Human Action Recognition for Real-time Applications

Ivo Reznicek and Pavel Zemcik
Faculty of Information Technology, Brno University of Technology, Bozetechova 1/2, Brno, Czech Republic

Keywords: Space-time Interest Points, Action Recognition, Real-time Processing, SVM

Abstract: Action recognition in video is an important part of many applications. While the performance of action recognition has been intensively investigated, not much research so far has been done in the understanding of how long a sequence of video frames is needed to correctly recognize certain actions. This paper presents a new method of measurement of the length of the video sequence necessary to recognize the actions based on space-time feature points. Such length is the key information necessary to successfully recognize the actions in real-time or performance critical applications. The action recognition used in the presented approach is the state-of-the-art one; vocabulary, bag of words and SVM processing. The proposed methods is experimentally evaluated on human action recognition dataset.

1 INTRODUCTION

Contemporary video technology produces a large number of video sequences which need to be stored, processed, and searched for various purposes. In this context, action recognition becomes increasingly significant with a growing number of cameras everywhere. These cameras provide video streams which may be used to secure human lives, properties, and hopefully to make our lives nicer and easier. One of the main application of action detection is the detection of human behavior or actions.

In order to detect actions, the space-time interest point features are often used. Some authors such as Wang (Wang et al., 2009; Wang et al., 2011; Reznicek and Zemcik, 2013) have shown that the best state-of-the-art performance is achieved by combining of several space-time feature points extractors. Unfortunately, real-time performance of video processing using these algorithms is very difficult to achieve using today's computer technology. Such tasks can only be done by using high performance computer clusters, where the processing load is distributed among several CPUs. Anyway, apart from the precision of the action recognition, it is interesting to learn how long video sequences are needed for successful recognition of the actions of interest. The reason is that the minimum necessary length of the video sequence determines important features of the applications. Such features include, for example, a delay between the start of specific human action and the computer system's response in human machine interfaces, computational performance in applications that search some recorded video sequences for certain actions, etc. Moreover, the high computational complexity of the space-time features based action recognition algorithms will shortly become less important, especially thanks to the quickly increasing computational performance observed in computer technology today.

The action detection algorithms based on space-time interest points may be applied in a variety of tasks. They can be used for on-line detection of human behavior for surveillance systems, as a support for video system operators as well as for searching for specific action or human behavior in video databases. Many of the applications of action recognition algorithms would benefit from real-time processing and this paper should help to reach such a performance.

In this paper, the length of the video necessary for action recognition is investigated along with the accuracy of the recognition. We have investigated the dependency between the number of video frames in a video sequence containing certain actions and the accuracy of the action recognition with an assumption that the accuracy of action recognition will grow with the length of the video sequence until it achieves state-of-the-art accuracy. It should be noted that in contemporary state-of-the-art systems, the length of the video sequences is not restricted. Using the results of this research, systems capable of well defined delay in action recognition as well as well defined ac-

Reznicek I. and Zemcik P..
Human Action Recognition for Real-time Applications.
DOI: 10.5220/0004826606460653
In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2014), pages 646-653
ISBN: 978-989-758-018-5

Figure 1: Examples of frames from the actions contained Hollywood2 dataset (Marszalek et al., 2009); *hugging, kissing and answering the phone* actions.

curacy can be built.

In this work, the proposed approach is evaluated for human action recognition on the "Hollywood2" dataset (Marszalek et al., 2009).

While alternative possibilities of action recognition, and specifically human action recognition, exist, the video (video only) systems remain the most usable ones because of the availability of video data. For example, human behavior detection may be performed using non-visual streams, such as depth maps. However, the depth maps can be obtained only from specific sensors, such as Kinect (Zhang, 2012) or from the similar ones. Even though the non-visual stream real-time processing is feasible to achieve on contemporary computers with reasonable accuracy, the principle drawback is in the need for special sensors and a lack of infrastructure available for these sensors. Moreover, alternative sensors often need to be calibrated, etc.

Section 1.1 describes the related work published on similar areas of computer vision science and section 1.2 presents the main characteristics of the dataset which will be used in this work for classifier evaluation.

After that Chapter 2 describes the classification pipeline; then Chapter 3 presents the experiments performed on the dataset; and finally in Chapter 4 conclusions are drawn.

1.1 Related Work

In the last decade, a number of papers with various concepts for action recognition were published. A significant part of those approaches are based on space-time feature point extraction, fixed-sized representation conversion, and finally, classifier creation. The most important approaches, shown below, are also explored in the presented work.

