INTERPRETABLE HUMAN ACTION RECOGNITION IN COMPRESSED DOMAIN

Vignesh Srinivasan[1,2], *Sebastian Lapuschkin*[1], *Cornelius Hellge*[1], *Klaus-Robert Müller*[2,3], and *Wojciech Samek*[1,2]

[1]Department of Video Coding & Analytics, Fraunhofer Heinrich Hertz Institute, Berlin, Germany
[2]Berlin Big Data Center, Berlin, Germany
[3]Department of Computer Science, Technische Universität Berlin, Germany

ABSTRACT

Compressed domain human action recognition algorithms are extremely efficient, because they only require a partial decoding of the video bit stream. However, the question *what* exactly makes these algorithms decide for a particular action is still a mystery. In this paper, we present a general method, Layer-wise Relevance Propagation (LRP), to understand and interpret action recognition algorithms and apply it to a state-of-the-art compressed domain method based on Fisher vector encoding and SVM classification. By using LRP, the classifiers decisions are propagated back every step in the action recognition pipeline until the input is reached. This methodology allows to identify *where* and *when* the important (from the classifier's perspective) action happens in the video. To our knowledge, this is the first work to interpret a compressed domain action recognition algorithm. We evaluate our method on the HMDB51 dataset and show that in many cases a few significant frames contribute most towards the prediction of the video to a particular class.

Index Terms— Action recognition, interpretable classification, motion vectors, fisher vector encoding, compressed domain

1. INTRODUCTION

Videos are an integral part of our daily lives. This has in turn created a huge demand in content driven analysis, e.g., for surveillance and copyright protection but also for classifying videos into different human action categories by automated annotation systems. Action recognition algorithms infer the action performed by a human in a video using visual cues which are gathered in the form of features. Hand crafted features like the Bag of Words (BOW) model [1], Scale Invariant Feature Transform (SIFT) [2], Histogram of Oriented Gradients (HOG) [3], Motion Boundary Histogram (MBH) [4] and Fisher vectors (FV) [5] are some of the widely used techniques for action recognition. These features are typically classified using a linear or non-linear Support Vector Machine (SVM) [6, 7]. Widely used deep learning strategies for action recognition include [8, 9, 10, 11].

In order to reduce the computational overhead of decoding the video as well as extracting and processing its frames, motion vectors from the compressed domain are used to analyze the video content. Compared to a pixel domain approach, [12] recorded an increase in speed of two orders of magnitude with only a slight decrease in accuracy (if all videos are encoded with the same codec [13]) using motion vectors.

The nonlinear transformations in computing the features often lead to a lack of straightforward interpretability of the classifiers decisions. Methods designed for interpreting BOW pipelines include e.g. [14] – describing a system to explain models incorporating vector quantization, histogram intersection kernels and SVMs – or the work of [15] – presenting an algorithm for identifying connected image regions (support regions) critical for the prediction of a linear classifier on top of a max-pooling feature aggregation step. While these methods are limited in their range of applicability, general *explanation* techniques such as Layer-wise Relevance Propagation (LRP) [16] and Deep Taylor [17] have been recently introduced and applied to image, text and biomedical signal classification problems [18, 19, 20]. These methods can be adapted to wide range of configurations for both BOW-type classifiers and deep neural networks.

In this paper, we utilize the LRP method [21] in the context of action recognition in compressed domain [12]. The classifier decisions are propagated back every layer in the classification process in the form of relevances to the Fisher vectors, local descriptors and finally to the input voxels. The motivation behind applying LRP to videos is

- *Localization*: Pinpoint the exact location of action in the video by highlighting voxels with high relevance.
- *Significant frames identification*: Identify frames that contribute most for the algorithms to conclude for a given action.
- *Feature ranking*: Compute how much each feature contributes to the output of the algorithm.
- *Visualization*: Examine the relevances to help unravel what the algorithm has learned.

This will enable identification and localization of the exact visual cues that the algorithm looks for in the frames when classifying a given video to a particular action.

