ACTION RECOGNITION USING COMBINED LOCAL FEATURES

Ivo Reznicek
Faculty of Information Technology, Brno University of Technology
Bozetechova 1/2, 602 00 Brno, Czech Republic

Pavel Zemcik
Faculty of Information Technology, Brno University of Technology
Bozetechova 1/2, 602 00 Brno, Czech Republic

ABSTRACT

This paper presents a new algorithm for recognition of actions based on local space-time features. The algorithm resulted from intensive research of classification and feature extraction and it is an extension of the earlier algorithms. The most important achievement is that it is shown that carefully selected combination of space-time features leads to a greater precision of recognition on some events compared with the state-of-the-art algorithms while it is comparable on all other events. The paper describes the algorithm, its main features and improvements, demonstrates the results achieved, and draws conclusions.

KEYWORDS

Action recognition, SVM, combination of features, space-time features.

1. INTRODUCTION

Object detection, video search, and action detection have become very popular and widely used in the past decade. These tasks can be successfully used in the applications, such as video surveillance and video search and retrieval. All these tasks frequently exploit very similar processing chain consisting of local features extraction [Laptev, I. and Lindeberg, T. 2003; Dollar, P. et al., 2005; Willems, G. et al., 2008; Klaser, A. et al., 2008; Scovanner, P. et al., 2007], creation of global descriptor from the local descriptors, and classification, where the global descriptor is usually related to whole image, whole video sequence, or some portion of video sequence.

The above tasks rely on local features as they quite well describe local information about interest points in spatial domain, in case of images, and in space-time domain in case of the video sequences. Various local descriptors can be combined into global feature vector using bag-of-words [Csurka, G. et al., 2004] representation which for any image or video sequence has a nice feature of resulting vectors having the same dimensionality and thus being usable as an input for classifier. Approaches based on such representation have proven to be capable of achieving state-of-the-art results [Wang, H. et al., 2011; Wang, H. et al., 2009; Le, Q. V. et al., 2011] for action recognition tasks.

The space-time detectors were first developed and introduced by Laptev in [Laptev, I. and Lindeberg, T. 2003]; the space-time features extend the standard Harris corner detector into space-time domain. Many of the subsequently developed detectors are based on Gabor filters [Dollar, P. et al., 2005] or on the determinant of the Hessian matrix [Willems, G. et al., 2008]. Feature descriptors that are used for description of the interest point local neighbourhood range from higher order derivatives, gradient information, optical flow, and brightness information [Dollar, P. et al., 2005; Laptev, I. et al., 2008; Schuldt, C. et al., 2004] to extensions of image descriptors, such as HOG3D [Klaser, A. et al., 2008], SURF [Willems, G. et., al 2008], or 3D-SIFT [Scovanner, P. et al., 2007].

Video processing and video processing evaluation methods almost always rely on datasets. Datasets being recently and widely used for this purpose include KTH [Schuldt, C. et al., 2004], Weizmann [Gorelick, L. et al., 2005], UCF sports [Rodriguez, M. D. et al 2008], IXMAS [Weinland, D. et al., 2007], and Hollywood2 actions [Marszalek, M. et al., 2009]. The most challenging dataset is the Hollywood2 actions; it contains set of videos of a standard resolution taken from Hollywood movies with 12 real world actions annotated; the best reported results [Wang, H. et al., 2011; Wang, H. et al., 2009; Le, Q. V. et al., 2011] are currently 50%-60% (using mean average precision measure). The mean average precision metric, in this context, is defined as a mean value of all the precision recall curve surfaces for all the classes of interest.

1.1 Related Work

While in the last decade a great number of papers with various concepts for action recognition have been published, a significant part of those approaches are based on feature extraction, fixed-sized representation conversion and classifier creation. The most interesting examples of approaches are listed below.

Wang et al. evaluated in [Wang, H. et al., 2009] several combinations of feature extractors and feature descriptors, using all the important datasets available at the time. In this approach, video sequences are represented by bag-of-words and the vocabulary is created using the *k*-means algorithm. For classification purposes, the non-linear support vector machine with χ^2 kernel is used. The results are reported and measured using mean average precision.