Wang et al. evaluated in (Wang et al., 2009) several combinations of feature extractors and feature descriptors using all important datasets available at the time. In this approach, video sequences are represented by a bag-of-words and vocabulary is created using k-means algorithm. For classification purposes, the non-linear support vector machine with χ_2 kernel (Zhang et al., 2006) is used. The results are reported and measured using mean average precision.

Wang et al. (Wang et al., 2011) proposed in his later work a new way of extracting the space-time interest points, called Dense trajectories. The Dense trajectories extractor is based on the assumption that the search of the extrema across all three dimensions is not efficient because of the different characteristics of the space domain and the temporal domain. In this approach, the points are detected in the spatial domain and then tracked across the temporal domain. After the point trajectory is 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 (Histogram of oriented gradients) and HOF (histogram of optical flow) descriptors (the same as in the STIP extractor (Laptev and Lindeberg, 2003)), trajectory descriptor, and MBH descriptor were used. The trajectory descriptors are based on the trajectory shape represented by relative point coordinates as well as appearance and motion information over a local neighborhood of some size along the trajectory. The MBH (Motion boundary histogram) descriptor (Dalal et al., 2006) separates the optical flow fields into horizontal and vertical components (MBx, MBy). Spatial derivatives are evaluated for each of them and the orientation information is quantized into histograms, similarly to the HOG descriptor. MBH represents the gradient of the optical flow with constant motion information suppressed and only information about the changes is kept.

All of the above feature descriptors are used separately; they are transformed into a bag-of-words (Csurka et al., 2004) representation and used for training the multichannel non-linear SVM with χ_2 kernel in a similar fashion as in (Ullah et al., 2010). The accuracy of the algorithm is evaluated on present-day' datasets and is compared with other state-of-the-art papers using a mean average precision measure.

Ullah et al. (Ullah et al., 2010) has presented an extension of the standard bag-of-words approach, where the video is segmented semantically into mean-

ingful regions (spatially and temporally) and the bag-of-words histograms are computed separately for each region.

Le Q. V. (Le et al., 2011) has presented a method for learning of features from spatio-temporal data using independent subspace analysis.

Jain Mihir et al. proposed (Jain et al., 2013) to decompose visual motion into dominant and residual motions that are used in the feature detection part of the processing and also in the feature description process. He extracted the DCS (Divergence-Curl-Shear) descriptor (Jain et al., 2013) based on differential motion scalar quantities, divergence, curl and shear features. Later on, the VLAD (Jegou et al., 2012) coding technique from image processing was applied to action recognition problem.

1.2 Dataset

Marszalek et al. in (Marszalek et al., 2009) proposed a dataset with twelve action classes and ten scene classes annotated, which is acquired from 69 Hollywood movies. The dataset[1] is built from movies containing human actions and processed using script documents and subtitle files which are publicly available for these movies. The script documents contain the scene captions, dialogs and the scene descriptions; however, they are usually not quite precisely synchronized with the video. The subtitles have video synchronization so they are matched to the movie scripts and this fact can be used for improvements in video clip segmentation. By analyzing 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 so that the two subsets do not share segments from the same movies.

These twelve action classes are : *answering the phone, driving car, eating, fighting, getting out of the car, hand shaking, hugging, kissing, running, sitting down, sitting up and standing up*. Examples of the frames contained in these video sequences are shown in Figure 1. The framerate of the videos is 25 fps.

Two training parts of the dataset exist: the automatic part generated using the above mentioned procedure, and the clean part which is manually corrected using visual information from the video. The test part is manually corrected in the same way as in the clean training part of the dataset. In both cases, the correction is performed in order to eliminate ”noise” from the dataset and consequently to create better classifiers. In the work described above, some experiments were performed. The processing chain consisted of feature extraction the SIFT (Lowe, 2004) image and STIP (Laptev and Lindeberg, 2003) space-time extractors, both converted into a bag-of-words representation and then used in multichannel χ^2 Support Vector Machine for classification purposes. The results are measured using a mean average precision metric across all of the classes and presented as the first evaluation performed on this dataset.

[1]The Hollywood2 dataset can be downloaded from: http://www.di.ens.fr/ laptev/actions/hollywood2/

2 CLASSIFICATION PIPELINE

In the presented work, we used the standard bag of words pipeline processing and the space-time features combined in the multi-kernel SVM. The method of processing is described in more detail below.

