the feature extraction of three descriptors. Therefore, the learning model needed to have three individual streams. Each of these streams go through a three step learning process: (i) multi-stream pre-training; (ii) fine-tuning; and (iii) ensembling.

In this work, the falling data provided by the datasets did not optimally contribute with the information required by our network. Therefore, more data was needed to train this learning model. Thus, the initial step was to train the first 14 layers of each of the VGG-16 on ImageNet [25] and, later, the UCF101 dataset [26]. This makes it possible for the network to understand the image's basic structures as well as to identify movement and sequences.

To guarantee that the network would learn the falling action itself, as a second step, we then froze these pre-trained layers and fine-tuned the remaining two. This fine-tuning procedure was associated with the feature vectors extracted (in Section 4.2) from each of the previously presented descriptors. Hence, since we implemented three streams, each individual stream was linked to a specific descriptor. In other words, each stream was fine-tuned with only one of the calculated feature vectors. The third and final step is the ensemble of the results of the individual classifications to combine their isolated outcomes regarding the best combination of features for fall classification.

4.4. Classification

To provide the final classification verdict, as mentioned in the previous section, we used ensembling techniques. To analyze which type of ensemble would provide the best results for our classification, we used three different techniques: (i) average and threshold; (ii) average and a support vector machine (SVM); and (iii) continuous values and SVM.

The first technique, average and threshold, computed the average based on the sum of each stream output and compared it to an empirically defined class threshold. The second ensembling method, on the other hand, had this class threshold defined and adjusted based on the assistance of an SVM. Finally, the continuous values and SVM classified the input regarding a generated vector with each of the streams output so that an SVM could find its separation region.

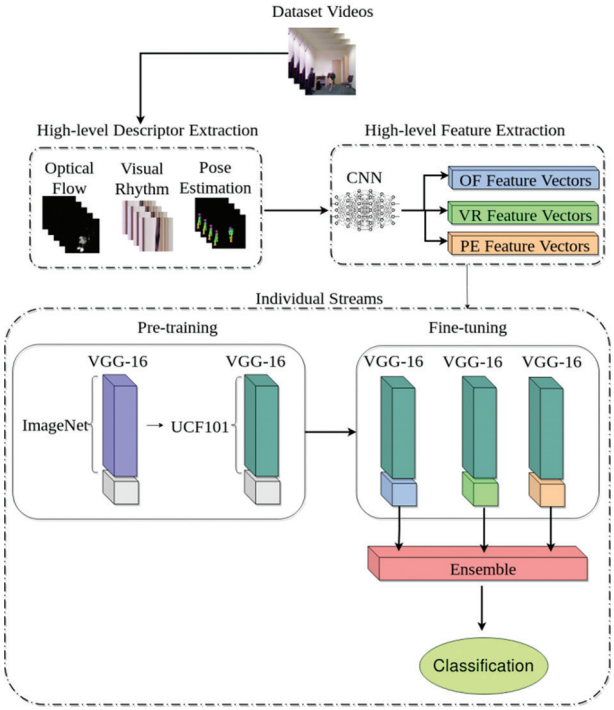

Fig. 6. Multi-stream architecture for fall detection

To analyze the impact of features from high-level descriptors regarding falls, all possible combinations of streams were explored in this work. Therefore, we organized the stream ensembling considering single, two by two, and all three stream results (Figure 7).

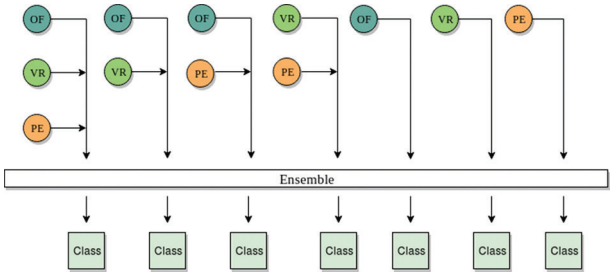

Fig. 7. Feature stream combination.