Wang et al. [Wang, H. et al., 2011] proposed in his further work a new way of extracting the time-space interest points, called Dense trajectories. The Dense trajectories extractor is based on the assumption that search for the extrema across all three dimensions is not efficient because of the different characteristics of the space domain and the temporal domain. With this approach, the points are detected in the spatial domain and then tracked across the temporal domain. After the point trajectory has been found, the descriptor is calculated around this trajectory, while the length of all trajectories is equal. A number of descriptors were examined with this extractor. The HOG and HOF descriptors (the same as in the STIP extractor [Laptev, I. and Lindeberg, T. 2003]), trajectory descriptor, and MBH descriptor were used.

All the above feature descriptors are used separately; they are transformed into the bag-of-words [Csurka, G. et al., 2004] representation and used for training the multichannel non-linear SVM with χ^2 kernel similarly as in [Ullah, M. M. et al., 2010]. The accuracy of the algorithm is evaluated on today's datasets and it is compared with other state-of-the-art papers using the mean average precision measure.

Ullah et al. [Ullah, M. M. et al., 2010] has presented some extension of the standard bag-of-words approach where the video is segmented semantically into meaningful regions (spatially and temporally) and the bag-of-words histograms are computed separately for each region. This work also introduces a number of experiments and the results are included in our work in the comparison of results.

Le Q. V. [Le, Q. V. et al., 2011] has presented a method for the learning of features from spatio-temporal data using the independent subspace analysis. A number of experiments are included in our work in the comparison of results.

To the best of our knowledge, however, no paper has been published where all earlier known feature-like systems are combined into one solution and where the best combination is evaluated for each separate purpose.

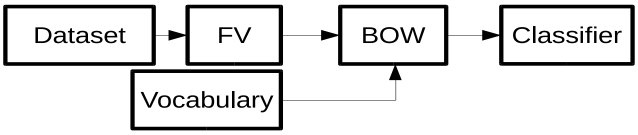
1.2 Dataset

Marszalek et al. in [Marszalek, M. et al., 2009] proposed a dataset with twelve action classes and ten scene classes annotated, which was acquired from 69 Hollywood movies. The dataset is built from movies containing human actions and processed using script documents and subtitle files which are publicly available for those movies. The script documents contain scene captions, dialogs, and scene descriptions; however, they are usually not quite precisely synchronized with the video. The subtitles have video synchronisation so they are matched to the movie scripts and this fact can be used to improve video clip segmentation. By analysing the content of movie scripts, the twelve most frequent action classes and their video clip segments are obtained. These segments are split into test and training subsets such that the two subsets do not share segments from the same movies. Two training parts of the dataset exist; the *automatic* part, it is generated using the above-mentioned procedure while the *clean* part is manually corrected using visual information from the video. The test part is manually corrected in the same way as the clean training part of the dataset. In both cases, the correction is performed in order to eliminate "noise" from the dataset and thus to create better classifiers.

2. BASE RECOGNITION ALGORITHM

The procedure is based on the extraction of feature vectors, their transformation, and creation of classifiers; the base processing pipeline is shown in Figure 1. For the videos being processed, the local feature vectors “FV” are extracted and then transformed into the bag-of-words “BOW” representation using the visual vocabulary. The bag-of-words vectors are then combined and used as an input to the classifier creation process.

Figure1. The base recognition pipeline: *videos from dataset are converted to feature vectors, which are transformed into fixed-size representation, which in turn is used for classifier creation.*



The input of the classifier engine is used for classification. The accuracy of the classification is evaluated in the processing phase using another subset of the dataset, the testing set. The outputs of the classifier are then compared with the annotations and the results are evaluated.