2.1 Feature Vectors Processing

A space-time interest points feature extractor produces a large number of feature vectors. The number of the feature vectors differ for different video shots, but every feature vector has the same dimensionality. The space-time feature points extractor consists of two parts: the search part that seeks for interest points location across both space and time domains and the description part that examines the neighborhoods of such detected feature point locations.

The consequent part of the feature vectors processing is the conversion of the feature vectors into the form, where a single fixed-sized feature vector depicts the video shot. This is achieved through a visual vocabulary which is used for a bag-of-words (Csurka et al., 2004) feature vector construction.

The visual vocabulary is created as a model for the representation of the low-level feature space, which is formed by a set P of representatives Pi (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 is too small, very large clusters will exist and the discriminative power of the whole solution may be adversely affected.

The k-means square-error partitioning method (Duda et al., 2000) can be used for these purposes. This algorithm iteratively processes data so that it assigns feature points to their closest cluster centers and recalculates the cluster centers. The k-means algorithm converges only to local optima of the squared distortion and does not determine the k parameter. It

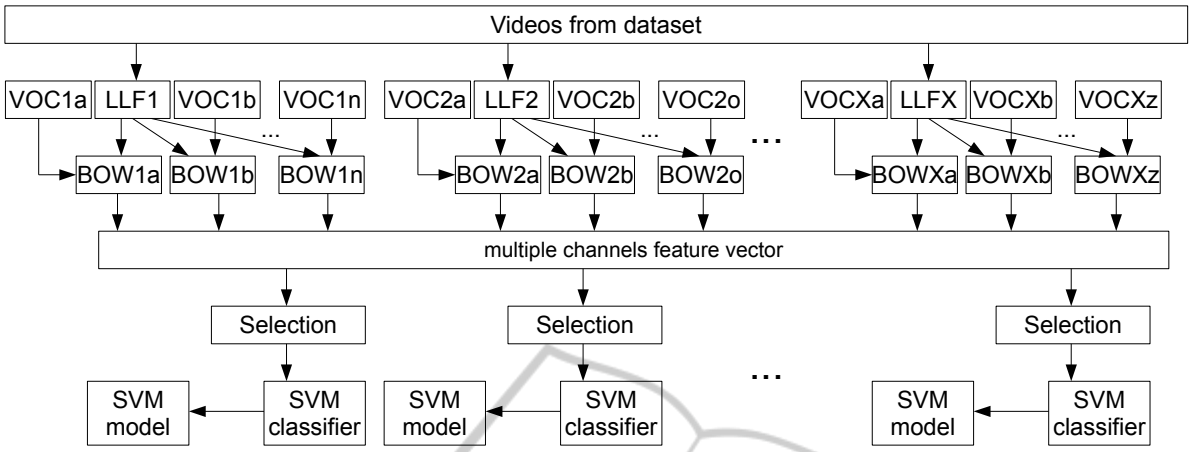

Figure 2: The standard algorithm schema; The *LLFx* boxes depict the low level feature extractors, the *VOCx* represent the vocabularies constructed from related feature extractors, the *BOWx* boxes depict the bag-of-words creation units, the *multiple channels feature vector* is constructed by concatenation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

can be parametrized through the specification of the number of iterations and number of output clusters.

The bag-of-words can represent the video sequence or its part using one feature vector with a constant dimension. The dimension does not depend on the number of the local space-time vectors nor on 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 output histogram is incremented. This simple form of assignment is sometimes called the hard assignment which 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 (nearest codewords 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 groups are assigned a weight corresponding to their closeness to the vector; finally each of the corresponding output histogram bins are added to the weight of the appropriate clusters. The most frequently used method of weight computation is through exponential function of the distance to the cluster center $w_i(a) = exp(-\frac{(d(a,p_i))^2}{2\sigma^2})$, where d is an Euclidean distance from the cluster center to the vector, while the 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.