5. EXPERIMENTS

In this section, we briefly describe the datasets used in our experiments, as well as the quantitative and qualitative results.

5.1. Dataset

Our method was evaluated on three well known falling sets throughout the literature: FDD, URFD [27] and Multicam datasets. These datasets are focused on binary falling and not falling classes. The URFD dataset is composed of 70 videos: (i) 30 videos of falls; and (ii) 40 videos displaying diverse activities. FDD disposes of 191 single camera surveillance videos from both elderly home environments, and office rooms. The Multicam contains 24 scenarios, in which 22 have falling situations, recorded with 8 differently positioned video cameras around the room.

5.2. Quantitative Results

Our quantitative analysis was based on sensitivity, specificity and accuracy metrics. They were chosen since previous studies validated their works using them and we can better evaluate the outputs of our learner. Accuracy is a metric for the model performance evaluation that correlates both positive and negative classes and measures how accurate are the learning results (Equation 9). In Equations 9 to 11, variables *TP, TN, FP, FN, P, N* stand for true positives, true negatives, false positives, false negatives, total number of positive, and total of negative samples, respectively.

$$A=(TP+TN)/(P+N) \quad (9)$$

Specificity is a metric that provides information related to, given a negative example, the probability of a result being negative (Equation 10). Sensitivity, also known as recall, is, given a positive example, the classification result being indeed positive (Equation 11). These metrics were chosen since we did not use balanced accuracy to evaluate our method.

$$Specificity = TN/(TN+FP) \quad (10)$$

$$Sensitivity=TP/(TP+FN) \quad (11)$$

To correctly train our network considering the action of falling, since the entire positive video can contain many other different actions, we cropped only the moment in which a fall occurred. Therefore, so that the network was not trained with what we consider a negative situation (walking, sitting, running, etc), we narrowed the training videos to only the period of frames containing a fall. This reduced the training set, however, it was a way to ensure that the training was not corrupted with false cases.

Considering that the training data was diminished and the not falling cases contained a lot more data, to try to balance this situation we applied a random downsampling of the dataset's negative class to match the size of the positive class. Later on, experiments were conducted considering an 80% training set and 20% testing set.

The results for our multi-stream for all of the descriptor combinations for the FDD dataset concerning the different ensemble methods explained in Section 4.4 can be seen in Tables 1, 2 and 3, respectively.

The best result obtained for the FDD dataset shown in Tables 1, 2 and 3 were yielded upon experiments with a five fold cross-validation, learning rate of 10^{-6}, a mini-batch of 1024, 500 epochs and the batches were normalized. It was possible to see that the use of an SVM during the ensemble technique was beneficial for the classification of the FDD dataset. Considering the average and threshold (Table 1) as the baseline of our method, we are able to observe that the use of the SVM both with the average (Table 2) and continuous values (Table 3) had an improvement concerning the combination of weaker features such as the visual rhythm and the pose estimation. Both the results for Table 2 and 3 were slightly similar, however, when using an SVM with the continuous values, the sensitivity was enhanced by almost 5% in a three stream combination.

Table 1. Results for the FDD dataset with the average and threshold.

	FDD avg		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	42.00%	92.00%	96.80%
Multi-Stream (OF+PE)	85.00%	99.00%	98.66%
Multi-Stream (OF+VR)	23.00%	100.00%	95.73%
Multi-Stream (PE+VR)	29.00%	100.00%	96.09%
Single-Stream (OF)	69.00%	90.00%	98.22%
Single-Stream (PE)	100.00%	25.00%	29.21%
Single-Stream (VR)	13.00%	100.00%	95.20%

Table 2. Results for the FDD dataset with the average and SVM.

