Magnitude-Orientation Stream Network and Depth Information applied to Activity Recognition

Carlos Caetano[a,*], Victor H. C. de Melo[a], François Brémond[b], Jefersson A. dos Santos[a], William Robson Schwartz[a]

[a]*Smart Surveillance Interest Group, Department of Computer Science Universidade Federal de Minas Gerais, Belo Horizonte, Brazil*
[b]*I.N.R.I.A., 2004 rte des Lucioles -BP 93, 06902 Sophia-Antipolis Cedex, France*

Abstract

The temporal component of videos provides an important clue for activity recognition, as a number of activities can be reliably recognized based on the motion information. In view of that, this work proposes a novel temporal stream for two-stream convolutional networks based on images computed from the optical flow magnitude and orientation, named Magnitude-Orientation Stream (MOS), to learn the motion in a better and richer manner. Our method applies simple non-linear transformations on the vertical and horizontal components of the optical flow to generate input images for the temporal stream. Moreover, we also employ depth information to use as a weighting scheme on the magnitude information to compensate the distance of the subjects performing the activity to the camera. Experimental results, carried on two well-known datasets (UCF101 and NTU), demonstrate that using our proposed temporal stream as input to existing neural network architectures can improve their performance for activity recognition. Results demonstrate that our temporal stream provides complementary information able to improve the classical two-stream methods, indicating the suitability of our approach to be used as a temporal video representation.

Keywords: Activity recognition, convolutional neural networks (CNNs),

*Corresponding author
Email address: carlos.caetano@dcc.ufmg.br (Carlos Caetano)

two-stream convolutional networks, spatiotemporal information, optical flow, depth information.

---

1. Introduction

Human activity recognition has been used in many real-world applications. In environments that require a higher level of security, surveillance systems can be used to detect and prevent abnormal or suspicious activities such as robberies and kidnappings. In addition, human activity recognition can be employed in systems for video retrieval, so that a user is able to search for videos containing specific activities. Another type of application is in health care, such as activities of daily living monitoring systems.

Surveillance applications have traditionally relied on network cameras monitored by human operators that must be aware of the activities carried out by people who are in the camera field of view. With the recent growth in the number of cameras to be analyzed, the efficiency and accuracy of human operators has reached the limit [1]. Therefore, security agencies have attempted computer vision-based solutions to replace or assist the human operator. Automatic recognition of suspicious activities is a problem that has attracted the attention of researchers in the area [2, 3, 4, 5].

A significant portion of the progress on the activity recognition task has been achieved with the design of discriminative feature descriptors exploring temporal information. Such information is based on motion analysis and is very important to represent the video in a more discriminative space, allowing the improvement of activity recognition.

Over the last decade, most of the works focused on designing handcrafted local feature descriptors [6, 7, 8, 9] or on encoding schemes using mid-level representations, such as Bag-of-Words (BoW) [10] or Fisher vector (FV) [11], followed by Support Vector Machines (SVM) classifier. Nowadays, large efforts have been directed to the employment of deep Convolutional Neural Networks (CNNs). These architectures learn hierarchical layers of representations

to perform pattern recognition and have demonstrated impressive results on many pattern recognition tasks (e.g., image classification [12] and face recognition [13]). Although the excellent improvements achieved in such tasks, activity recognition lacks on performance when using CNNs. Many works [14, 15, 16] point that the potential reason behind such gap falls in two cases: (i) current datasets do not have enough videos for training and are too much noisy; and (ii) current CNN architectures are still not able to handle temporal information (or to take full advantage of it), consequently letting spatial (appearance) information prevail.

A major breakthrough spatiotemporal information representation was achieved by Simonyan and Zisserman [17], who directly incorporated motion information by using optical flow instead of learning it from scratch, showing significant improvement over other approaches. Known as two-stream network, their architecture is composed of two stream of data: (i) spatial network, which takes as input the raw RGB pixels; and (ii) temporal network, which takes as input dense optical flow displacement fields (vertical and horizontal components) computed across the frames. The final predictions are computed as the average of the output scores from the two streams. On a recent study, Varol et al., [18] investigated the impact of different motion estimators to be used as input for temporal stream networks. Their experiments confirmed the advantage of motion-based representations and emphasized the importance of a accurate motion estimation for learning efficient representations for human activity recognition. Other works from the literature also mention that although the temporal evolution patterns can be learned implicitly with CNNs, an explicit modeling is preferable and might provide better results [19].

To further improve the representation of spatiotemporal information, a new temporal stream for the two-stream networks to perform activity recognition, named *Magnitude-Orientation Stream* (MOS), is developed in this work. The method is based on non-linear transformations on the optical flow components to generate input images for the temporal stream. Our hypothesis is based on the assumption that the motion information on a video sequence can be described

by the spatial relationship contained on the local neighborhood of magnitude and orientation extracted from the optical flow. More specifically, we assume that the motion information is adequately specified by fields of magnitude and orientation. In view of that, our method captures not only the displacement, by using orientation, but also magnitude providing information regarding the velocity of the movement. Moreover, we also employ depth information estimated from the RGB video data to use as weighting scheme on the magnitude information to compensate the distance of the subjects performing the activity to the camera which we call MOS+D.

The use of *Magnitude-Orientation Stream* (MOS) for activity recognition was first introduced in our previous work [20]. This paper incorporates several new aspects in comparison with our previous work. Those aspects are highlighted in the following:

- New formulation which weight the magnitude by the depth information in order to circumvent problems related to activities taken regardless of their distance in relation to the camera.
- A detailed revision of the literature with the intention of including the recently published works which were not analyzed in our previous work.
- A study regarding the behavior of *Magnitude-Orientation Stream* (MOS) by providing a more detailed comparison of when our method prevail and also where it fails comparing to literature approaches.

According to the experimental results, our proposed temporal stream used as input to existing neural network architectures is able to recognize activities accurately on two well-know datasets (UCF101 [21] and NTU [22]) outperforming the results achieved by the original two-stream network as well as other deep networks available in the literature. Moreover, we show that optical flow pre-processing (i.e., extraction of magnitude and orientation information) is beneficial bringing improvements over using raw optical flow information and

helps on guiding the network to extract certain motion information, possibly complementary, that by using a single modality (RGB) it could not extract.

The remainder of this paper is organized as follows. In Section 2, we give a explanation of the works in literature that explore temporal information to perform activity recognition. In Section 3, we introduce our approach to extract temporal information based on magnitude and orientation and also the new weighting scheme used to ponder magnitude by the depth information. Then, Section 4 presents our experimental results, validating the performance achieved. Finally, Section 5 presents conclusions obtained.

2. Related Works

In this section, we present a literature review of works that are close to the idea proposed in our approach. These methods can be categorized by: (i) temporal information extracted from videos through the use of handcrafted local feature descriptors (Section 2.1), and (ii) recent works that employ neural networks to learn temporal information (Section 2.2).

2.1. Methods based on Handcrafted Feature Descriptors

To characterize motion and appearance of local features, Laptev et al. [7] computed histogram descriptors of space-time volumes in the neighborhood of detected points. Each volume is subdivided into a grid of cuboids and, for each cuboid, they compute Histogram of Oriented Gradients (HOG) [23] and Histogram of Optical Flow (HOF). The HOG descriptor is computed by dividing the cuboid into regions and accumulating a histogram binned by gradient directions over the pixels, while HOF is binned according to the flow orientations and weighted according to its magnitude. Then, normalized histograms are concatenated and named HOG-HOF.

Dalal et al. [24] introduced the Motion Boundary Histogram (MBH). First applied to human detection, the motion boundary coding scheme captures the local orientations of motion edges based on HOG feature descriptors [23]. Treating the horizontal and vertical components of the optical flow as independent

"images", the authors take their local gradients separately, find the corresponding magnitudes and orientations and use these as weighted votes to the local orientation histograms. Later on, the MBH was used on several works to describe motion information for activity recognition [7, 9, 25].