2.1 Feature extraction

The purpose of the local feature extractors is to search for local extrema across the space and time domain of the input video and when the extremum is detected, the neighbourhood pixels across the space and time domain are used to obtain the feature vector describing such extrema. Alternatively, in the case of dense sampling, the extrema are not searched for and uniform sampling of space is used instead to obtain the feature vectors. In such a case, no search is required but a larger number of features need to be evaluated.

The following feature extractors were presented for action recognition: STIP, Cuboids, HesSTIP and Dense Trajectories their fundamentals will be presented below.

In the STIP extractor, the key points are searched for using the extended Harris corner detector [Harris, C. and Stephens, M. 1988]. Subsequently, for each of the detected points, the space-time patch is extracted and the HOGHOF [Laptev, I. and Lindeberg, T. 2003] descriptor is computed. The descriptor consists of the histogram of gradient descriptor and the histogram of optical flow descriptor which are simply concatenated. HOG captures the static appearance information while HOF captures the local motion information.

The Cuboids extractor is based on the 2D Gaussian smoothing kernel, which is applied spatially, and the quadrature pair of 1D Gabor filters, which is applied temporally. The non-maxima suppression and thresholding are performed and as a result of this process, the key point locations are detected. The cuboids descriptor is simply computed by concatenating the gradients obtained for each pixel in each dimension of the processed patch. Another type of cuboids extractor is also known, where the key point search procedure is replaced by the Harris corner detector.

In the HesSTIP extractor, the key points are detected using the space-time extension of the Hessian saliency measure (which is usually used for blob detection in images). The detector measures the saliency using the determinant of the 3D Hessian matrix. The descriptor vector is obtained as follows. The space-time patch is divided into cells. For each cell, the vector of weighted sums of uniformly sampled responses of the Haar-wavelets along the three axes is computed. Vectors from all cells are then concatenated.

The dense trajectories extractor is depicted in Section 1.1; to describe the detected trajectories the HOG, HOF, MBx and MBy descriptors were used. Generally, every feature extractor generates a set of feature vectors, all of which have the same dimension from a single video file.

2.2 Visual vocabulary and bag-of-words

The visual vocabulary is created as a model for representation of the low-level feature space and it is formed by a set P of representatives P_i (points) in n-dimensional space. The size of the vocabulary has to be adjusted to a suitable value so that the representation of the space is compact and accurate enough at the same time. If the size is too large, nearly all low-level features become representatives of the visual vocabulary. If the size were too small, very large clusters would exist and the discriminative power of the whole solution might be adversely affected.

K-means square-error partitioning method [Duda, R. O. et al., 2000] can be used for such purpose. This algorithm iteratively processes data such that it assigns feature points to their closest cluster centres and recalculates the cluster centres. The *k*-means algorithm converges only to local optima of the squared distortion and does not determine the *k* parameter. It can be parametrized through specifying the number of iterations and the number of output clusters.

The bag-of-words [Csurka, G. et al., 2004] can represent the video sequence or its part using one feature vector with the same dimension, irrespective of the number of local space-time vectors or the video shot length; the bag-of-words representation can be (in its simple form) constructed in the following way. The input of this process is the set *S* of local feature vectors $s \in S$ and a vocabulary while the output is a histogram of the occurrences of matched input vectors. For each input vector, exactly one bin in the output histogram is incremented. This simple form of assignment is sometimes called the *hard assignment* and also has some disadvantages. The main disadvantage is that only slightly different input local feature vectors may be accumulated into totally different output histogram bins (the nearest code words are different); this may cause total dissimilarity of two similar input vectors.

The above issue is addressed in the *soft assignment* approach; the soft assignment is performed as follows. A small group of the clusters very close to the vector being processed is retrieved instead of a single cluster; all the clusters from such a group are assigned a weight corresponding to their closeness to the vector; finally each of the corresponding output histogram bins is incremented by the weight of the appropriate clusters.