	FDD SVM/avg		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	74.00%	100.0%	98.57%
Multi-Stream (OF+PE)	84.00%	100.0%	98.84%
Multi-Stream (OF+VR)	71.00%	100.00%	98.40%
Multi-Stream (PE+VR)	32.00%	100.00%	98.49%
Single-Stream (OF)	77.00%	100.0%	98.22%
Single-Stream (PE)	100.00%	24.00%	28.24%
Single-Stream (VR)	26.00%	100.00%	95.91%

Results for the URFD dataset can be seen in Tables 4, 5 and 6. Similar to the FDD results, the experiments were conducted with a five fold cross-validation, a mini-bath of 1024, learning rate of 10^{-6}, 500 epochs and normalized batches. We have observed that the use of a smaller learning rate helped to improve the results of both of the URFD and FDD datasets that had a smaller amount of positive cases to use while in training. In addition, the accuracy values also had an improvement with the use of an SVM in the classification. The enhancement was made upon sensitivity values as well, as it can be seen that Table 6 had an 11% growth compared to Table 5 in a three-stream case combination.

Table 3. Results for the FDD dataset with an SVM.

	FDD SVM		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	82.00%	99.81%	98.84%
Multi-Stream (OF+PE)	82.00%	99.43%	98.49%
Multi-Stream (OF+VR)	77.00%	99.90%	98.66%
Multi-Stream (PE+VR)	24.00%	100.0%	95.82%
Single-Stream (OF)	77.00%	99.71%	98.49%
Single-Stream (PE)	100.0%	24.00%	28.00%
Single-Stream (VR)	26.00%	100.0%	95.00%

Table 4. Results for the URFD dataset with the average and threshold.

	URFD avg		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	42.00%	100.0%	96.82%
Multi-Stream (OF+PE)	84.00%	99.0%	98.40%
Multi-Stream (OF+VR)	20.00%	100.00%	95.59%
Multi-Stream (PE+VR)	30.00%	100.0%	96.10%
Single-Stream (OF)	61.00%	100.0%	97.86%
Single-Stream (PE)	98.00%	26.00%	29.75%
Single-Stream (VR)	13.00%	100.0%	95.11%

The Multicam dataset was the largest dataset used during the experiments, its results can be seen in Tables 7, 8 and 9. As expected based on the same pattern of the results yielded by the previous tested datasets, Table 9 both sensitivity and specificity values had improvements compared to the other tables when the ensemble was assisted by the SVM. These results were obtained also using a five fold cross-validation, a mini-batch of 1024, learning rated of 10^{-3}, normalized batches and 1000 epochs. It is possible to observe that Table 9 had the best results, however when dealing with a dataset with a larger amount of data Tables 7 and 8 had similar values.

Table 5. Results for the URFD dataset with the average and SVM.

	URFD SVM/avg		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	66.00%	100.0%	98.13%
Multi-Stream (OF+PE)	71.00%	100.0%	98.40%
Multi-Stream (OF+VR)	61.00%	100.00%	97.86%
Multi-Stream (PE+VR)	23.00%	100.0%	95.73%
Single-Stream (OF)	63.00%	100.0%	97.95%
Single-Stream (PE)	98.00%	26.00%	30.37%
Single-Stream (VR)	19.00%	100.0%	95.55%

Table 6. Results for the URFD dataset with an SVM.

	URFD SVM		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	77.00%	100.0%	98.75%
Multi-Stream (OF+PE)	81.00%	100.0%	98.84%
Multi-Stream (OF+VR)	61.00%	100.00%	97.86%
Multi-Stream (PE+VR)	19.00%	100.0%	95.55%
Single-Stream (OF)	76.00%	100.0%	98.57%
Single-Stream (PE)	98.00%	24.00%	28.33%
Single-Stream (VR)	24.00%	100.0%	95.82%

As it can be seen the accuracy values did not necessarily increase if the number of streams increased. However, we believe that some high-level features can be appropriate to detect specific actions, for example the pose estimation descriptor has a better detection rate when dealing with falling situations while the visual rhythm with negative cases for some of the datasets. Both of these features assisted in the growth of specificity and sensitivity values for the optical flow descriptor.

Table 7. Results for the Multicam dataset with the average and threshold.