2. MODEL AND EXPLANATION

Fig. 1 gives an overview of the action recognition model [12] and also the LRP algorithm [16]. The motion vectors are used to compute spatio-temporal features − Histogram Of Flow (HOF) and Motion Boundary Histogram (MBH). To compute these features, histograms of motion vectors from an overlapping $32 \times 32 \times 5$ spatio-temporal cube are considered. Both HOF and MBH consist of eight motion bins for different orientations and one no-motion bin. The descriptors for a video are obtained by stacking the histograms over all the cubes over all the time slices. MBH is computed by a derivative of the flow. MBHx and MBHy, the x and y derivatives, have been

This work was supported by the German Ministry for Education and Research as Berlin Big Data Center BBDC (01IS14013A) and by the German Ministry for Economic Affairs and Energy as Celtic-Plus Project VIRTUOSE. This publication only reflects the authors views. Funding agencies are not liable for any use that may be made of the information contained herein.

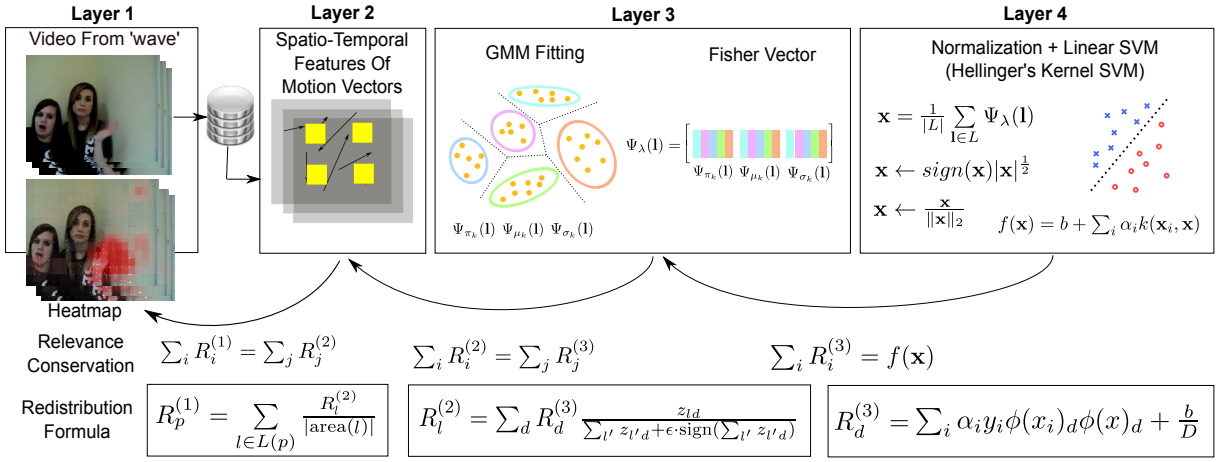

Fig. 1. FV computation and explaining the classifier decision through LRP. The motion vectors are used to compute the spatio-temporal features – HOF and MBH, which are in turn used to perform FV mapping using a GMM. The output of the linear SVM classifier is used to compute relevances. The relevances are propagated further until contributions made by each voxel is obtained. In the heatmap, which is overlaid on the frame, *red* color indicates positive relevance while *blue* indicates negative relevance.

shown to reduce the influence of camera motion in action recognition [4]. These are then mapped to FV, a robust and state-of-the-art feature mapping approach widely used in computer vision and video processing [5]. After power- and ℓ_2 normalization of the FV, a linear SVM classifier is used to classify the videos. The mean average precision (MAP) obtained for this dataset was 42.77%.

LRP [16] aims to decompose the predictions of a trained classifier in terms of mappings performed during prediction time, in order to attribute to each input component x_i a proportional share

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{z_j} R_j^{(l+1)} \text{ with } z_j = \sum_{i'} z_{i'j} + b_j \qquad (1)$$

by which it contributes to the classifier output, explaining its relevance to the final prediction in its given state. Here, z_{ij} signifies the value of some forward mapping operation[1] of an input component i at layer l to component j in layer $(l+1)$ and b_j is a bias term. Above formula is then applied iteratively – under consideration of the classifier architecture and beginning at the last layer of computation performed by the model – until the input layer is reached. The algorithm is initialized with $R_j^{(l+1)} = f(x)$ as the top layer relevance input. This fulfills the layer-wise conservation property [16]

$$\forall l : \sum_i R_i^{(l)} = f(x), \qquad (2)$$

ensuring implicitly normalized relevance values within all layers of computation. Positive values $R_i^{(l)}$ speak for the presence of a prediction target in input component i and negative values against it. Note that equation 1 is the most basic decomposition rule from [16] and can be extended to support max-pooling instead of sum-pooling feature aggregation.

