

DEEP TEMPORAL PYRAMID DESIGN FOR ACTION RECOGNITION

Ahmed Mazari

Hichem Sahbi

Sorbonne University, CNRS LIP6, Paris, France

ABSTRACT

Deep convolutional neural networks (CNNs) are nowadays achieving significant leaps in different pattern recognition tasks including action recognition. Current CNNs are increasingly deeper, data-hungrier and this makes their success tributary of the abundance of labeled training data. CNNs also rely on max/average pooling which reduces dimensionality of output layers and hence attenuates their sensitivity to the availability of labeled data. However, this process may dilute the information of upstream convolutional layers and thereby affect the discrimination power of the trained representations, especially when the learned categories are fine-grained.

In this paper, we introduce a novel hierarchical aggregation design, for final pooling, that controls granularity of the learned representations w.r.t the actual granularity of action categories. Our solution is based on a tree-structured temporal pyramid that aggregates outputs of CNNs at different levels. Top levels of this hierarchy are dedicated to coarse categories while deep levels are more suitable to fine-grained ones. The design of our temporal pyramid is based on solving a constrained minimization problem whose solution corresponds to the distribution of weights of different representations in the temporal pyramid. Experiments conducted using the challenging UCF101 database show the relevance of our hierarchical design w.r.t other related methods.

Index Terms— deep representation learning, temporal pyramids, video action recognition

1. INTRODUCTION

Video action recognition is a major challenge in computer vision which consists in assigning labels (also known as classes or categories) to sequences of video frames. The challenge in action recognition stems from (i) the difficulty to learn mapping models that assign action categories to frames while being resilient to their acquisition conditions (namely occlusion, illumination, spatial-temporal resolution/scale/length, camera motion and velocity, truncation, background clutter, etc.) and also (ii) the hardness in hand-labeling large collections of training videos prior to build these mapping models.

Existing action recognition techniques are usually based on machine learning [4, 5, 1, 8, 28]; their general principle consists first in describing video frames using handcrafted

or learned representations and then assigning these representations to action categories using variety of machine learning and inference techniques including support vector machines [14, 8, 27, 24, 17, 25] and deep networks [13, 12, 2, 11, 6, 16]. In particular, deep learning models are successful but their success is highly dependent on the availability of large collections of hand-labeled videos¹ which are usually difficult to collect and expensive – even at reasonable frame rates – especially when handling fine-grained action categories. As a result, existing labeled training sets, for action recognition, are at least two orders of magnitude smaller compared to other neighboring tasks (such as the well studied image classification) while action recognition is intrinsically more challenging. Furthermore, training and fine-tuning these models, together with their hyper-parameters for the challenging task of action recognition, is known to be memory and time demanding even when using highly efficient GPU resources and reasonable size videos.

The increase in the discrimination power of the aforementioned convolutional networks (due to an increase in the number of their parameters) comes to the detriment of an increase of their sensitivity to the acquisition conditions especially on challenging datasets such as the UCF101. Hence, these networks become more data-hungry and more subject to over-fitting. Pooling based on average or max operators, also known as aggregation, attenuates such effect and makes it possible to reduce the sensitivity and hence enhances the resilience of these CNNs to the lack of training data and thereby to the acquisition conditions. However, pooling produces a downside effect: a loss in the discrimination power especially when videos belong to fine-grained action categories. Put differently, convolutional layers without pooling help discriminating fine-grained categories while pooling helps discriminating coarse-grained categories; choosing a granularity for pooling is clearly a challenging task that *requires an appropriate design and this constitutes the main contribution of this work*.

In this paper, we introduce a novel hierarchical aggregation design that balances the discrimination power of CNN outputs and their resilience to video acquisition conditions. Our solution is based on a temporal pyramid that aggregates the outputs of CNNs at different levels, resulting into a hierarchical representation. Top levels of this hierarchy are

¹that cover all the intra-class variability of action categories.

dedicated to coarse action categories while deep levels are dedicated to fine-grained ones. The design principle of our temporal pyramid is based on solving a constrained minimization problem whose solution corresponds to the distribution of weights of different representations in the temporal pyramid. Experiments conducted, on action recognition, using the challenging UCF101 database show the substantial gain and the complementary aspect of our hierarchical design w.r.t other related methods [3, 13].