	Multicam avg		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	84.00%	92.00%	88.33%
Multi-Stream (OF+PE)	62.00%	76.00%	70.45%
Multi-Stream (OF+VR)	89.00%	95.00%	92.52%
Multi-Stream (PE+VR)	92.00%	82.00%	86.56%
Single-Stream (OF)	58.00%	90.00%	76.35%
Single-Stream (PE)	27.00%	62.00%	47.17%
Single-Stream (VR)	91.00%	92.00%	91.71%

Table 8. Results for the Multicam dataset with the average and SVM.

	Multicam SVM/avg		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	77.00%	93.00%	86.21%
Multi-Stream (OF+PE)	58.00%	86.00%	73.65%
Multi-Stream (OF+VR)	74.00%	98.00%	87.76%
Multi-Stream (PE+VR)	88.00%	90.00%	89.50%
Single-Stream (OF)	55.00%	91.00%	75.79%
Single-Stream (PE)	25.00%	68.00%	49.58%
Single-Stream (VR)	91.00%	92.00%	91.69%

In a general applicability, our method outputs as interesting results as some of the methods found in the literature which used the same datasets (Tables 10, 11 and 12). However, we are not able to directly compare the methods of the literature to our approach, since we do not know how the dataset was manipulated. Nonetheless, our method had better specificity and, in some cases, accuracy results when compared to the others.

In general, the small amount of data provided by these datasets, considering what we use for training, made it difficult to maintain an above 95% accuracy for all of the cases. It is possible to observe that the dataset that had a larger amount of information, Multicam, yielded more stable results compared to FDD and URFD. Compared to Carneiro et al. [1], our method even though not using the RGB information was still able to output compatible results.

Table 9. Results for the Multicam dataset with an SVM.

	Multicam SVM		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	90.00%	94.00%	92.33%
Multi-Stream (OF+PE)	58.00%	90.00%	76.48%
Multi-Stream (OF+VR)	91.00%	93.00%	92.21%
Multi-Stream (PE+VR)	91.00%	92.00%	91.64%
Single-Stream (OF)	55.00%	91.00%	75.79%
Single-Stream (PE)	25.00%	68.00%	49.58%
Single-Stream (VR)	91.00%	92.00%	91.69%

Table 10. FDD comparison with other methods.

	FDD comparison		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	82.00%	99.81%	98.84%
Nunez-Marcos et al.[13]	99.00%	97.00%	97.00%
Zerrouki and Houacine [11]	-	-	97.02%
Carneiro et al. [1]	99.90%	98.32%	98.43%

Table 11. URFD comparison with other methods.

	URFD comparison		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	77.00%	100.0%	98.75%
Nunez-Marcos et al.[13]	100.0%	92.00%	95.00%
Zerrouki and Houacine [11]	-	-	96.88%
Kwolek and Kepski [12]	100.0%	92.50%	95.71%
Carneiro et al. [1]	100.0%	98.61%	98.77%

Table 12. Multicam comparison with other methods.

	Multicam comparison		
Methods	**Sensitivity**	**Specificity**	**Accuracy**
Multi-Stream (OF+PE+VR)	90.00%	94.00%	92.33%
Nunez-Marcos et al.[13]	99.00%	96.00%	-

5.2. Qualitative Results

As expected, the method had some troubles when dealing with situations that had similar movement as falling actions. The videos that were not classified correctly were related to abruptly sitting actions, squatting or even lying down (Figure 8).

Fig. 8. Misleading fall detection squatting frame example.

6. CONCLUSIONS AND FUTURE WORK

In this work, we proposed and evaluated a multi-stream learning model based on convolutional neural networks to cope with a falling classification problem. Therefore, our approach consisted in extracting hand-crafted high-level features (optical flow, visual rhythm, and pose estimation) from public data set videos and using each one as an input to a distinct VGG-16 classifier. In addition to the feature combination, we also studied the best of three ensemble techniques to cope with our binary classification problem.

We believe that a multi-stream model can assist in classification since an outlier result from one of the streams can be corrected based on the other. Using high-level features also assisted in covering unnecessary information from the video frames such as the background and other unimportant details. It was also possible to observe that the SVM assisted in the balance and increase of the sensitivity and specificity metrics when used in the ensemble. In addition, compared to previous work we were able to maintain accuracy even though not using the RGB frame information itself.