The HOG feature descriptor was extended by Kläser et al. [8], named as HOG3D. It is based on histograms of 3D gradient orientations computed using an integral video representation. The gradients are binned into regular polyhedrons in a multi-scale fashion in space and time. Therefore, HOG3D combines appearance and motion information.

Aiming at encoding both local static appearance and motion information, as in the HOG3D, but avoiding high dimensionality and a relatively expensive quantization cost, Shi et al. [26] proposed the Gradient Boundary Histograms (GBH). Instead of using image gradients, the authors use time-derivatives of image gradients to emphasize moving edge boundaries. For each frame, they compute image gradients and apply temporal filtering over two consecutive gradient images. Then, they compute the magnitude and orientation for each pixel which are used to build a histogram of orientation as in HOG.

Colque et al. [27] developed a feature called Histograms of Optical Flow Orientation and Magnitude (HOFM). Different from HOF that only encodes orientation information, HOFM captures the orientation and the magnitude of flow vectors providing information regarding the velocity of the moving objects. They build a 3D matrix based on the orientation and magnitude information provided by the optical flow field, where each line corresponds to a given orientation range and each column to the magnitude ranges. The authors then extended it to capture information regarding appearance and density of regions by encoding the entropy of the orientation flow [28].

Aiming at capturing richer information from the optical flow, Caetano et al. [29] proposed the Optical Flow Co-occurrence Matrices (OFCM). The descriptor is based on the extraction of a set of statistical measures from co-occurrence matrices computed using the magnitude and orientation from optical flow information. Their hypothesis for designing the OFCM is based on the as-

sumption that the motion information on a video sequence can be described by the spatial relationship contained on local neighborhoods of the flow field.

A major breakthrough on local feature-based approaches was achieved by Wang et al. [9] which proposed an method to describe videos by dense trajectories. Trajectory shapes encode local motion information by tracking spatial interest points over time. To generate the trajectories, they sample interest points in space and time, and track them based on displacement information using an efficient dense optical flow algorithm. The HOG, HOF and MBH feature descriptors are used to describe the trajectories which are then encoded by Bag-of-Words (BoW) mid-level representation. Afterwards, the authors improved it to the Improved Dense Trajectories (IDT) [25] using the homography between consecutive frames to estimate the camera motion and Fisher vector encoding.

Although there are many approaches based on local feature descriptors, these works often require over engineering (e.g., feature extraction, mid-level representation and classifier training). On contrary, CNNs are a class of deep learning models that replace all engineering with a single neural network trained end to end from pixel values to classifier outputs [30].

2.2. Methods based on Neural Network Approaches

Convolutional Neural Networks (CNNs) have achieved impressive state-of-the-art results on image classification [12]. Therefore, many works have tried to apply CNNs to learn spatiotemporal information for activity recognition task. A natural choice, the 3D convolutional network was presented by Ji et al. [31], where they tried to learn both appearance and motion features with 3D convolution operations. Their method works by stacking consecutive segments of human subjects in videos and by applying 3D convolutions over such volume aiming that the first layer learns spatiotemporal features. Tran et al. [32] also explored 3D CNNs. However, in contrast with Ji et al. [31], their method takes full video frames as inputs and does not rely on any preprocessing.

Karpathy et al. [30] also used CNN aiming to learn motion features. The

authors investigated different temporal information fusion schemes, learning local motion direction/speed with global information. Although significant gains in accuracy compared to the works based on handcrafted features, only little improvement was achieved when compared to single-frame CNN models, showing that the current CNN architectures are unable to efficiently learn motion features.

A major breakthrough was achieved by Simonyan and Zisserman [17]. Instead of trying to learn motion information as Karpathy et al. [30] and Tran et al. [32], the authors incorporated it by using optical flow. Known as two-stream network, their architecture is composed of two stream of data: (i) spatial network, which takes as input the raw RGB pixels; and (ii) temporal network, which takes as input dense optical flow displacements computed across the frames. Final predictions are computed as the average of the output scores from the two streams, showing significant improvement over other approaches. Our method differs from them by capturing not only the displacement but also velocity information provided by optical flow magnitude.

By employing the aforementioned two-stream network, Wang et al. [33] conducted experiments showing the impact on results when changing the network architecture. In addition, they also introduced some data augmentation techniques to improve the network training. To that end, the authors used three distinct architectures (ClarifaiNet [34], GoogLeNet [35] and VGG-16 [36]) showing that the best results are achieved by VGG-16 deeper architecture. Afterwards, the authors improved it to the Temporal Segment Networks (TSN) [37] by studying different types of input modalities to two-stream and by employing the Inception with batch normalization network architecture [38].

Perez et al. [39] used MPEG motion vectors [40] as a different input for a two-stream network to explore temporal information. Such vectors are used to perform motion estimation in video compression where pixels are grouped in macroblocks and motion vectors are then computed for each block. They show that both optical flow and MPEG motion vectors provide equivalent accuracies, but the latter allows a more efficient implementation. Later, Varol et al., [18]

also studied the impact of different motion information to be used as input for the networks. They investigated the dependency of activity recognition on the quality of motion estimation by experimenting three types of optical flow inputs: (i) MPEG motion vectors; (ii) Farneback optical flow estimator; and (iii) Brox optical flow. Their experiments confirmed the advantage of motion-based representations and highlight the importance of good quality motion estimation for learning efficient representations for human activity recognition. As such magnitude and orientation can be estimated from any motion field, we believe that such different inputs might be boosted with our approach.

As another type of input information, Zhu and Newsam [41] performed an investigation with depth information for large-scale human activity recognition in video without using any depth sensor, such as Kinect-like devices. To that end, the authors estimated the depth information directly from the video itself by using two state-of-the-art approaches to extract depth from images and experimented it by feeding two different CNN models. Although we also use estimated depth information in our approach, it is important to emphasize that differently from Zhu and Newsam [41], we do not employ depth information as input for the network instead, we use it to weight the magnitude values to circumvent problems related to activities executed regardless of their distance to the camera.

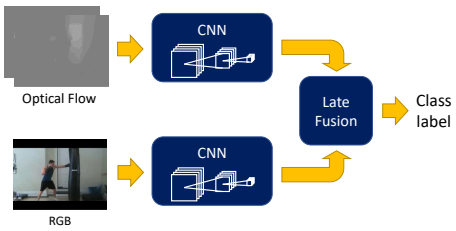
To make a spatial network learn to relate which parts of the image are moving, Park et al. [15] proposed a feature amplification technique by using magnitude information of the optical flow on the spatial network. To that end, they extract feature maps of the last convolutional layer of the spatial network, compute optical flow magnitudes and resize it to be the same size of the previously extracted feature maps. Finally, they perform element-wise product to amplify the activations. Our work differs from them in that we use the magnitude information right on the beginning of the network, letting it learn how the velocity information contributes on the activity recognition process.

To capture temporal dynamics of body parts over time, Zolfaghari et al. [42] proposed a combination of networks based on a three-stream architecture. Their

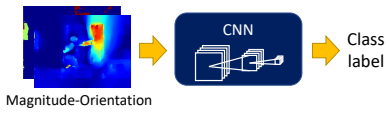
method relies on three different inputs: the raw RGB images, optical flow information and human pose. For the later stream, the authors used a network for human body part segmentation which provides body pose information. According to the authors, highlight to the pose network is because it yields the spatial localization of the person, allowing to apply the approach to spatial activity localization in a straightforward manner. To perform a direct comparison to the method proposed by Zolfaghari et al. [42], we also employed a three-stream architecture, however instead of using body pose information as a third stream, here we employ the magnitude and orientation information.