2. PROPOSED METHOD

Considering a collection of videos $\mathcal{S} = \{\mathcal{V}_i\}_{i=1}^n$, with each one being a sequence of frames $\mathcal{V}_i = \{f_{i,t}\}_{t=1}^{T_i}$; in this paper, and unless explicitly mentioned, the symbol i is omitted and $\mathcal{V}_i, f_{i,t}$ are simply rewritten as \mathcal{V}, f_t respectively. As shown subsequently, frames in \mathcal{V} are described using “end-to-end” trained network representations (see details in the subsequent sections). Without a loss of generality, we assume T_i constant and simply denoted as T ; otherwise frame sampling could be achieved to make T_i constant.

In what follows, we first present the branches used to build a representation at the frame-level. Then, we show how these frame-level representations are aggregated and combined at the video-level in order to achieve highly effective action recognition. Finally, we discuss our hierarchical aggregation design and mainly the learning of its parameters.

2.1. Deep frame-wise representations

In order to describe the visual content of a given video \mathcal{V} , we rely on a two-stream process; the latter provides a complete description of *appearance* and *motion*, based on [13, 3], that characterizes the spatio-temporal aspects of moving objects and their interactions. The output of the appearance stream (denoted as $\{\psi_a(f_t)\}_t \subset \mathbb{R}^q$ with $q = 2048$ in practice) is based on the deep residual network (ResNet-152) trained on ImageNet [26]. Besides the high performances reported in ImageNet classification [22], the particularity and the strength of this network resides in its skip connections which (i) reduce the sensitivity of the network to its architecture and (ii) reduce the effect of gradient collapse/explosion thereby making the optimization and fine-tuning of this network parameters (through stochastic gradient descent) effective and numerically more stable. The output of the motion stream (denoted as $\{\psi_m(f_t)\}_t$) is based either on 3D CNN [3] or 2D CNN [13]; the former is trained with normalized multi-frame optical flows² while the latter is trained with heatmaps colorized at the video-level.

²Normalized means that the values of the optical flow range between 0 and 255.

2.2. Deep hierarchical aggregation

Given a video \mathcal{V} , we introduce in this section an aggregation process that combines representations obtained at the frame-level of \mathcal{V} . A good aggregation design should tradeoff the global description of videos while capturing their details that distinguish possible fine-grained categories of actions. Hence, the design principle of our aggregation process is tree-structured and relies on a hierarchy of convolutional network representations. Without a loss of generality, we consider a binary hierarchy of L levels, where $\mathcal{N}_{k,\ell}$ stands for the set of frames that belong to the k -th node and the ℓ -th level. Top levels of this hierarchy provide *coarse* (long-term) video representations that capture global motion and appearance of actions while deep levels capture *fine* (and timely-resolute) details of these actions, such as “beginning”, “middle” and “late” aspects of actions, resulting into *coarse-to-fine* spatio-temporal representations.

Each node $\mathcal{N}_{k,\ell}$ is assigned an appearance representation, referred to as $\Psi_a^{k,\ell}(\mathcal{V})$; this representation is defined as $\Psi_a^{k,\ell}(\mathcal{V}) = \frac{1}{|\mathcal{N}_{k,\ell}|} \sum_{t \in \mathcal{N}_{k,\ell}} \psi_a(f_t)$. Given a set of action categories $\mathcal{C} = \{1, \dots, C\}$; we train multiple classifiers (denoted $\{g_c\}_{c \in \mathcal{C}}$) on top of the hierarchy of these representations. In practice, we use SVMs as classifiers whose kernels correspond to linear combinations of elementary kernels dedicated to $\{\mathcal{N}_{k,\ell}\}_{k,\ell}$. SVMs are suitable choices as they allow us to weight the impact of nodes in the hierarchy and put more emphasis on the most relevant granularity of the learned representations. Hence, depending on the granularity of action categories, SVMs will prefer top or deep layers of the hierarchy.

Considering a training set of videos $\{(\mathcal{V}_i, y_{ic})\}_i$ associated to an action category c , with $y_{ic} = +1$ if \mathcal{V}_i belongs to the category c and $y_{ic} = -1$ otherwise, the SVM associated to this action category c is given by $g_c(\mathcal{V}) = \sum_i \alpha_i^c y_{ic} \mathcal{K}(\mathcal{V}, \mathcal{V}_i) + b_c$, here b_c is a shift, $\{\alpha_i^c\}_i$ is a set of positive parameters and \mathcal{K} is a positive semi-definite kernel; details about the setting of \mathcal{K} , as a part of SVM training and hierarchical aggregation design of our temporal pyramid, are given in the subsequent section.