The authors of [18] have successfully applied LRP to a FV-based object detection system for single images, revealing unexpected and strongly biased prediction strategies of the learned model in direct comparison to several deep neural networks and discovering flaws in the PASCAL VOC 2007 [22] data set. Quantitative results have shown that for FV mapping, the introduction of an additional numerical stabilizer term to equation 1 results in pixel-level relevance maps representing the classifier reasoning more accurately. The LRP formulas used in this work are shown in Fig. 1.

[1]Fig. 1 shows three forward mappings: (1) motion vector → descriptor, (2) descriptor → FV, (3) FV → SVM output

3. HEATMAP COMPUTATION

This section provides a step by step approach to compute LRP for videos, following a more in-depth explanation of the process shown in Fig. 1.

3.1. Global Descriptor Level Relevances

The model to explain [12] is a linear SVM classifier on top of an *improved* FV mapping layer, as given in Fig. 1. That is, after computing the FV mappings and sum-aggregating over all global descriptors, power- and ℓ_2 normalization steps are applied, which have been shown to reduce the sparsity of the descriptor and increase model accuracy. This is equivalent to applying a Hellinger's kernel function to the unnormalized FV [5]. We compute global descriptor level relevances $R_d^{(3)}$, as given in Fig. 1. $R_d^{(3)}$ are the relevances per dimension d of the FV passed through the predictor, where the decision function of the model is decomposed in its dual form. The kernel mapping $\Phi(\cdot)$ realizes the normalization steps applied after the FV mapping step and i is indexing the model's support vector parameters.

3.2. Local Feature Level Relevances

In order to compute local feature level relevances $R_l^{(2)}$, we make use of z_{ld}, which describes the output of the FV mapping step of a descriptor l to output dimension d. For numerical stabilization, we extend the denominator in equation 1 to $z_j + \epsilon \cdot \text{sign}(z_j)$, where $\epsilon = 100$ and the sign-function is adapted such that $\text{sign}(x \geq 0) = 1$ and -1 otherwise, to avoid divisions by very small values due to compensating mappings z_{ld} [16, 18]. Note that choosing appropriate parametrizations for the decomposition, complementing the forward mappings of the classifier, is critical to achieving the best possible results.

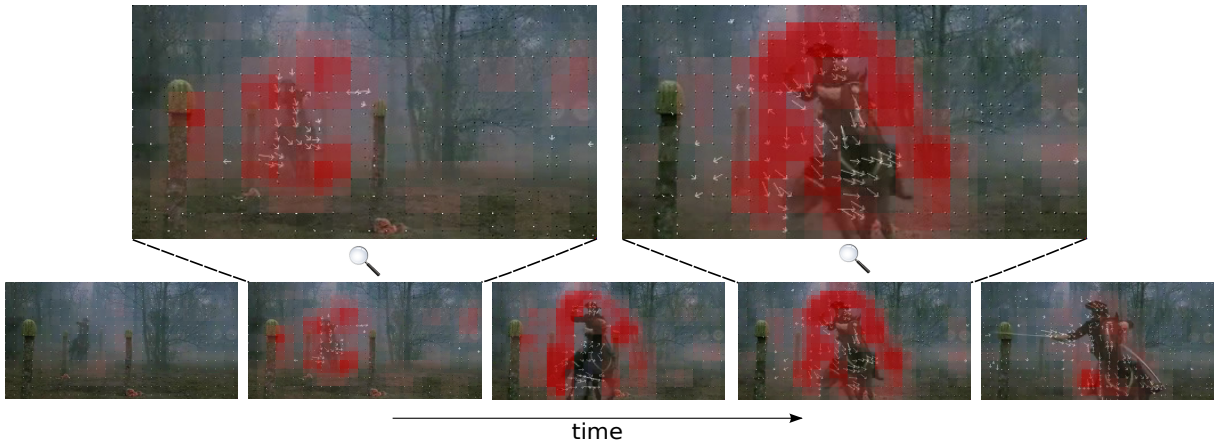

Fig. 2. The relevances from the final heatmapping layer $R^{(1)}_{(v,t)}$ computed at different time instances. The above figure is taken from a test video from the class *ride horse* with the relevances and corresponding motion vectors overlaid on top of the frame representing an actor riding on a horse moving towards the camera.

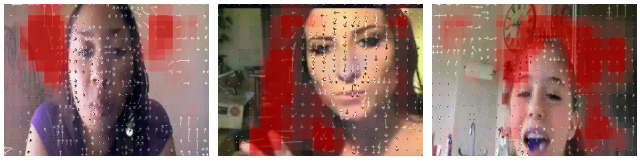

Fig. 3. Frames with the highest relevance $R^{(1)}$ from different videos for the class *chew*.