Revisiting 3D convolutions, Carreira and Zisserman [43] argue that the reason 3D convolutions might be unable to improve over their flat counterparts lies on the dataset. They take state-of-the-art activity recognition architectures, inflate them to 3D convolutions, and evaluate them on the novel Kinetics dataset [44]. They show that 3D convolution yields better results, showing that the previous tries of modeling temporal information with 3D convolution failed due to noisy datasets and/or lack of data.In further experiments, Carreira and Zisserman evaluate both RGB and optical flow to the 3D convolution and show that optical flow still has a leverage. Hence, we believe that our orientation-magnitude representation might as well provide improvements using 3D convolution on Kinetics.

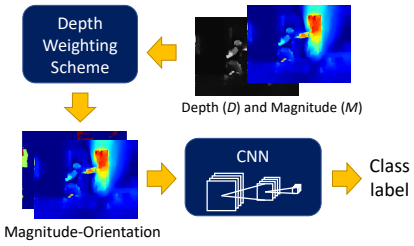
As it can be inferred from the reviewed methods, most of them use either convolution operations over raw pixels or optical flow to model temporal information. The former do not decouple spatial and temporal information, letting appearance information prevail [14], while the latter approaches rely on horizontal and vertical components of the optical flow. Despite the optical flow-based methods produce promising results, they focus only on displacement information. In view of that, aiming at capturing more information from the optical flow, our method encodes not only the displacement, by using orientation, but also captures the magnitude providing information regarding the velocity of the movement. Moreover, we also employ the use of depth information to compensate the distance of the subjects to the camera.



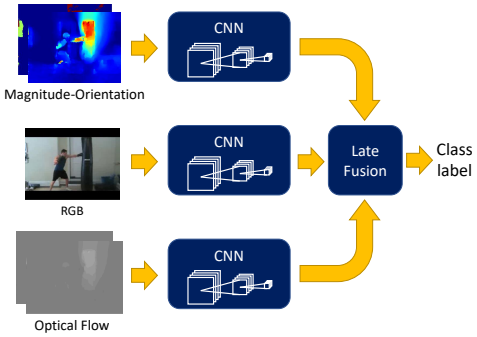
(a)Two-Stream approach.



(b) Our Magnitude-Orientation Stream (MOS) approach.



(c) Our MOS weighted by depth (MOS+D) approach.



(d) Fusion of our approach and Two-Stream approach.

Figure 1: Architectures considered in this work for extracting spatiotemporal information.

3. Proposed Approach

In this section, we present our approach for performing activity recognition with our proposed *Magnitude-Orientation Stream* (MOS). For completeness, we first present the basic concepts of the network architectures we use to learn the data representation. Then, we detail our method showing how to incorporate magnitude and orientation as temporal information for the network input. Finally, we explain the approach used to estimate the depth information from monocular videos [45] and how we employ it as a magnitude weighting scheme to compensate the distance of the subjects to the camera, which we call MOS+D.

3.1. Employed Architectures

In this section, we present the basic concepts of the Very Deep Two-Stream (VD2S) [33] and Temporal Segment Networks (TSN) [37], which are the baseline network architectures we use to learn the data representation based on the magnitude and orientation.

3.1.1. Very Deep Two-Stream

Motivated by the successful results achieved by deep architectures (e.g., VGG-16) in object recognition task, Wang et al. [33] improved the two-stream network by adapting it to use the VGG-16 on activity recognition, which they called Very Deep Two-Stream (VD2S). As mentioned on Section 2, the two-stream network is composed by two different networks receiving distinct flows of data (spatial and temporal). The spatial stream receives as input the RGB frames while the temporal stream receives an optical flow image as input.

The spatial network is built on a single frame image and, therefore, its architecture is the same as those for object recognition on the image domain. Thus, at each iteration of the training step, 256 training videos are uniformly sampled across the classes and a single frame is randomly selected. Moreover, to avoid overfitting, the authors employ two data augmentation techniques: (i) cropping and flipping four corners and the center of the frame; and (ii) a multi-scale cropping method than randomly sampling the cropping width and height from 256, 224, 192, 168. Finally, they resize the cropped regions to $224 \times 224 \times 3$.

The temporal network receives images of optical flow as input. The process for computing the optical flow is explained as follows. For each frame F on time t, optical flow O_t is computed considering F_t and F_{t+1}. The resulting optical flow O_t is composed by two channels: (i) $\mathcal{O}_t^x$, denoting an image containing the x (horizontal) displacement field; and (ii) $\mathcal{O}_t^y$, denoting an image containing the y (vertical) displacement field. Moreover, to avoid storing the displacement fields as floats, the horizontal and vertical components of the flow are linearly

rescaled to a [0, 255] interval as

$$\mathcal{I}^f_{t_{i,j}} = \begin{cases} 0, & \text{if } \mathcal{O}^f_{t_{i,j}} < l \\ 255, & \text{if } \mathcal{O}^f_{t_{i,j}} > h \\ 255 \times \frac{(O^f_{t_{i,j}} - l)}{(h-l)}, & \text{otherwise} \end{cases}, \tag{1}$$

where f represents the image channel (flow component x or y), h is the higher bound maximum optical flow value, l is the lower bound minimum optical flow value and $\mathcal{I}^f$ the optical flow image. The same data augmentation techniques used in spatial network are used in the temporal stream. Finally, the input of the temporal network is composed by stacking 10 randomly images $\mathcal{I}^f$ of optical flow fields ($224 \times 224 \times 20$) [17].

To perform the combination of the two networks, a late fusion scheme is employed by using a weighted linear combination of their prediction scores, where the weight is set as 2 for temporal network and 1 for spatial network, giving, therefore, more importance to the temporal information. Figure 1(a) illustrates the Deep Two-Stream network.

3.1.2. Temporal Segment Networks

Most CNN frameworks usually focus their learning methods on short-term motions by working on a single stack of frames, thus lacking the capacity to incorporate long-range temporal structure. To learn a video representation that is able to capture such structure, Wang et al. [37] developed the Temporal Segment Networks (TSN) which extracts short snippets over the video by employing a sparse sampling scheme to capture the long-range temporal structure.

The basic idea of the work proposed by Wang et al. [37] is the following. Given a video, it is divided into K segments and for each segment, their approach randomly samples T snippets which are used as inputs for a two-stream network. After the predictions of each snippet, the authors employed a fusion scheme by an aggregation function (averaging, maximum or weighted averaging). Finally, Softmax is applied to predict the probability classes for the whole video.

Regarding the network architecture, Wang et al. [37] adapted the Inception with Batch Normalization (BN-Inception) to the design of two-stream following

the same input scheme as the Very Deep Two-Stream [33]: (i) spatial stream, which receives RGB images; and (ii) temporal stream, that operates on a stack of optical flow images. Moreover, they employed the same data augmentation techniques and late fusion scheme to perform the combination of the two networks as in Very Deep Two-Stream [33].