2.3. Coarse-to-fine hierarchical aggregation design

Let $\mathcal{N} = \cup_{k,\ell} \mathcal{N}_{k,\ell}$ be the union of all possible nodes (frame sets) in the hierarchy of *depth* up to L levels and *width* up to 2^L nodes; in this section, we introduce our hierarchical aggregation design that allows us to combine multiple representations in \mathcal{N} . Our method is based on learning a convex combination of representations, and finds the “optimal” weights of this combination while training multi-class SVMs.

In what follows, unless explicitly mentioned, we write $\Psi_a^{k,\ell}(\mathcal{V}_i)$ for short as $\Psi^{k,\ell}(\mathcal{V}_i)$. We consider $\{\Psi^{k,\ell}(\mathcal{V}_i)\}_{i=1}^n$ as a training set of representations and $y_i \in \{1, \dots, C\}$ as the label (or category) of $\Psi^{k,\ell}(\mathcal{V}_i)$ taken from a well defined ground-truth; in practice $C = 101$ (see experiments). Multi-

class SVMs use the mapping $\Psi^{k,\ell}(\mathcal{V})$ that takes a given video \mathcal{V} from an input space into its representation space and find the unknown label of \mathcal{V} as

$$\arg \max_{c \in \mathcal{C}} g_c^{k,\ell}(\mathcal{V}), \quad (1)$$

here $g_c^{k,\ell}(\mathcal{V}) = \langle w_c^{k,\ell}, \Psi^{k,\ell}(\mathcal{V}) \rangle + b_{k,\ell}$, with $w_c^{k,\ell}$, $b_{k,\ell}$ being respectively hyperplane normal and bias associated to a given category $c \in \mathcal{C}$ and node (k, ℓ) .

In order to combine different nodes in the hierarchy and hence design appropriate aggregation, we use multiple representation learning that generalizes the above SVM framework [18]. Its main idea consists in finding a kernel \mathcal{K} as a convex linear combination of positive semi-definite (p.s.d) elementary kernels $\{\mathcal{K}_{k,\ell}\}_{k,\ell}$ associated to $\{\mathcal{N}_{k,\ell}\}_{k,\ell}$. Thus, the kernel value between two videos $\mathcal{V}, \mathcal{V}'$ is defined as

$$\mathcal{K}(\mathcal{V}, \mathcal{V}') = \sum_{k=1}^L \sum_{\ell=1}^{2^k} \beta_{k,\ell} \mathcal{K}_{k,\ell}(\mathcal{V}, \mathcal{V}'), \quad (2)$$

here $\beta_{k,\ell} \geq 0$, $\sum_{k,\ell} \beta_{k,\ell} = 1$ and each kernel $\mathcal{K}_{k,\ell}$ operates using only the subset $\mathcal{N}_{k,\ell}$ (in practice, $\mathcal{K}_{k,\ell}(\mathcal{V}, \mathcal{V}') = \langle \Psi^{k,\ell}(\mathcal{V}), \Psi^{k,\ell}(\mathcal{V}') \rangle$). Resulting from the closure of the p.s.d of $\{\mathcal{K}_{k,\ell}\}_{k,\ell}$ w.r.t the sum, the final kernel \mathcal{K} will also be p.s.d. Hence, using a primal SVM formulation, we predict the unknown category of a given video \mathcal{V} as $\arg \max_{c \in \mathcal{C}} g_c(\mathcal{V})$, with $g_c(\mathcal{V}) = \sum_{k,\ell} \beta_{k,\ell} \langle w_c^{k,\ell}, \Psi^{k,\ell}(\mathcal{V}) \rangle + b_c$ and b_c , $\{w_c^{k,\ell}\}_{k,\ell}$ being respectively the bias and the hyperplane normals associated to a given class c for different nodes. We choose the parameters $\beta = \{\beta_{k,\ell}\}_{k,\ell}$, $b = \{b_c\}$ and $w = \{w_c^{k,\ell}\}_{k,\ell}$ by solving the following constrained minimization problem

$$\begin{aligned} \min_{\beta, w, b, \xi} \quad & \frac{1}{2} \sum_{k,\ell} \sum_c \beta_{k,\ell} \langle w_c^{k,\ell}, w_c^{k,\ell} \rangle + \sum_{j=1}^n \xi_j \\ \text{s.t.} \quad & \xi_j = \max_{c' \in \mathcal{C} \setminus c} l(g_c(\mathcal{V}_j) - g_{c'}(\mathcal{V}_j)), \end{aligned} \quad (3)$$