The most frequently used method of weight computing is through exponential function of the distance to the cluster centre $w_i(a) = \exp\left(-\frac{(d(a, p_i))^2}{2\sigma^2}\right)$, where *d* is Euclidean distance from the cluster centre to the vector while σ is a parameter and controls the width of the function. This function needs to be evaluated for each of the clusters in the group. Finally, *soft assignment* parameters correspond to the number of the very close vectors to be considered and the σ which controls the shape of soft-weighting function.

2.3 Classifier

The classifier can be described as a blackbox unit which has two modes of operation: the training phase, where the model for certain input labelled data is created, and the classification phase, where the classifier is able to decide how the tested data should be labelled. Generally, inside this box, many algorithms can be used (SVM [Zhang et al. 2007], neural networks [Kriesel D. 2007], Bayesian classifier [Friedman N. et al., 1997], etc.), the common property is that classifier creation is dependent on the set of parameters and its quality is based on these parameters. The input of the classifier is typically an input vector typical of an object, the output is a vector of class likelihood.

For action recognition and image-content recognition the most popular classifier type is the SVM (Support Vector Machine) with various kernel functions (for example, linear kernel, rbf kernel or χ^2 kernel).

3. OPTIMAL COMBINATION OF FEATURES

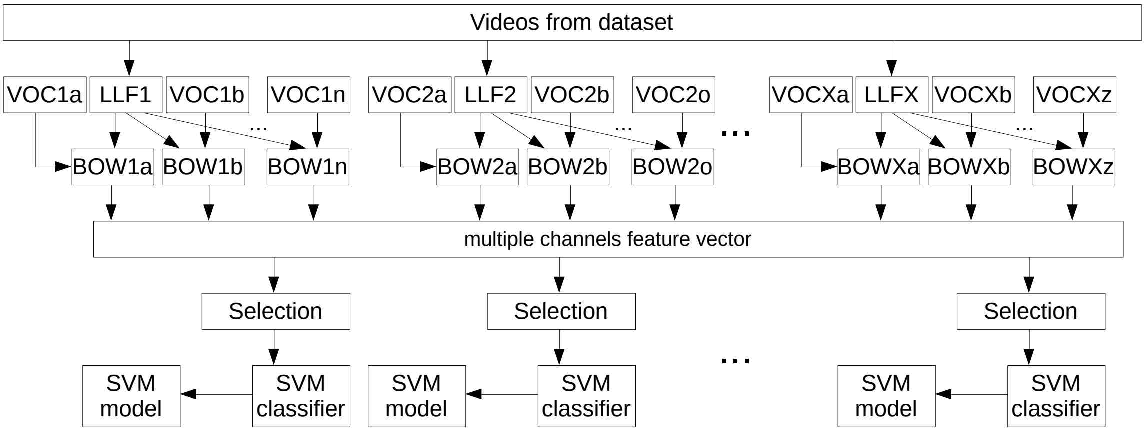
The above presented algorithm can be extended through a combination of different features forming the feature vector. It will be shown that a proper combination of the features can lead to an improvement of the performance beyond the state of the art when selected individually for some of the event classes.

The algorithm is depicted in Figure 2; a larger number of feature extractors are used for processing. For each feature extractor a larger number of visual vocabularies are created and used for the creation of all possible bag-of-words representations. Everything is concatenated into a multiple channels feature vector. The *selection* unit combines several input channels and it is passed to the multikernel SVM classifier creation process. This operation is repeated several times.

The classifiers created through the above mentioned processing need to be explored and the best one will be further used. The classifiers are evaluated against a chosen metric. The selected one is evaluated using the testing dataset and is measured using, for example, the average precision metric.

The whole processing needs three types of dataset: the training one, which is used for the creation of classifiers; the validation one, which is helpful in best solution selection, and the testing one, which is used for measuring the whole-system accuracy.

Figure 2. The algorithm block diagram; The *LLFx* boxes depict the feature extractors, the *VOCx* represent the vocabularies constructed from related feature extractors, the *BOWx* boxes depict the bag-of-words units, the multiple channels feature vector is constructed by concatenation of all vectors, but the positions of all subparts need to be kept.