3.2. Magnitude-Orientation Stream

Our Magnitude-Orientation Stream (MOS) follows the same fundamentals as the Two-Stream networks. However, aiming at extracting more information from the optical flow, MOS captures the displacement information by using orientation of the optical flow and the velocity of the movement considering the optical flow magnitude. The spatial relationship contained on local neighborhoods of magnitude and orientation captures not only displacement, by using orientation, but also magnitude providing information regarding the velocity of the movement. The method is based on non-linear transformations on the optical flow components aiming to generate input images for the temporal stream. To incorporate such information on the temporal stream, we compute the dense optical flow as in [33]. In this way, for each video composed by n frames, we compute $n-1$ optical flows $\mathcal{O}$. Once the optical flow is available, we compute the magnitude and orientation information as

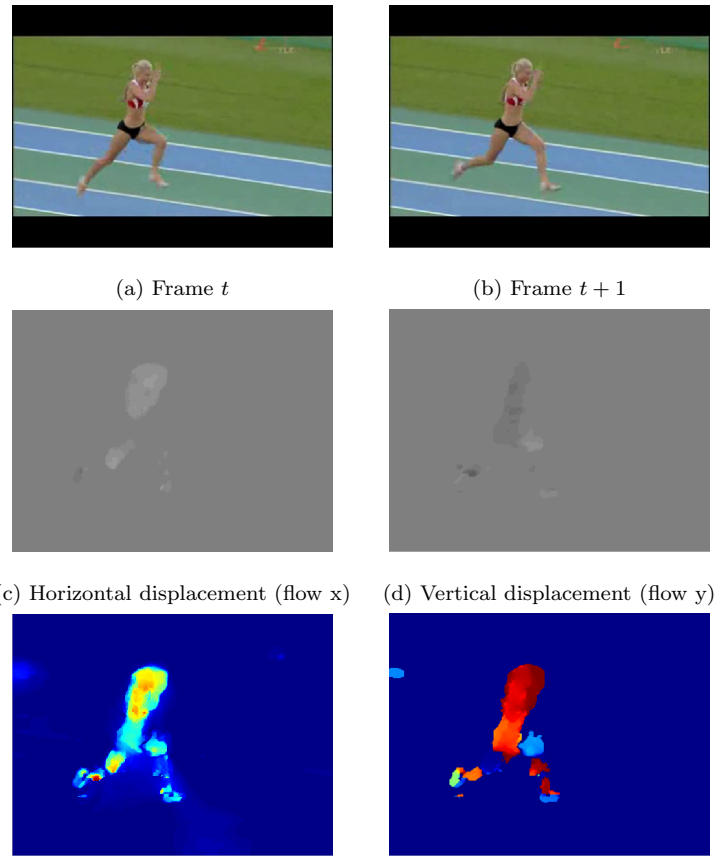
$$M_{i,j} = \sqrt{(\mathcal{O}^x_{i,j})^2 + (\mathcal{O}^y_{i,j})^2} \tag{2}$$

and

$$\theta_{i,j} = tan^{-1}\left(\frac{\mathcal{O}^y_{i,j}}{\mathcal{O}^x_{i,j}}\right), \tag{3}$$

where M and θ are the magnitude and orientation information, respectively.

Since the values obtained in M and θ are composed by real numbers, they are linearly rescaled to a [0, 255] using Equation 1. Moreover, since the orientation values are estimated for every pixel of the optical flow, it can generate noisy values from regions of the image without any movement. Therefore, we



(a) Frame t (b) Frame $t+1$

(c) Horizontal displacement (flow x) (d) Vertical displacement (flow y)

(e) Magnitude (M) (f) Orientation (θ')

Figure 2: Comparison between optical flow displacement information, magnitude and orientation extracted from two consecutive frames (t and $t+1$) of an activity sample extracted from the UCF101 dataset [21].

performed a filtering on θ based on the values of M as

$$\theta'_{i,j} = \begin{cases} 0, & \text{if } M_{i,j} < m \\ \theta_{i,j}, & \text{otherwise} \end{cases}, \tag{4}$$

where m is a magnitude threshold value. Figure 2 illustrates a comparison between the magnitude and orientation information with the optical flow x and y displacements extracted from two consecutive frames.

With the rescaled magnitude and orientation information, which can be seen as two image channels, we use the same data augmentation techniques as

in [33]. Therefore, the input is composed by 10 stacked images ($224 \times 224 \times 20$). Figure 1(b) illustrates the Magnitude-Orientation Stream network stages.

3.3. Depth Information Estimation

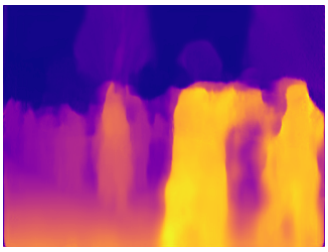
The use of depth information has shown several advantages in a number of visual recognition tasks including human activity recognition [46]. Compared with RGB video sequences, depth information have shown several advantages in the context of activity recognition, for instance Liang and Zheng [47] claim that depth data can provide 3D structural information so that the motion information of activities can be more discriminative.

Our main goal on using depth information is to circumvent problems related to activities taken regardless of their distance in relation to the camera. As an example, the "BandMarching" class in the UCF101 dataset [21]. Figure 3(c) shows the magnitude information extracted from a "BandMarching" video. As can be seen, although every person in the scene should have similar magnitude/velocity information, people closer to the camera present much higher magnitude values than people that are distant from the camera. Such difference happens because the pixel displacement near the camera is much higher than the displacement of distant pixels. To circumvent such problem, we apply a normalization scheme to the magnitude information by weighting the magnitude by the depth information.

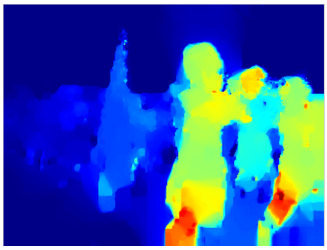
Since the videos from classic activity recognition datasets, such as UCF101 [21] and HMDB51 [48] were not recorded using a depth sensor to capture depth information, here we extract it from the RGB data employing a fast state-of-the-art approach [45] to estimate depth data from monocular views. In this way, for each video composed by n frames, we compute n depth maps. Once the depth maps are available, we first apply a Gaussian filter on each depth map with the aim of softening erroneous estimates and then we weight the magnitude information as

$$M^{'}_{i,j} = \begin{cases} M\{i,j\} \times (D_{i,j} + 1), & \text{if } D_{i,j} < d \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

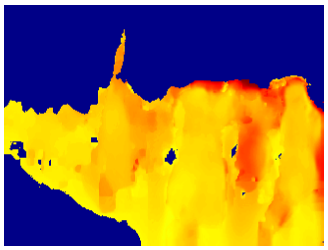
(a) Frame t



(b) Depth map (D)



(c) Magnitude (M)



(d) Magnitude weighted by depth ($M^{'}$)

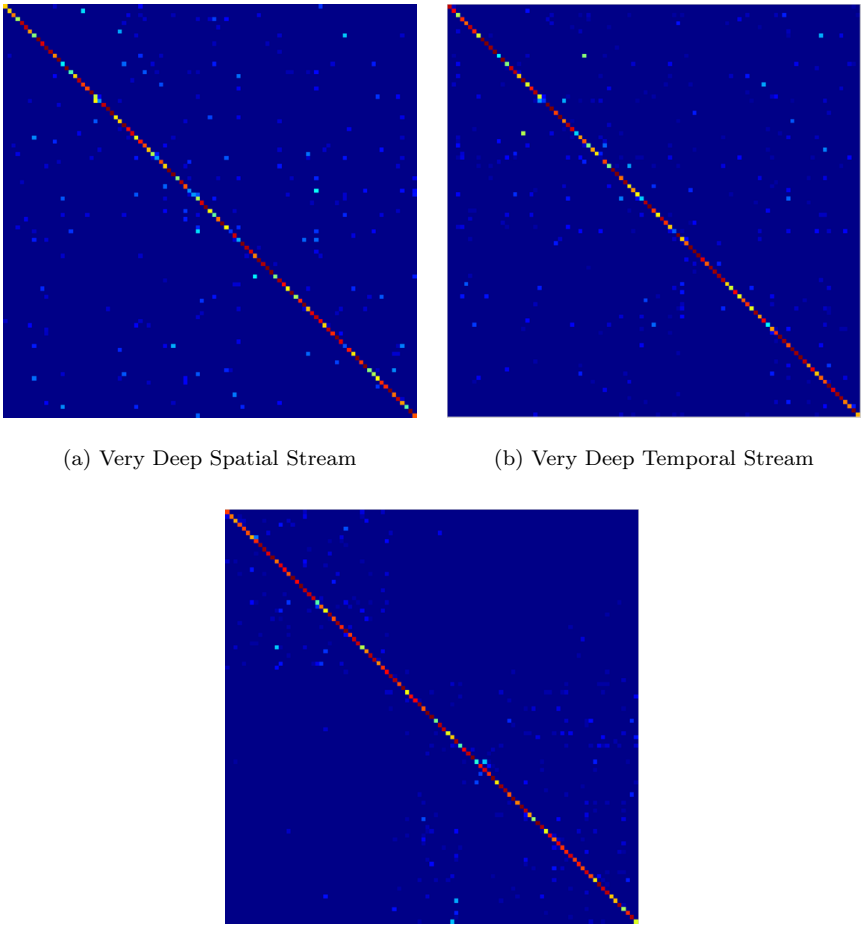
Figure 3: Comparison between magnitude information and the weighted magnitude by depth computed from an activity sample extracted from the UCF101 dataset [21].