here $c \in \mathcal{C}$ is the actual label of \mathcal{V}_j , $\xi = \{\xi_j\}_j$ acts as a softmax and $l(\cdot)$ is a convex loss function. As this problem is not convex w.r.t the training parameters β, w, b, ξ taken jointly and convex when taken separately, an EM-like iterative optimization procedure can be used: first, β is fixed and the above problem is solved w.r.t w, b, ξ using quadratic programming, then w, b, ξ are fixed and the resulting problem is solved w.r.t β using linear programming. This iterative process stops when the values of all these parameters remain unchanged (from one iteration to another) or when it reaches a maximum number of iterations (see for instance [18, 7]).

3. EXPERIMENTS

In this section, we evaluate the performance of action classification using the challenging UCF101 database [19]. This

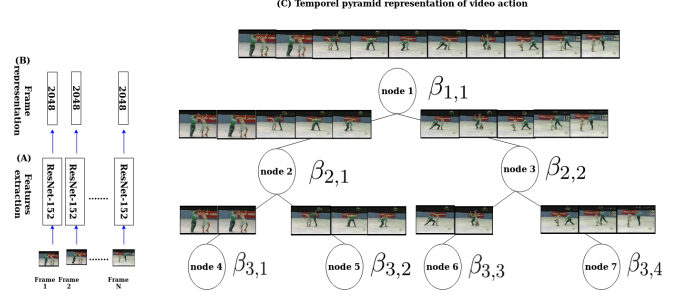


Fig. 1. This figure shows steps (A, B and C) of deep hierarchical aggregation that controls (via $\{\beta_{k,\ell}\}_{k,\ell}$) the granularity of the learned representations. **Better to zoom the PDF.**

dataset includes 13320 video shots taken from various actions belonging to 101 categories. These videos have diverse contents and were taken under extremely challenging and uncontrolled conditions, with many viewpoint changes (see examples of video frames in Fig. 2). Each video is processed in order to extract its underlying CNN representations at the frame-level, followed by their hierarchical aggregation at the video-level as discussed in section 2; Fig. 1 is an illustration of the whole video representation process.

3.1. Setting and evaluation protocol

The purpose of our evaluation is to show the performance of the hierarchical aggregation design of our temporal pyramid (TP) compared to different coarse and fine aggregations as well as other baselines. We also extend the comparison of action classification against reported results in the related work. We plugged our temporal pyramid into support vector classifiers in order to evaluate their performances. Again the targeted task is action classification (a.k.a recognition); given a video shot described with a temporal pyramid, the goal is to predict which action (class) is present into that shot. For this purpose, we trained a one-vs.-all SVM classifier for each class; we use the train-test split2 evaluation protocol (suggested in [19]) in order to compare the performance of our method against the related work under exactly the same conditions. We repeat this training and testing process through different classes and we take the average accuracy over all the classes of actions.

3.2. Performance and comparison

Baselines. We first show a comparison of action recognition performance, using our temporal pyramid, against two baselines: global average pooling and also spectrograms; the *former* produces a global representation that averages all the frame descriptions while the *latter* keeps all the frame representations and concatenate them prior to their classification. Note that these two comparative baselines are interesting as they correspond to two extreme cases of our hierarchy,

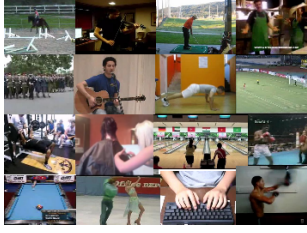


Fig. 2. Sample of classes from UCF-101 dataset. From top-left to bottom-right, classes are: riding horse, playing violin, golf swing, pizza tossing, military parade, playing guitar, pushups, soccer penalty, bench press, haircut, bowling, punch, billiard, ice-dancing, typing.