The algorithm uses the non-linear support vector machine [Zhang et al., 2007] with multichannel Gaussian kernel [Zhang et al., 2007; Wang et al., 2011]. The kernel shall be defined as:

$$K(A,B) = exp(-\sum_{c\in C} \frac{1}{A_c} D_c(A,B)) \tag{1}$$

where A_c is the scaling parameter, which is determined as a mean value of mutual distances D_c between all the training samples for the channel c, $D_c(A, B)$ is the χ^2 distance between two bag-of-words, and A and B are the input vectors of the form:

$$A_i = (\underbrace{a_1 \ldots a_{n1}}_{channel\ \langle 1,n_1\rangle} , \underbrace{a_{n_1+1} \ldots a_{n2}}_{channel\ \langle n_1+1,n_2\rangle} , \ldots , \underbrace{a_{n_i-1} \ldots a_{n_i}}_{channel\ \langle n_i-1,n_i\rangle}) \tag{2}$$

The set of channels C can be defined as:

$$C = \{\langle 1,n_1\rangle, \langle n_1+1,n_2\rangle, \ldots, \langle n_i-1,n_i\rangle\} \tag{3}$$

The bag-of-words distance $D_c(A, B)$ may be obtained as:

$$D_c(A,B) = \frac{1}{2} \sum_{n\in c} \frac{(a_n - b_n)^2}{a_n + b_n} \tag{4}$$

The best ratio of input channels $\{c_k, c_l, \ldots, c_z\} \in C$ for a given training set is estimated using the coordinate descent method. The set of input channels needs to be specified outside of the training process.

Besides this SVM, the building procedure requires the number of input parameters that affect the classifier accuracy; these parameters are automatically evaluated using the cross-validation approach [Hsu, C. W. et al., 2003]. The classifier creation process may be apprehended in the whole procedure as a black-box unit where only the set of input channels is specified, and for a given input the best performing classifier is created automatically.

The number of channels used may induce a very large space which needs to be searched. The number of possible combinations of this space can be computed as a sum of the sequence which may be defined as follows: $count=\binom{|C|}{1}+\binom{|C|}{2}+\ldots+\binom{|C|}{|C|}=2^{|C|}-1$, where |C| represents the number of channels.

Currently, we are able to achieve a good performance by an ad-hoc (manual or blind) specification of the input channel combination (it will be further shown in Chapter 4), the algorithm for automatic channels selection is now under development and was not used in this paper.

4. EXPERIMENTAL RESULTS

The main achievement of the presented work is the confirmation of the hypothesis that a suitable combination of different features for action recognition does improve the accuracy of the whole processing chain; this idea has been explored and evaluated using one of the most challenging datasets [Marszalek, M. et al., 2009] available today. The following twelve action classes were evaluated, namely: *answering the phone, driving car, eating, fighting, getting out of the car, hand shaking, hugging, kissing, running, sitting down, sitting up and standing up*.

In our experiments, the clean part of the training dataset was used for the classifier training procedure (823 samples). The automatic part of the training dataset was re-annotated and used for validation purposes (810 samples). The original testing dataset (884 samples) was used for measuring the solution using average precision for every class, the over-all classes mean average precision is reported as well.

The following feature extractors were used in the experiment, the associated list of descriptors is given in parentheses, every combination extractor and descriptor was used as a standalone features set plus all the dense trajectories descriptors were concatenated and used as well:

- Dense Trajectories (Trajectory, HOG, HOF, MBH),
- HesSTIP (ESURF)
- Cuboids (Cuboids)
- STIP (HOGHOF)

Some vocabularies were created using the *k*-means algorithm with 12 iterations; this number represents a trade-off between the processing duration and the output vocabulary achievement. To create these vocabularies, ca. 2 million local low-level features were used and were extracted from all training videos of the dataset. Vocabulary sizes were set to 1000, 6000 and 8000, all possible combinations, feature extractors and. vocabularies sizes were used.

Table 1. Results of average precision of the four best performing experiments on the validation dataset.