where d is a depth threshold value we used and D is the depth map. The intuition for using such threshold lies on the premise that activities of interest being performed in the video usually do not take place in the background, therefore, we can filter noisy movements that are not of interest. Then, the weighted magnitude values are linearly rescaled to a [0, 255] using Equation 1.

Figure 2 illustrates the original magnitude information and the weighted magnitude information by depth. As can be seen, some magnitude information is lost due to erroneous estimations on the depth map (hat and head of the person in front). After that, the weighted magnitude information and orientation are used as input for a CNN. Figure 1(c) illustrates the Magnitude-Orientation Stream weighted by depth stages, which we call MOS+D.

Finally, to incorporate spatial information to our approach, we employ a late fusion technique with the Two-Stream network (by employing VD2S [33] or TSN [37]), as illustrated in Figure 1(d).

4. Experimental Results



(a) Very Deep Spatial Stream

(b) Very Deep Temporal Stream

(c) Magnitude-Orientation Stream (MOS)

Figure 4: Confusion matrices on UCF101 split 1. False positives and false negatives were highlighted to show where each method fails.

This section describes the experimental results obtained with the proposed method for the activity recognition problem and performs comparisons to our baseline, the Very Deep Two-Stream network (VD2S) [33] and Temporal Segment Networks (TSN) [37]. To isolate only the contribution brought by our method, the baselines were tested on the same datasets with the same split of training and testing sets. The evaluations are performed considering two well-

known datasets for the activity recognition problem, the UCF101 [21] and the NTU [22], in which we employ the evaluation protocols and metrics proposed by their authors.

4.1. Datasets

The *UCF101* [21] is an activity recognition dataset composed by videos collected from YouTube. It has a large diversity of activities and the presence of large variations in camera motion, object viewpoint, appearance, pose and scale, cluttered background, and illumination conditions. There are $13,320$ videos from 101 activity categories grouped into 25 groups. Each group can consist of 4-7 videos of an activity. The videos from the same group may share some common features, such as similar background or similar viewpoint. We follow the original protocol using three train-test splits. The performance is evaluated by computing the average recognition across all classes over the three splits as in [33].

The *NTU* [22] is a publicly available 3D activity recognition dataset. It consists of $56,880$ videos from 60 activity categories which are performed by 40 distinct subjects. The videos were collected by three Kinect cameras. The dataset provides four different data information: (i) RGB frames; (ii) depth maps; (iii) infrared sequences; and (iv) skeleton joints. We follow the original cross-subject evaluation protocol, which split the 40 subjects into training and testing. The performance is evaluated by computing the average recognition across all classes.

4.2. Implementation Details

4.2.1. Pre-training

As stated by [33], the UCF101 dataset training split is very small to train a deep convolutional network. In view of that, a possible solution used by several works [17, 33, 14, 37] is to use ImageNet models as the initialization for network training. In this way, here we also employed the ImageNet model as pre-training.

4.2.2. Training

Following the implementation details used by our baselines [33, 37], we set the learning rate initially to 0.005. For the Very Deep Two-Stream network (VD2S) [33], the learning rate decreases at every $5,000$ iterations dividing it by 10. The maximum iteration was set as $15,000$. We follow a similar scheme for the Temporal Segment Networks (TSN) [37] reducing the learning rate after 12.000 and 18.000 iterations. For TSN, the maximum iteration is set as 20.000.We kept the same schedule for all training sets.

Similarly to [17, 33, 37], the network weights are learned using the mini-batch stochastic gradient descent with a momentum set to 0.9 and weight decay of 0.0005. We also set high dropout ratio for the fully connected layers (0.9 and 0.8).

Krizhevsky et al. [12] demonstrated that data augmentation techniques can be very effective to avoid overfitting. In view of that, we cropped and flipped four corners and the center of the frame. In addition, we applied a multi-scale cropping method and randomly sampled the cropping width and height from $\{256, 224, 192, 168\}$ (finally, we resize the cropped regions to 224×224). It is important to state that our baseline [33] used the same data augmentation procedure.

4.2.3. Test

To perform a fair comparison, we applied the same test scheme used by our baseline [33], described as follows. First, we sample 25 magnitude/orientation flow images for the testing. Then, from each of these, we obtain 10 convolutional network inputs (by cropping and flipping four corners and the center). Finally, the prediction score for the input video is obtained by averaging the sampled images scores and their crops. The same testing scheme was used by the original two-stream convolutional network [17]. For the fusion of MOS and other streams, we use a non-weighted linear fusion which consists of a combination of their prediction scores.

4.2.4. Optical Flow Extraction

As mentioned on Section 3, the magnitude/orientation images are computed from the optical flow information. To that end, we extract the optical flow information using the TVL1 algorithm [49], implemented in OpenCV with CUDA. For the sake of comparison, our baseline [33] used the same optical flow algorithm. To obtain the magnitude and orientation images information we empirically set the parameters $h = 15$ and $l = -15$ to compute M; and $h = 180$, $l = -180$ and $m = 128$ to compute $\theta^{'}$.

4.3. Depth Information Estimation

To extract the depth information to be used as weighting scheme for the magnitude information (MOS+D), we used the method provided by Godard et al., [45] with default parameters and the pre-trained model on Cityscapes dataset [50]. Implementation and model were made available by the authors[1]. To obtain the weighted magnitude by depth images information we empirically set the parameters $d = 215$.

4.4. Evaluation

Table 1 report the activity recognition performance of our Magnitude-Orientation Stream (MOS) with VGG-16 architecture in contrast with the baseline on UCF101. This table shows a comparison of our method to the three different streams of our Very Deep Two-Stream (VD2S) baseline [33]: (i) Very Deep Spatial Stream (VDSS); (ii) Very Deep Temporal Stream (VDTS); and Very Deep Two-Stream (VD2S). According to the results, a considerable improvement was achieved with Magnitude-Orientation Stream when compared to the baseline single streams, reaching 90.8% of accuracy on split 1 of the UCF101 dataset. There is an improvement of 5.1 percentage points (p.p.) when compared to the Very Deep Temporal Stream [33] and 11.0 p.p. when compared

[1] `https://github.com/mrharicot/monodepth`

Table 1: Activity recognition accuracy (%) results of Magnitude-Orientation Stream with VGG-16 architecture and Very Deep Two-Stream (VD2S) [33] baseline on UCF101 [21] activity dataset. Results for the baseline were obtained running the code provided by the authors [33]. Note that our results were achieved with only our single Magnitude-Orientation Stream (temporal information) while the results of [33] consider two streams (spatial and temporal information).