3.3. Frame Level Relevances

Since the local features are computed from an overlapping spatio-temporal grid of size $32 \times 32 \times 5$, $R^{(2)}_l$ are distributed over these different voxels in the video as $R^{(1)}_{(v,t)}$, where v describes a voxel coordinate at time t. All the pixels in a voxel share the same relevance. This is the main reason for the block shaped structures visible in all the heatmap figures given here.

3.4. Feature-wise Frame Level Relevances

The action recognition model in [12] makes use of HOF, MBHx and MBHy features. The local feature relevance $R^{(2)}_l$, can be considered as an augmentation of the relevances of each feature. Since the features here are stacked, feature-wise relevance $R^{(2)}_{(l,f)}$, where f corresponds a particular local feature, can be obtained by taking an appropriate subset of $R^{(2)}_{(l)}$. $R^{(2)}_{(l,f)}$ can again be distributed over the voxels in the video.

4. ANALYSIS

By keeping mind the motivation for applying LRP in videos as elaborated in section. 1, we analyze different aspects and advantages of computing the relevance scores in videos.

4.1. Localization

Heatmaps computed by LRP are an intuitive way to understand and visualize what the algorithm has learned. LRP for SVM classification provides positive, negative and neutral evidence for a given class. Positive relevances computed for true positive videos lead us the particular voxels where action (from the classifier's perspective) occurs. Fig. 2 displays the movement of an actor riding a horse in a given video. The positive relevances displayed by *red* provides for an insightful method to identify and localize when and where the action is performed in a video. The localization of the action can also be found in Fig. 4a and 4c, where the actor performs action *pushup* and *hit*, respectively. High relevance can be observed over and around the actor's upper body as the action is being performed.

4.2. Significant Frames Identification

Given a video, all the frames are given as input to the algorithm. Although, to decide on the output, the algorithm needs not take into account all the frames equally. Frames that are most crucial for the algorithm can be found by aggregating the relevance over each frame. Fig. 6 shows the sum of relevances for each frame plotted against the number of frames in a video for the class *sit-up*. Frames with an upward and downward motion of the actor produce a high relevance score, while the frames where the actor pauses get lower relevance.

4.3. Feature Rank

Features that contribute most to the output of the algorithm can also be found by computing feature-wise frame level relevances. Fig. 4 gives a sample frame from 4 videos of different classes after distributing the relevances of each feature to the input voxels in the LRP process. As can be seen in all the subfigures in Fig. 4, videos were found to have different relevances for different features. This was also confirmed by computing the contribution of each feature for all the videos, as shown in Fig. 5. MBHy displayed high relevances while HOF obtained the least relevance score indicating that contriubtion of MBHy was more accounted for by the classifier for this particular dataset. This can be attributed to two factors - MBH is a derivative of the flow and hence is already a more robust feature. Another factor is from an intuitive observation that many classes in this dataset had an actor performing vertical motion like *sit up* and *pushups*. In Fig. 4a, since the actor performs pushups, most motion vectors are found to be in the vertical direction. Hence, contributing for higher MBHy relevance.

Fig. 4. Feature-wise $R^{(1)}$ plotted for a frame with high relevance from a video belonging to class *pushup, turn, hit* and *golf* respectively. The top row shows relevances of MBHy, while the center and bottom row represents relevances of HOF and MBHx, respectively.

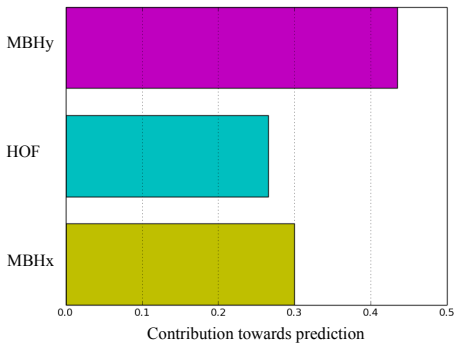

Fig. 5. The contribution of each feature towards classifiers decision for all true positive videos in the dataset

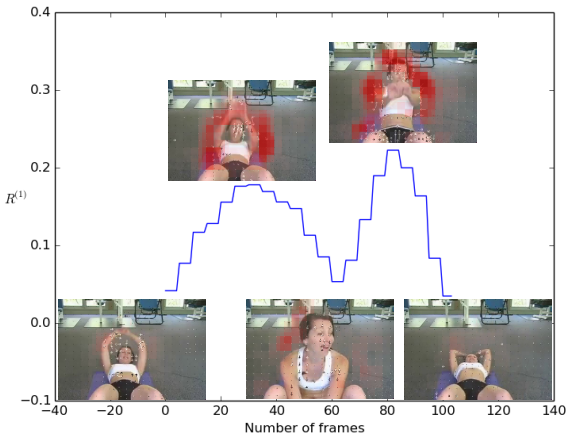

Fig. 6. Sum of relevance over each frame plotted against number of frames in the video for the action *sit-up*.