Action	1	2	3	4	BEST	Selected classifier
answering the phone	0.379	0.299	0.322	0.423	0.423	4
driving car	0.571	0.62	0.554	0.578	0.62	2
Eating	0.327	0.355	0.295	0.37	0.37	4
getting out of the car	0.377	0.237	0.304	0.273	0.377	1
Running	0.629	0.683	0.736	0.702	0.736	3
sitting down	0.487	0.559	0.511	0.574	0.574	4
sitting up	0.286	0.204	0.385	0.331	0.385	3
standing up	0.486	0.55	0.394	0.527	0.55	2
Fighting	0.625	0.594	0.55	0.561	0.625	1
hand shaking	0.493	0.541	0.439	0.594	0.594	4
Hugging	0.355	0.339	0.417	0.369	0.417	3
Kissing	0.531	0.630	0.594	0.609	0.630	2
Mean average precision	0.462	0.468	0.458	0.478	0.484	

The soft-assignment approach was used for the bag-of-words representation with the following parameters: $\sigma = 1$, the number of searched closest vectors was 16; these values were evaluated in [Reznicek, I. and Zemcik, P. 2011] and are suitable for bag-of-words creation from space-time low-level features.

Bag-of-words representations generated from all the possible combinations feature extractors and vocabularies become the input channels to the SVM creation process. SVMs were created as described in chapter 3. The dataset used induces the multiclass classification. The one-against-all approach was used and no relation between classes has been considered.

The number of input channels in our experiment is 24 and the total number of possibilities is then: $\binom{24}{1}+\binom{24}{2}+\ldots+\binom{24}{24}\simeq 16{,}7.10^{6}$.

We have searched about 0.1% of the desired space in a semi-automatic way and the four most interesting results (combinations) for the validation part of the dataset are presented in Table 1. The average precision is reported for each class and the mean average precision is reported for the whole validation dataset.

Table 2 represents the results *for our class-based best input channel combinations* (as shown in Table 1) achieved using the test part of the Hollywood2 dataset in the column *OUR* and they are compared to the three other authors’ papers [Wang et al., 2011; Le et al., 2011; Ullah et al., 2010] which represent today's state-of-the-art for Hollywood2 dataset.

Our combination-based solution outperformed all other state-of-the-art methods in four classes, namely *driving car, running, sitting down, standing up*; in the other cases, the solution does not reach the state-of-the-art performance but it is still comparable.

As the performance of classifiers based on the combination of features is known only after the validation phase, the best solution based on the combination of features or another approach can be chosen individually for each type of action; therefore, improvement in four out of twelve actions leads to the best-known classification mechanism, also shown in Table 2.

Table 2. Results of average precision of the selected classifiers, compared with the state-of-the-art.

Action	**OUR**	[Wang et al 2011]	[Le et al. 2011]	[Ullah et al 2010]	**BEST KNOWN**
answering the phone	0.259	**0.326**	0.299	0.248	**0.326**
driving car	**0.91**	0.880	0.852	0.881	**0.91**
eating	0.491	**0.652**	0.597	0.614	**0.491**
getting out of the car	0.408	**0.527**	0.454	0.474	**0.527**
running	**0.834**	0.821	0.757	0.743	**0.834**
sitting down	**0.655**	0.625	0.594	0.613	**0.655**
sitting up	0.206	0.200	**0.257**	0.255	**0.257**
standing up	**0.663**	0.652	0.647	0.604	**0.663**
fighting	0.723	**0.814**	0.772	0.765	**0.814**
hand shaking	0.286	0.296	0.203	**0.384**	**0.384**
hugging	0.364	**0.542**	0.382	0.446	**0.542**
kissing	0.601	**0.658**	0.579	0.615	**0.658**
Mean average precision	0.533	**0.583**	0.533	0.553	**0.589**

5. CONCLUSIONS AND FUTURE WORK

The present work focuses on the recognition of gestures and actions in video sequences. The purpose of the work was to demonstrate that recognition of actions can be improved through combinations of different space-time features.