		Split 1	**Split 2**	**Split 3**	**Average**
	Approach	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**
Baseline	VDSS [33]	79.8	77.3	77.8	78.4
	VDTS [33]	85.7	88.2	87.4	87.0
	VD2S [33]	90.9	91.6	91.6	**91.4**
Our results	MOS	90.8	89.3	91.5	90.5
	MOS + VDSS [33]	93.1	91.9	92.6	**92.5**
	MOS + VDTS [33]	91.4	92.2	93.6	**92.4**
	MOS + VD2S [33]	93.7	93.1	94.8	**93.8**

to the Very Deep Spatial Stream [33]. This shows that the optical flow pre-processing (i.e., extraction of magnitude and orientation information) brings improvement over using raw optical flow information. Furthermore, it is worth noting that our best result using Magnitude-Orientation Stream on split 1 is close to the best reported (Very Deep Two-Stream), which was achieved by using a combination of two different streams (spatial and temporal informations), while we only used our single Magnitude-Orientation Stream (temporal information). The same observations can be considered when analyzing the results of our temporal stream on splits 2 and 3. Therefore, such results can be considered remarkably good and confirm that pre-processing the inputs helps on guiding the network to extract certain information and although the temporal evolution patterns can be learned implicitly with CNNs, an explicit modeling is preferable and is able to achieve better results.

Figure 4 shows the confusion matrices of Very Deep Spatial Stream, Very Deep Temporal Stream and our Magnitude-Orientation Stream for the UCF101

split 1 (we highlighted the false positives and false negatives to make it more visible on where each method fails). We can observe that our approach fails on classes that are more semantically closer to each other[2], whereas the Very Deep Spatial Stream and the Very Deep Temporal Stream fails in a random manner. In addition, the three methods produce false positives and false negatives different from each other, indicating the possibility of fusion.

To exploit a possible complementarity of the three approaches (very deep spatial stream, very deep temporal stream and our magnitude-orientation stream), we combined the different streams by employing a late fusion technique using a weighted linear combination of their prediction scores. According to the results showed in Table 1, any type of combination performed with our Magnitude-Orientation Stream provides better results than Very Deep Two-Stream [33], with the best result achieving an improvement of 2.4 p.p. over Very Deep Two-Stream [33]. To verify the statistical significance of these combination results, a statistical test for the differences between the means was performed using a Student t-test [51], paired over the dataset splits. The test consists of determining a confidence interval for the differences and simply checking if the interval includes zero (i.e., if the confidence interval does not include zero, the difference is significant at that confidence level). Thus, at 95% confidence level, we can conclude that the difference is significant for our combination results.

We also report the activity recognition performance of our Magnitude-Orientation Stream (MOS) with Inception architecture in comparison with the Temporal Segment Networks (TSN) [37] baseline. Table 2 shows a comparison of our approach and three different streams: (i) Temporal Segment Stream (TSS), (ii) Spatial Segment Stream (SSS), and (iii) Temporal Segment Networks (TSN). According to the results, a considerable improvement was achieved with Magnitude-Orientation Stream when compared to the Temporal Segment Net-

[2]Since the activities on the confusion matrices are sorted according to its labels (e.g., ApplyEyeMakeup, ApplyLipstick, or BaseballPitch, Basketball, BasketballDunk), near regions in the confusion matrix denote semantically closer activities.

Table 2: Activity recognition accuracy (%) results of Magnitude-Orientation Stream with Inception architecture and Temporal Segment Networks (TSN) [37] baseline on UCF101 [21] activity dataset. Results for the baseline were obtained running the code provided by the authors [37]. Note that our results were achieved with only our single Magnitude-Orientation Stream (temporal information) while the results of [37] consider two streams (spatial and temporal information).

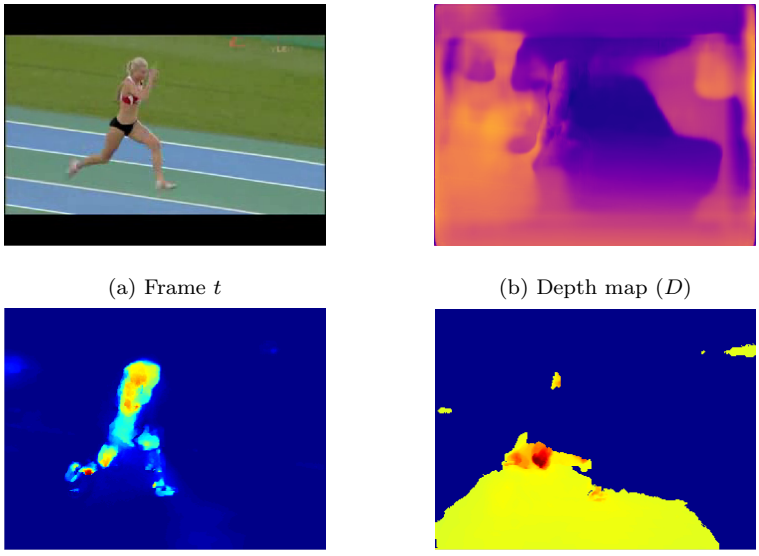
		Split 1	**Split 2**	**Split 3**	**Average**
	Approach	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**
Baseline	SSS [37]	85.5	84.9	84.5	85.1
	TSS [37]	87.6	90.2	91.3	89.7
	TSN [37]	93.5	94.3	94.5	**94.0**
Our results	MOS	91.5	93.0	92.9	92.4
	MOS + SSS [37]	96.2	96.7	96.1	**96.3**
	MOS + TSS [37]	93.4	94.7	94.8	**94.3**
	MOS + TSN [37]	96.5	97.0	96.8	**96.7**

works (TSN) [37] baseline single streams, reaching 92.4% of accuracy on UCF101 dataset. We can note an improvement of 7.3 percentage points (p.p.) when compared to the Spatial Segment Stream (SSS) [37] and 2.7 p.p. when compared to the Temporal Segment Stream (TSS) [37]. Once more, such results confirm that pre-processing the optical flow inputs helps guiding the network to extract a better information.

We also exploited a possible complementarity of the spatial and temporal streams from TSN and our MOS approach. Here, we applied the same late fusion technique used on VGG-16 architecture experiments consisting of a weighted linear combination of the prediction scores. Last line of Table 2 shows the combination results, with the best result achieving 2.7 p.p of improvement when compared to TSN [33]. We verified the statistical significance of these combination results using the Student t-test [51], paired over the dataset splits. We can conclude that, at 95% confidence level, the difference is significant for our combination results.

Table 3: Activity recognition accuracy (%) results of Magnitude-Orientation Stream with Inception architecture with and without depth weighting on UCF101 [21] activity dataset.

	Split 1	**Split 2**	**Split 3**	**Average**
Approach	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**
MOS	91.5	93.0	92.9	92.4
MOS+D	88.6	89.8	89.9	89.4



(a) Frame t (b) Depth map (D)

(c) Magnitude (M) (d) Harmed Magnitude weighted by depth ($M^{'}$)

Figure 5: Comparison between magnitude information affected when weighted by poorly estimated depth maps.

Table 3 shows the results achieved with our proposed magnitude weighted by depth scheme (MOS+D) on UCF101 dataset. We note a difference of 3.0 p.p. smaller than our main method. Such worse results are due to poorly estimated depth maps. Although the magnitude weighted depth scheme circumvent problems related to activities taken regardless of their distance in relation to the camera, as shown in Figure 3, if the depth maps were not good estimated it can harm too much the magnitude information, as can be seen in Figure 5.

Table 4: Activity recognition accuracy comparison on UCF101 activity dataset [21].