For future work, we intend to continue investigating relevant high-level features used for fall detection. In addition, the studies will be conducted to test better parameters for the individual learners and use other architectures for each stream. Finally, we will also investigate the use of contexts other than falling to observe if this learning method is able to maintain its accuracy and general applicability.

7. ACKNOWLEDGMENTS

The current archival periodical article is based on the conference presentation [1]. The authors thank CAPES, FAPESP (grants #2014/12236-1 and #2017/12646-3), CNPq (grants #305169/2015-7, #421521/2016-3 and #307062/2016-3) for the financial support, as well as Semantix Brasil for the infrastructure and support provided during the development of the present work.

8. REFERENCES

[1] S. Carneiro, G. Silva, G. Leite, R. Moreno, S. Guimaraes, H. Pedrini, "Multi-Stream Deep Convolutional Network Using High-Level Features Applied to Fall Detection in Video Sequences", Proceedings of the 26th International Conference on Systems, Signals and Image Processing, Osijek, Croatia, 5-7 June 2019, pp. 293–298.

[2] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556, 2014.

[3] C.-W. Lin, Z.-H. Ling, Y.-C. Chang, C. J. Kuo, "Compressed-Domain Fall Incident Detection for Intelligent Home Surveillance", Proceedings of the International Symposium on Circuits and Systems, Kobe, Japan, 23-26 May 2005, pp. 3781–3784.

[4] D. T. Anderson, J. M. Keller, M. Skubic, X. Chen, Z. He, "Recognizing Falls from Silhouettes", Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, USA, 30 August - 3 September 2006, pp. 6388–6391.

[5] H. Qian, J. Zhou, Y. Mao, Y. Yuan, "Recognizing Human Actions from Silhouettes Described with Weighted Distance Metric and Kinematics", Multimedia Tools and Applications, Vol. 76, No. 21, 2017, pp. 21889–21910.

[6] M. Yu, Syed . Naqvi, J. Chambers, "Fall Detection in the Elderly by Head Tracking", Proceedings of the 5th Workshop on Statistical Signal Processing, Cardiff, UK, 31 August - 3 September 2009, pp. 357–360.

[7] M. Yu, S. M. Naqvi, J. Chambers, "A Robust Fall Detection System for the Elderly in a Smart Room", Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, Dallas, TX, USA, 14-19 March 2010, pp. 1666–1669.

[8] C. Rougier, J. Meunier, A. St-Arnaud, J. Rousseau, "Robust Video Surveillance for Fall Detection based on Human Shape Deformation", Proceedings of the IEEE Transactions on Circuits and Systems for Video Technology , Vol. 21, No. 5, 2011, pp. 611–622.

[9] B.-S. Lin, J.-S. Su, H. Chen, C. Y. Jan, "A Fall Detection System based on Human Body Silhouette", Proceedings of the 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, 16-18 October 2013, pp. 49–52.

[10] D. P. Kumar, Y. Yun, I. Yu-Hua Gu, "Fall Detection in RGB-D Videos by Combining Shape and Motion Features", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 20-25 March 2016, pp. 1337–1341.

[11] N. Zerrouki, F. Harrou, A. Houacine, Y. Sun, "Fall Detection using Supervised Machine Learning Algorithms: A Comparative Study", Proceedings of the 8th International Conference on Modelling, Identification and Control, Algiers, Algeria, 15-17 November 2016, pp. 665–670.

[12] B. Kwolek, M. Kepski, "Improving Fall Detection by the Use of Depth Sensor and Accelerometer", Neurocomputing, Vol. 168, 2015, pp. 637– 645.

[13] A. Nunez-Marcos, G. Azkune, I. Arganda-Carreras, "Vision-based Fall Detection with Convolutional Neural Networks", Wireless Communications and Mobile Computing, Vol. 2017, 2017.