4.4. Visualization

Heatmaps can also be used to unravel and visualize the learning process of the algorithm. Fig. 3 gives the heatmap for a few true positive videos from the class *chew*. Since this approach makes use of only the motion vectors, large motion vectors can skew the histogram components. The videos from *chew* which have a close-up shot of a person chewing, exhibit this phenomenon. Motion vectors observed in videos from *chew* appear to have relatively large motion vectors from the movement of head compared to the motion vectors due to the act of chewing. Incidentally, the heatmap also produces the strongest relevance in that region. This indicates that the algorithm has found and learned a common pattern in videos from the class *chew* - that of small movement in head - rather than chewing.

The heatmaps from the class *golf* also provided for some interesting observations as shown in Fig. 4d. The algorithm has learned the motion of the swing of an object correctly by using spatio-temporal features. However, in a video, coincidentally, the swing of a towel obtained a high relevance indicating that the algorithm used it to classify the video as belonging to class *golf*.

5. DISCUSSION AND FUTURE WORK

In this work, we have presented the LRP method to interpret and understand the predictions of a compressed domain action recognition algorithm. LRP efficiently propagates the classifiers decisions back to voxels in the video, thus finding the contribution of the different voxels in the form of relevances. We demonstrated localization of the action performed by identifying the significant voxels and frames. In addition feature-wise relevance was computed demonstrating the contribution made by each feature towards the classifiers decision.

Future work will use the heatmaps as cue for recomputing the features and classify the videos again from voxels with relatively high relevance, in this manner unsupervised preprocessing via LRP may ultimately contribute to denoising and thus higher accuracy.

6. REFERENCES

[1] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *IJCV*, vol. 79, no. 3, pp. 299–318, 2008.

[2] I. Laptev, "On space-time interest points," *IJCV*, vol. 64, no. 2-3, pp. 107–123, 2005.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, vol. 1, pp. 886–893.

[4] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *IJCV*, vol. 103, no. 1, pp. 60–79, 2013.

[5] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Computer Vision–ECCV 2010*, pp. 143–156. Springer, 2010.

[6] K-R Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, 2001.

[7] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Adv. in NIPS*, 2014, pp. 568–576.

[9] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE TPAMI*, vol. 35, no. 1, pp. 221–231, 2013.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. CVPR*, 2014, pp. 1725–1732.

[11] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014.

[12] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding, and classification for action recognition," in *Proc. CVPR*, 2014, pp. 2593–2600.

[13] V. Srinivasan, S. Gül, S. Bosse, J. Meyer, T. Schierl, C. Hellge, and W. Samek, "On the robustness of action recognition methods in compressed and pixel domain," in *Proc. EUVIP*, 2016, pp. 1–6.

[14] J.R.R. Uijlings, A.W.M. Smeulders, and R.J.H. Scha, "The visual extent of an object," *IJCV*, 2012.

[15] L. Liu and L. Wang, "What has my classifier learned? visualizing the classification rules of bag-of-feature model by support region detection," in *Proc. CVPR*, 2012, pp. 3586–3593.

[16] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, pp. e0130140, 2015.

[17] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, 2016, in press.

[18] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "Analyzing classifiers: Fisher vectors and deep neural networks," in *Proc. CVPR*, 2016, pp. 2912–2920.

[19] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, "Interpretable deep neural networks for single-trial eeg classification," *Journal of Neuroscience Methods*, vol. 274, pp. 141–145, 2016.

[20] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek, "Explaining predictions of non-linear classifiers in nlp," in *Proc. of Workshop on Representation Learning for NLP*. 2016, pp. 1–7, Association for Computational Linguistics.

[21] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "The lrp toolbox for artificial neural networks," *Journal of Machine Learning Research*, vol. 17, no. 114, pp. 1–5, 2016.

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.