	Approach	UCF101 Acc. (%)
Handcrafted Methods	HOF + BoW [7]	61.8
	HOG-HOF + BoW [7]	71.8
	MBH + BoW [24]	77.1
	GBH + BoW [26]	68.5
	HOG3D + BoW [8]	61.4
	HOF + FV [7]	65.9
	HOG-HOF + FV [7]	75.4
	MBH + FV [24]	81.0
	GBH + FV [26]	74.2
	HOG3D + FV [8]	64.7
	IDT [25]	85.9
	IDT + higher FV [52]	87.9
	IDT + MVSV [53]	83.5
NN Methods	Deep Networks [30]	65.4
	Composite LSTM [54]	75.8
	C3D [32]	85.2
	Factorized CNN [55]	88.1
	Two-Stream [17]	88.0
	Two-Stream F [14]	92.5
	KVMF [56]	93.1
	TSN (3 modalities) [37]	94.2
	Two-Stream I3D [43]	**98.0**
Our results	MOS (VGG-16)	90.5
	MOS (VGG-16) + VD2S	93.8
	MOS (Inception)	92.4
	MOS+D (Inception)	89.4
	MOS (Inception) + TSN	**96.7**

Table 4 [3] presents results on UCF101 dataset for many works. The first part of the table shows results of methods that extract temporal information using handcrafted features. We compare our MOS approach with the results of local feature-based methods, such as Bag-of-Words (BoW) + features, Fisher vector (FV) + features, and Improved Dense Trajectories (IDT). The best result by such type of methods was achieved with IDT + higher FV [52], reaching 87.9%. Our best result using the proposed approach combined with Very Deep Two-Stream outperforms that by 5.9 percentage points.

The second part of Table 4 shows the results achieved with neural networks (NN) approaches. According to the results, by only using our Magnitude-Orientation Stream (MOS), we outperform many methods ([30, 54, 32, 55, 17, 37]). It is worth mentioning that we also improved the results achieved by the original two-stream [17]. Using the VGG-16 architecture, we outperform it by 2.5 p.p. (temporal stream) and by 5.8 p.p. (combining it with Very Deep Two-Stream). Further, using the Inception architecture, we outperform it by 4.4 p.p. (temporal stream) and by 8.7 p.p. (combining it with TSN). Finally, we can observe that our best result only did not outperform Carreira and Zisserman I3D method [43]. However, it is important to emphasize that they used a huge dataset for pre-training. Nevertheless, we believe our results are remarkably good since 3D convolutional operations are more computationally expensive than the 2D convolutional operations used in our approach. For instance, the Two-Stream I3D network used by Carreira and Zisserman [43] has 25 million parameters, while the 2D Two-Stream employed by us has less than a half (12 million parameters).

To show that the poor results achieved on UCF101 dataset with our magnitude weighted by depth scheme were caused by poorly estimated depth maps, here we employed the NTU dataset which provides accurate depth maps captured by a depth sensor (Kinect). Table 5 shows the results achieved on NTU

[3] Results for features + BoW were obtained from [26] and features + FV were obtained from [57].

Table 5: Activity recognition accuracy (%) results of Magnitude-Orientation Stream with Inception architecture with and without depth weighting scheme on NTU [22] activity dataset.

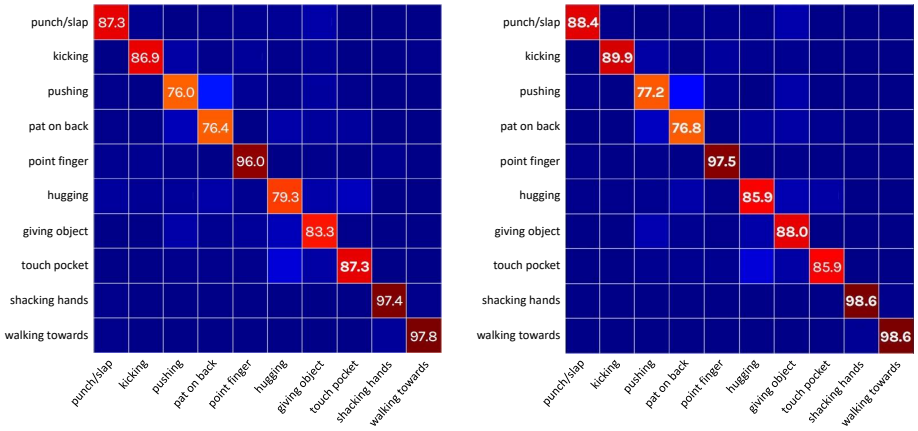
	Approach	Cross-subject Acc. (%)
Literature Results	Geometric features [58]	70.3
	VA-LSTM [59]	79.4
	CMN [42]	80.8
	STA-hands [60]	**82.5**
Our Results	MOS	73.1
	MOS+D	72.6
	MOS + MOS+D (Non-weighted Linear Fusion)	**75.4**

dataset with and without our weighting scheme by using the cross-subject evaluation protocol. We can note very similar results achieved by both methods, however, when a non-weighted linear fusion is applied on the methods we can achieve an improvement of 2.3 p.p. when compared to MOS. This shows that, although the methods achieved very similar results, the network is learning different information from each data. Figure 4.4 illustrates a part of the confusion matrices focused on activities that involve interactions of two people showing that for all these activities, the non-weighted linear fusion improved the results. Such improvement can be considered thanks to the depth information, since in such activities people are in different depth planes (see Figure 4.4). Again, we emphasize here that our intention to use the NTU dataset was only to validate that use of accurate the depth information in our approach leads to better learning for motion in different depth planes.

black

4.5. Discussion

To better analyze our proposed approach, we take a closer look at activities from UCF101 that our method achieved higher performance than the baselines. For instance, some activities that were most correctly classified by MOS and



(a) MOS (b) Non-weighted Linear Fusion

Figure 6: Confusion matrices from NTU dataset focused on activities that involves interactions of two people.

(a) Kicking (b) Touch Pocket (c) Shaking Hands

Figure 7: Example of activities involving interactions of two people on NTU dataset. As can be noted, people are in different depth planes during such activities.

misclassified by the baselines are *apply lipstick, front crawl, basketball, shaving beard, rafting,* among others. We note that the baselines usually confused the activities *apply lip stick* and *shaving beard* with *haircut*, *brushing teeth* or *apply eye makeup* which are activities with movements on very close areas. Moreover, the baselines confused the activity *front crawl* with *breast stroke*, which are both swimming styles. Another interesting analysis is the confusion from the activity *basketball* and *volleyball spiking*, where we can note that the confusion lies on the fact that both activities start with a jump followed by a arm movement with a ball. Furthermore, the activity *rafting* is highly confused with *kayaking*

which can be explained by the fact that both happens in a river with some type of boat but differs in velocity.

The correctly classifications of such activities by our MOS approach show that feeding the network with explicit orientation information instead of x and y displacements, could improve the classification of activities with movements on very close areas or even with similar movements. Besides, we can note the importance of using magnitude information (velocity), since the velocity information can be used to distinguish between closer activities with different velocities.

We also investigated the cases where our method failed. The most misclassified activities correspond to cases, such as *cricket bowling, pizza tossing, walking with dog* and activities involving playing an instrument. Our method confused *cricket bowling* with *bowling*, in which both activities are composed by movements with the arm with a ball. In addition, the activity *pizza tossing* is confused with many other activities. Furthermore, the analysis of the misclassified videos revealed that the method presented difficulties with activities with very similar movements differentiating by the object used, such as playing instrument activities (cello, guitar and sitar; or daf, dhol and tabla). The same difficulties were also noted on the baseline methods. Another misclassification of our approach is *walking with dog* with *horse riding*. Such analysis indicates that the use of object information could help enhancing the classification.