[14] E. Akağunduz, M. Aslan, A. Senğur, H. Wang, M. C. Ince, "Silhouette Orientation Volumes for Efficient Fall Detection in Depth Videos", IEEE Journal of Biomedical and Health Informatics, Vol. 21, No. 3, 2017, pp. 756–763.

[15] L. Anishchenko, "Machine Learning in Video Surveillance for Fall Detection", Proceedings of the Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology, Yekaterinburg, Russia, 7-8 May 2018, pp. 99–102.

[16] N. Zerrouki, A. Houacine, "Combined Curvelets and Hidden Markov Models for Human Fall Detection", Multimedia Tools and Applications, Vol. 77, No. 5, 2018, pp. 6405–6424.

[17] A. Lowhur, M. C. Chuah, "Dense Optical Flow based Emotion Recognition Classifier", Proceedings of the 12th International Conference on Mobile Ad Hoc and Sensor Systems, Dallas, TX, USA, 19-22 October 2015, pp. 573–578.

[18] B. S. Torres, H. Pedrini, "Detection of Complex Video Events through Visual Rhythm", The Visual Computer, Vol. 34, No. 2, 2018, pp. 145–165.

[19] T. Moreira, D. Menotti, H. Pedrini, "First-Person Action Recognition Through Visual Rhythm Texture Description", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 5-9 March 2017, pp. 1–4.

[20] F. B. Valio, H. Pedrini, N. J. Leite, "Fast Rotation-Invariant Video Caption Detection based on Visual Rhythm", Proceedings of the Iberoamerican Congress on Pattern Recognition, Pucón, Chile, 15-18 November 2011, pp. 157–164.

[21] S. J. F. Guimaraes, M. Couprie, A. A. Araújo, and N. J. Leite, "Video Segmentation based on 2D Image Analysis", Pattern Recognition Letters, Vol. 24, No. 7, 2003, pp. 947–957.

[22] S. J. F. Guimaraes, A. A. Araújo, M. Couprie, N. J. Leite, "Video Fade Detection by Discrete Line Identification," Proceedings of the Object Recognition Supported by User Interaction for Service Robots, Quebec City, Quebec, Canada, Canada, 11-15 August 2002, pp. 1013–1016.

[23] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh,"Real time Multi-Person 2D Pose Estimation using Part Affinity Fields", arXiv preprint arXiv:1611.08050, 2016.

[24] T. G. Dietterich, "Ensemble Methods in Machine Learning", Proceedings of the International Workshop on Multiple Classifier Systems, Springer, Cagliari, Italy, 21-23 June 2000, pp. 1–15.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database", Proceedings of the Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20-25 June 2009, pp. 248–255.

[26] K. Soomro, A. R. Zamir, M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild", arXiv preprint arXiv:1212.0402, 2012.

[27] B. Kwolek, M. Kepski, "Human Fall Detection on Embedded Platform using Depth Maps and Wireless Accelerometer", Computer Methods and Programs in Biomedicine, Vol. 117, No. 3, 2014, pp. 489–501.

[28] H. A. Maia, D. T. Concha, H. Pedrini, H. Tacon, A. S. Brito, H. L. Chaves, M. B. Vieira, S. M. Villela, "Action Recognition in Videos Using Multi-Stream Convolutional Neural Networks", Deep Learning Applications, Springer International Publishing, 2020, pp. 95–111.

[29] R. Quispe, D. Ttito, A. Rivera, H. Pedrini, "Multi-Stream Networks and Ground-Truth Generation for Crowd Counting", International Journal of Electrical and Computer Engineering Systems, Vol. 11, No. 1, 2020, pp. 25–33.

[30] H. Tacon, A. S. Brito, H. L. Chaves, M. B. Vieira, S. M. Villela, H. A. Maia, D. T. Concha, H. Pedrini, "Multi-Stream Architecture with Symmetric Extended Visual Rhythms for Deep Learning Human Action Recognition", Proceedings of the 15th International Conference on Computer Vision Theory and Applications, Valletta, Malta, February 27-29, 2020, pp. 351–358.