Setting	action recognition performance on UCF101
Temporal pyramid (level 1)	66.15%
Temporal pyramid (level 2)	66.74%
Temporal pyramid (level 3)	67.14%
Temporal pyramid (level 4)	67.41%
Temporal pyramid (level 5)	67.45%
Temporal pyramid (level 6)	67.47%
Temporal pyramid +Multiple Rep	68.58%
Spectrograms	64.41%

Table 1. Performances (on split 2 of UCF101) for level-wise and with multiple representation learning (referred to as “Multiple Rep”) as described in section 2.3. As already described, level 1 corresponds to the global average pooling.

namely the root and the leaf levels; in particular, the spectrogram (of a video \mathcal{V} with T frames) is obtained when the number of leaf nodes, in the temporal pyramid, is exactly equal to T . We also consider as baselines: level-wise representations of our temporal pyramid. Early observations, reported in Table. 1, show that our hierarchical representation design makes it possible select the best configuration (combination) of level representations in order to improve the classification accuracy; indeed, the results show a clear gain when using this design compared to all the other levels when taken separately as well as spectrograms. This gain results from the match between the granularity of the learned representations in the temporal pyramid and the actual granularities of action categories.

Comparison w.r.t the related work. We also compared the classification performances and the complementary aspect of our temporal pyramid design against related work including [13] and [3]. The method in [13] is based on colorized heatmaps, as a variant of the global average pooling baseline; the latter corresponds to timely-stamped and averaged frame-wise probability distributions of human keypoints. From the results in Table. 2, our hierarchical design, brings a substantial gain of at least 12 points w.r.t colorized heatmaps. The method in [3] is based on 3D CNNs including two-streams; one for motion and another one for appearance. While these two-streams are highly effective, their combination with our temporal pyramid, through a simple late fusion, brings a

noticeable gain. We also observe the same behavior on all the combinations – of this two stream CNN with the other baselines; despite the fact that bridging the last few percentage gap is very challenging, for each setting our temporal pyramid succeeds in improving the performances.

Method	action recognition performances
col. heatM [13]	64.38%
col. heatM [13] + TP	77.34%
Spect	64.41%
Spect + TP	68.40%
Spect + col. heatM [13]	66.87%
Spect + col. heatM [13] + TP	74.65%
3D 2-stream (motion) [3]	96.41%
3D 2-stream (appearance) [3]	95.60%
3D 2-stream (combined) [3]	97.94%
3D 2-stream (motion) [3] + TP	97.50%
3D 2-stream (appearance) [3] + TP	95.77%
3D 2-stream (combined) [3] + TP	97.94%
3D 2-stream (motion) [3] + col. heatM [13]	94.89%
3D 2-stream (appearance) [3] + col. heatM [13]	94.32%
3D 2-stream (combined) [3] + col. heatM [13]	97.02%
3D 2-stream (motion) [3] + col. heatM [13] + TP	95.70%
3D 2-stream (appearance) [3] + col. heatM [13] + TP	94.60%
3D 2-stream (combined) [3] + col. heatM [13] + TP	97.56%
3D 2-stream (motion) [3] + spect	95.64%
3D 2-stream (appearance) [3] + spect	94.72%
3D 2-stream (combined) [3] + spect	97.70%
3D 2-stream (motion) [3] + spect + TP	95.77%
3D 2-stream (appearance) [3] + spect + TP	94.95%
3D 2-stream (combined) [3] + spect + TP	97.74%
3D 2-stream (motion) [3] + col. heatM [13] + spect	95.12%
3D 2-stream (appearance) [3] + col. heatM [13] + spect	94.70%
3D 2-stream (combined) [3] + col. heatM [13] + spect	97.32%
3D 2-stream (motion) [3] + col. heatM [13] + spect + TP	96.35%
3D 2-stream (appearance) [3] + col. heatM [13] + spect + TP	95.10%
3D 2-stream (combined) [3] + col. heatM [13] + spect + TP	97.51%

Table 2. Comparison w.r.t the related work (on split2 of UCF101 dataset); in this table: “heatM” stands for colorized heatmaps, “spect” for spectrograms and TP for temporal pyramid + multiple representation. We observe a clear gain (highlighted in bold) when TP is used and combined w.r.t the related work.

4. CONCLUSION

We introduced in this paper an action recognition method based on convolutional neural networks and a novel hierarchical aggregation design. The latter defines pooling operations at different granularities and makes it possible to fit the actual granularity of action categories resulting into a clear gain in performance compared to global average pooling and also spectrograms. Our method is based on solving a constrained minimization problem whose solution corresponds to the level-wise weight distributions which also maximize performances. Comparison, using the challenging UCF101 dataset, shows the validity and the complementary aspect of our method with respect to the related work. As a future work we are currently investigating the application of our hierarchical aggregation to activity recognition, on longer duration video datasets, and this requires deeper temporal pyramids.