5. Conclusions

In this work, we proposed a novel temporal stream for two-stream convolutional networks, named Magnitude-Orientation Stream (MOS). The method is based on simple non-linear transformations on the optical flow components generating input images composed of magnitude and orientation information. The spatial relationship contained on local neighborhoods of magnitude and orientation captures not only displacement, by using orientation, but also magnitude providing information regarding the velocity of the movement. Moreover,

we present a weighting scheme that weight the magnitude by the depth information in order to circumvent problems related to activities taken regardless of their distance to the camera. We demonstrated that MOS outperforms all classic approaches based on local handcrafted features of the literature. Furthermore, simply by using only our temporal stream, we outperform original CNN two-stream approaches based on temporal and spatial information as well as other recent works that employ neural networks, suggesting its suitability to learn temporal information. Another interesting finding is that the combination of our temporal stream with the Very Deep Two-Stream and also Temporal Segment Networks methods improves the activity recognition.

Acknowledgments

The authors would like to thank the Brazilian National Research Council – CNPq (Grants 311053/2016-5, 204952/2017-4 and 438629/2018-3), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project). The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the GeForce Titan X GPU used for this research.

References

References

[1] H. Keval, CCTV Control Room Collaboration and Communication: Does it Work?, in: Human Centred Technology Workshop, 2006.

[2] S. Danafar, N. Gheissari, Action recognition for surveillance applications using optic flow and SVM, in: ACCV, 2007.

[3] V. Reddy, C. Sanderson, B. Lovell, Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture, in: CVPRW, 2011.

[4] A. Wiliem, V. Madasu, W. Boles, P. Yarlagadda, A suspicious behaviour detection using a context space model for smart surveillance systems, Comput. Vis. Image Underst.

[5] J. Wang, Z. Xu, Spatio-temporal texture modelling for real-time crowd anomaly detection, Computer Vision and Image Understanding.

[6] P. Scovanner, S. Ali, M. Shah, A 3-dimensional Sift Descriptor and Its Application to Action Recognition, in: ACM MM, 2007.

[7] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: CVPR, 2008.

[8] A. Kläser, M. Marszałek, C. Schmid, A Spatio-Temporal Descriptor Based on 3D-Gradients, in: BMVC, 2008.

[9] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: CVPR, 2011.

[10] J. Sivic, A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos, in: ICCV, 2003.

[11] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image Classification with the Fisher Vector: Theory and Practice, Int. J. Comput. Vision.

[12] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: NIPS, 2012.

[13] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, in: CVPR, 2015.

[14] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: CVPR, 2016.

[15] E. Park, X. Han, T. L. Berg, A. C. Berg, Combining multiple sources of knowledge in deep CNNs for action recognition, in: WACV, 2016.

[16] A. Diba, A. M. Pazandeh, L. Van Gool, Efficient Two-Stream Motion and Appearance 3D CNNs for Video Classification, in: ECCV, 2016.

[17] K. Simonyan, A. Zisserman, Two-stream Convolutional Networks for Action Recognition in Videos, in: NIPS, 2014.

[18] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[19] C. Li, Q. Zhong, D. Xie, S. Pu, Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation, in: IJCAI, 2018.

[20] C. A. Caetano, V. H. C. D. Melo, J. A. dos Santos, W. R. Schwartz, Activity recognition based on a magnitude-orientation stream network, in: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2017.

[21] K. Soomroand, A. R. Zamir, M. Shah, UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild, Tech. rep., CRCV-TR (2012).

[22] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, Ntu rgb+d: A large scale dataset for 3d human activity analysis, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[23] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005.

[24] N. Dalal, B. Triggs, C. Schmid, Human Detection Using Oriented Histograms of Flow and Appearance, in: ECCV, 2006.

[25] H. Wang, C. Schmid, Action Recognition with Improved Trajectories, in: ICCV, 2013.

[26] F. Shi, R. Laganiere, E. Petriu, Gradient Boundary Histograms for Action Recognition, in: WACV, 2015.

[27] R. V. H. M. Colque, C. Caetano, W. R. Schwartz, Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos, in: SIBGRAPI, 2015.

[28] R. V. H. M. Colque, C. Caetano, M. T. L. de Andrade, W. R. Schwartz, Histograms of Optical Flow Orientation and Magnitude and Entropy to Detect Anomalous Events in Videos, IEEE Transactions on Circuits and Systems for Video Technology.

[29] C. Caetano, J. A. dos Santos, W. R. Schwartz, Optical Flow Co-occurrence Matrices: A novel spatiotemporal feature descriptor, in: ICPR, 2016.

[30] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-Scale Video Classification with Convolutional Neural Networks, in: CVPR, 2014.

[31] S. Ji, W. Xu, M. Yang, K. Yu, 3D Convolutional Neural Networks for Human Action Recognition, IEEE Trans. Pattern Anal. Mach. Intell.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning Spatiotemporal Features With 3D Convolutional Networks, in: ICCV, 2015.

[33] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, Towards Good Practices for Very Deep Two-Stream ConvNets, CoRR.

[34] M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, Springer International Publishing, 2014, pp. 818–833.

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper With Convolutions, in: CVPR, 2015.

[36] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: ICLR, 2015.

[37] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, in: ECCV, 2016.

[38] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift., in: ICML, 2015.

[39] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, A. Rocha, Video pornography detection through deep learning techniques and motion information, Neurocomputing.

[40] I. E. G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia, John Wiley & Sons, Inc., 2003.

[41] Y. Zhu, S. Newsam, Depth2action: Exploring embedded depth for large-scale action recognition, in: Computer Vision – ECCV 2016 Workshops, 2016.

[42] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, T. Brox, Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection, in: International Conference on Computer Vision (ICCV), 2017.

[43] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the Kinetics dataset, in: CVPR, 2017.

[44] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., The kinetics human action video dataset, Tech. rep., arXiv preprint arXiv:1705.06950 (2017).

[45] C. Godard, O. Mac Aodha, G. J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: CVPR, 2017.

[46] J. Wang, Z. Liu, Y. Wu, Human Action Recognition with Depth Cameras, Springer Publishing Company, Incorporated, 2014.

[47] B. Liang, L. Zheng, A survey on human action recognition using depth sensors, in: International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2015.

[48] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: A large video database for human motion recognition, in: ICCV, 2011.

[49] C. Zach, T. Pock, H. Bischof, A Duality Based Approach for Realtime TV-L1 Optical Flow, in: Proceedings of the 29th DAGM Conference on Pattern Recognition, 2007.

[50] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[51] R. Jain, The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling., Wiley, 1991.

[52] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice, Computer Vision and Image Understanding.

[53] Z. Cai, L. Wang, X. Peng, Y. Qiao, Multi-view Super Vector for Action Recognition, in: CVPR, 2014.

[54] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised Learning of Video Representations Using LSTMs, in: ICML, 2015.

[55] L. Sun, K. Jia, D. Y. Yeung, B. E. Shi, Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks, in: ICCV, 2015.

[56] W. Zhu, J. Hu, G. Sun, X. Cao, Y. Qiao, A Key Volume Mining Deep Framework for Action Recognition, in: CVPR, 2016.

[57] F. Shi, Local Part Model for Action Recognition in Realistic Videos, Ph.D. thesis, School of Electrical Engineering and Computer Science, Faculty of Engineering, University of Ottawa (2014).

[58] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive recurrent neural networks for high performance human action recognition from skeleton data, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

[59] S. Zhang, X. Liu, J. Xiao, On geometric features for skeleton-based action recognition using multilayer lstm networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017.

[60] F. Baradel, C. Wolf, J. Mille, Human action recognition: Pose-based attention draws focus to hands, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017.