2.2 SVM Models Creation

The bag-of-words feature vectors are combined using a non-linear multi-kernel support vector machine (Zhang et al., 2006), as depicted in Figure 2, with a multichannel gaussian kernel (Zhang et al., 2006). The kernel shall be defined as:

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

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 from a set of channels C, $D_c(A,B)$ is the χ^2 distance between two bag-of-words, A and B are the input vectors of the form:

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

where set of channels C can be defined as:

$$C = \{\langle 1,n_1 \rangle, \langle n_1+1,n_2 \rangle, \dots, \langle n_i-1,n_i \rangle\} \quad (3)$$

The bag-of-words distance $D_c(A,B)$ is defined as:

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

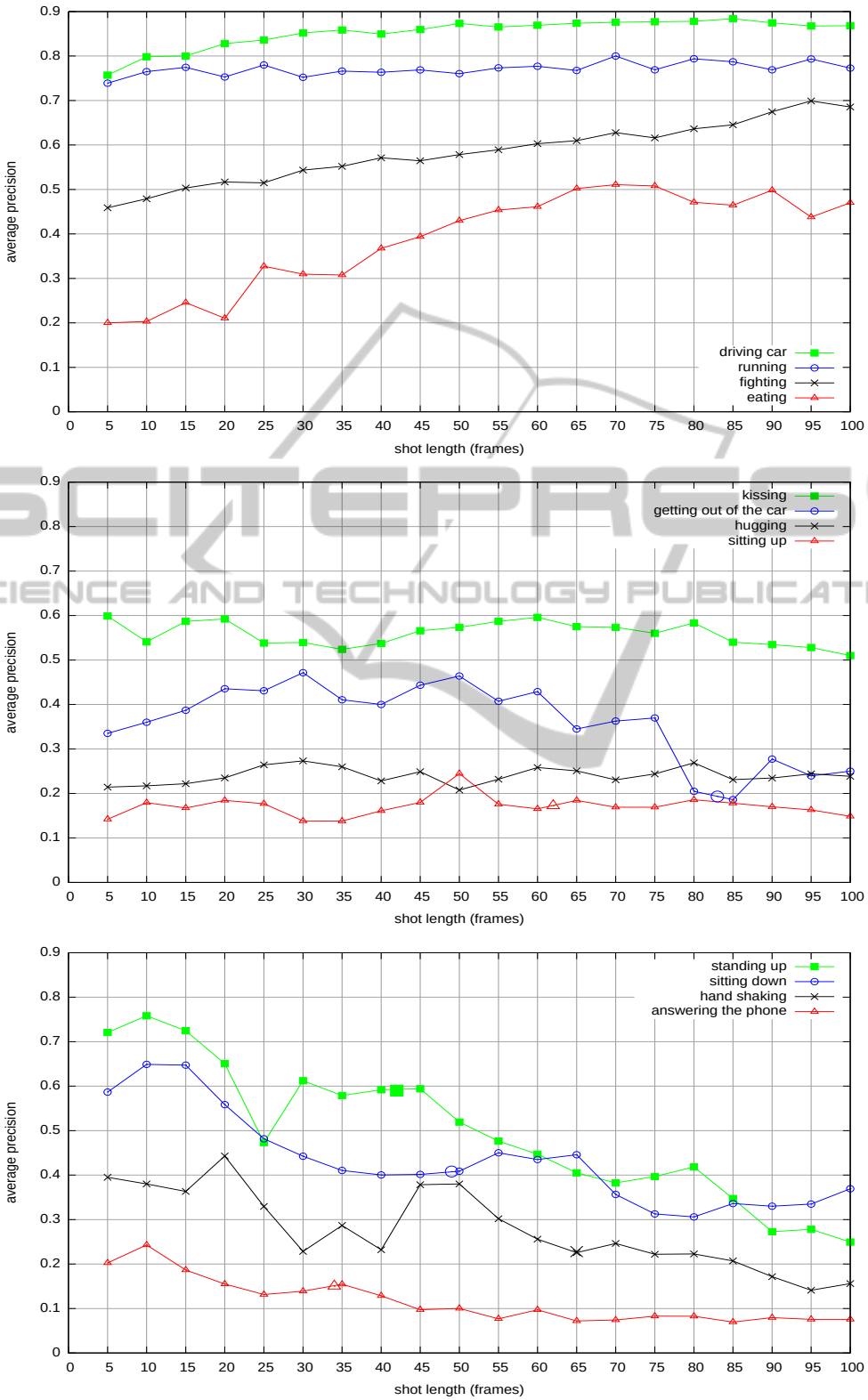

Figure 3: Dependency of the average precision on the length of the shot achieved for all classes contained in the Hollywood2 dataset. The dependency is split into 3 separate charts in order to improve readability. It should be noted that the big marks indicate the average action length shown in the charts for actions shorter than 100 frames.