5. REFERENCES

- [1] L. Wang, H. Sahbi. Directed Acyclic Graph Kernels for Action Recognition. Proceedings of the IEEE International Conference on Computer Vision. 2013.
- [2] X. Wang, L. Gao, J. Song, H. Shen. Beyond frame-level CNN: saliency-aware 3-D CNN with LSTM for video action recognition. IEEE Signal Processing Letters, 24(4), 510-514, 2017
- [3] J. Carreira, A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. IEEE CVPR, 2017
- [4] H. Pirsiavash, D. Ramanan. Detecting Activities of Daily Living in First-person Camera Views. IEEE Conference on Computer Vision and Pattern Recognition, 2012
- [5] L. Chen, L. Duan, and D. Xu, Event Recognition in Videos by Learning From Heterogeneous Web Sources, in IEEE CVPR, 2013, pp. 2666-2673
- [6] J. Liu, A. Shahroudy, D. Xu, G. Wang. Spatio-temporal LSTM with trust gates for 3D human action recognition. In European Conference on Computer Vision (pp. 816-833). Springer, 2016.
- [7] M. Jiu, H. Sahbi. Nonlinear deep kernel learning for image annotation. IEEE Transactions on Image Processing, volume 26, number 4, 1820-1832, 2017.
- [8] D. Xu, S-F. Chang. Visual Event Recognition in News Video using Kernel Methods with Multi-Level Temporal Alignment. In IEEE CVPR, 2007
- [9] H. Wang, C. Yuan, W. Hu, C. Sun. Supervised class-specific dictionary learning for sparse modeling in action recognition. Pattern Recognition, 2012
- [10] C. Schuldt, I. Laptev, B. Caputo. Recognizing human actions: a local SVM approach. In ICPR, 2004
- [11] T. Postadjian, A. Le Bris, H. Sahbi, C. Mallet. Investigating the potential of deep neural networks for large-scale classification of very high resolution satellite images. ISPRS Annals 4, 183-190, 2017.
- [12] C. Feichtenhofer, A. Pinz, R-P. Wildes. Spatiotemporal Multiplier Networks for Video Action Recognition. IEEE CVPR, 2017
- [13] V. Choutas, P. Weinzaepfel, J. Revaud, C. Schmid. PoTion: Pose MoTion Representation for Action Recognition. IEEE CVPR, 2018
- [14] H. Sahbi, X. Li. Context-based support vector machines for interconnected image annotation. Asian Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010.
- [15] 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, ECCV, 2016
- [16] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. ArXiv preprint arXiv:1511.04119, 2015.
- [17] K. Grauman, T. Darrell. The Pyramid Match Kernel: Efficient Learning with Sets of Features. Journal of Machine Learning Research, 2007
- [18] M. Gönen, E. Alpaydın. Multiple Kernel Learning Algorithms. Journal of Machine Learning Research, 2011
- [19] K. Soomro, A-R. Zamir and M. Shah, UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild, CRCV-TR-12-01, November, 2012.
- [20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, A. Zisserman. The Kinetics Human Action Video Dataset. ArXiv:1705.06950, May 2017
- [21] C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning Kernels based on Centered Alignment. JMLR, vol 13, pp. 795-828, Mar. 2012
- [22] B. Zoph, V. Vasudevan, J. Shlens, Q-V. Le. Learning Transferable Architectures for Scalable Image Recognition. IEEE CVPR, 2018
- [23] T-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C-L. Zitnick, P. Dollár. Microsoft COCO: Common Objects in Context. IEEE CVPR, 2014
- [24] L. Wang, H. Sahbi. Bags-of-Daglets for Action Recognition. IEEE International Conference on Image Processing (ICIP), 2014.
- [25] L. Wang, H. Sahbi. Nonlinear Cross-View Sample Enrichment for Action Recognition. European Conference on Computer Vision. Springer, 2014.
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. IEEE CVPR, 2009.
- [27] P. Scovanner, S. Ali, M. Shah. A 3-dimensional sift descriptor and its application to action recognition. Proceedings of the 15th ACM international conference on Multimedia. ACM, 2007.
- [28] F. Yuan, G-S. Xia, H. Sahbi, V. Prinet. Mid-level Features and Spatio-Temporal Context for Activity Recognition. Pattern Recognition. volume 45, number 12, 4182-4191, 2012