While a suitable general method for selecting of features to be combined is not yet known, our experiments demonstrate the feasibility of the idea because some of the feature combinations outperform the current state-of-the-art for four of twelve actions classes and it nearly matches the state-of-the-art for most of the remaining classes.

The implementation of the action recognition system was performed using the Hollywood2 dataset with a measurable improvement over the state of the art. The procedure of creating of a classifier based on a combination of features was also shown.

Future work includes research into algorithms for automatic selection of features, research into methods of feature fusion, and also general action recognition methods.

ACKNOWLEDGEMENTS

This work has been made possible thanks to the funding through the CRAFTERS ConstRaint and Application driven Framework for Tailoring Embedded Real-time Systems, Artemis JU, project 7H12006 and V3C - Visual Computing Competence Center, Technology Agency of the Czech Republic, TAČR, TE01020415V3C.

REFERENCES

Csurka, G. et al, 2004. Visual categorization with bags of keypoints. *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

Dalal, N. et al, 2006. Human detection using oriented histograms of flow and appearance. *In ECCV*, pages 428–441.

Dollar, P. et al, 2005. Behavior recognition via sparse spatio-temporal features. *In VS-PETS*, pages 65–72.

Duda, R. O. et al, 2000. *Pattern classification.* Wiley, New York; Chichester.

Friedman, N. et al, 1997. Bayesian network classifiers, *Machine Learning*, 29:2/3.

Gorelick, L. et al, 2005. Actions as space-time shapes. *In ICCV*, pages 1395–1402.

Harris, C. and Stephens, M., 1988. A combined corner and edge detector. *In Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hsu, C. W. et al, 2003. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.

Klaser, A. et al, 2008. A spatio-temporal descriptor based on 3d-gradients. *In BMVC.*

Kriesel, D., 2007. *A Brief Introduction to Neural Networks*, available at http://www.dkriesel.com

Laptev, I. and Lindeberg, T., 2003. Space-time interest points. *In ICCV*, pages 432–439. IEEE Computer Society.

Laptev, I. et al, 2008. Learning realistic human actions from movies. *In CVPR*. IEEE Computer Society.

Le, Q. V. et al, 2011. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. *In CVPR*, pages 3361–3368. IEEE.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *In IJCV*, 60:91–110.

Marszalek, M. et al, 2009. Actions in context. *In CVPR*, pages 2929–2936. IEEE.

Reznicek, I. and Zemcik, P., 2011. On-line human action detection using space-time interest points. *In Zbornik prispevkov prezentovanch na konferencii ITAT*, september 2011, pages 39–45. Faculty of Math-ematics and Physics.

Rodriguez, M. D. et al, 2008. Action mach a spatio-temporal maximum average correlation height filter for action recognition. *In CVPR*. IEEE Computer Society.

Schuldt, C. et al, 2004. Recognizing human actions: A local svm approach. *In ICPR* (3), pages 32–36.

Scovanner, P. et al, 2007. A 3-dimensional sift descriptor and its application to action recognition. In ACM Multimedia, pages 357–360. ACM.

Ullah, M. M. et al, 2010. Improving bag-of-features action recognition with non-local cues. *In BMVC*, pages 1–11.

Wang, H. et al, 2011. Action recognition by dense trajectories. *In CVPR*, pages 3169–3176. IEEE.

Wang, H. et al, 2009. Evaluation of local spatio-temporal features for action recognition. *In BMVC*.

Weinland, D. et al, 2007. Action recognition from arbitrary views using 3d exemplars. In ICCV, pages 1–7. IEEE

Willems, G. et al, 2008. An efficient dense and scale-invariant spatio-temporal interest point detector. *In ECCV* (2), volume 5303 of Lecture Notes in Computer Science, pages 650–663. Springer.

Zhang, J. et al, 2007. Local features and kernels for classification of texture and object categories: a comprehensive study. *In International Journal of Computer Vision*, 73:2007.