The best ratio of combined channels $\{c_k, c_l, \ldots, c_z\} \in C$ for a given training set is estimated using a coordinate descend method. The set of input channels needs to be specified outside of the training process. Beside this SVM building procedure requires the number of input parameters that affect the classifier accuracy; these parameters are automatically evaluated using the cross-validation approach (Hsu et al., 2010). The classifier creation process may be apprehended in the whole procedure as a black-box unit which for a given input automatically creates the best performing classifier.

3 EXPERIMENTS

The purpose of the experiments performed in the presented work has been to determine the minimum length of a video shot containing the action to be recognized, and for which the recognition accuracy is still comparable to the state-of-the-art solutions. Such experiments correspond to real-time search in video or, for example, to a situation where a video record is searched for the action (at randomly selected positions, positions determined by the application).

We have used the pipeline presented in Chapter 2 and the Hollywood2 dataset presented in Chapter 1.1 (Marszalek et al., 2009). This dataset, as mentioned earlier, contains twelve action classes from Hollywood movies, 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*.

We investigated the recognition algorithm behavior in such a way that pieces of video containing the action were presented to the algorithm at randomly selected positions inside the actions. For example, the actions were known to start earlier than the beginning of the processed piece of video, and ended only after the end of the presented piece of video. For this purpose, we had to reannotate the Hollywood2 dataset (all three its parts - train, autotrain and test) to obtain precise beginning and ending frames of the actions.

In our experiments, we have been trying to depict a dependency between the length of video shot, being an input to the processing, and the accuracy of the output. We have set the minimum shot length to 5 frames, more precisely the 5 frames from which the space-time point features are extracted. The maximum shot length was set to 100 frames and the frame step was set to 5 frames.

The space-time features extractor process N previous and N consequent frames of the video sequence in order to evaluate the points of interest for a single frame. Therefore, 2*N should be added to every figure concerning the number of frames to get the total number of frames of the video sequence to be processed. In our case, N was equal to 4 so that, for example, the 5 frames processed in Figure 3 mean 13 frames of the video.

A classifier has been constructed for every video shot length considered. The training samples were obtained from the training part of the dataset in the following way: the information of the start and stop position in the currently processed sample was used and large number of the randomly selected subshots were obtained. The training dataset has 823 video samples in total and from each sample, we extracted 6 subshots on average.

The actual evaluation of the classifier has been done four times in order to obtain the information about reliability of the solution. Also, the above mentioned publications used the 823 samples for evaluation purposes and we wanted our results to be directly comparable. The results shown in Table 1 and Figure 3 present the average of the results of the four runs. For this purpose we have randomly determined a position of starting frame of a testing subshot within a testing sample four times. The above approach brings us two benefits - the final solution accuracy can be measured using an average precision metric and the results obtained through the testing can be easily comparable to the published state-of-the-art solutions. The results were compared with the accuracy achieved on the video sequences with completely unrestricted size that are close to the state-of-the-art (Reznicek and Zemcik, 2013).

The parameters for feature processing and classification purposes were as follows: the tested feature extractor is the dense trajectories extractor (Wang et al., 2011), which produces four types of descriptors, namely: HOG, HOF, DT and MBH. These four feature vectors were used separately. For each descriptor a vocabulary of 4000 words was produced using the *k*-means method and the bag-of-words representation was produced with the following parameters: $\sigma = 1$, the number of searched closest vectors is 16; these values and codebook size were evaluated in (Reznicek and Zemcik, 2013) and are suitable for bag-of-words creation from space-time low-level features. In the multi-kernel SVM creation process all four channels (bag-of-words representations of HOG, HOF, DT and MBH descriptors) are combined together, no searching for a better combination is performed.

The above described evaluation procedure was repeated for every class contained in the Hollywood2 dataset. For each class, we are presenting the graph of

Table 1: Results of the experiments. The first four columns show the accuracy (average precision) for the selected video sizes, the consequent column shows the reference accuracy reached for unrestricted video size, and the final column shows the minimum number of frames needed to achieve 90% of the precision achieved using the unrestricted video size.

Action	Video size (frames) 5	10	30	90	Unrestricted video size accuracy	Number of frames to achieve 90% accuracy
driving car	0.757	0.798	0.852	0.874	0.848	10
running	0.739	0.765	0.752	0.769	0.812	10
fighting	0.459	0.479	0.543	0.675	0.718	90
eating	0.2	0.203	0.309	0.498	0.326	25
kissing	0.599	0.541	0.540	0.535	0.597	5
getting out of the car	0.335	0.36	0.471	0.277	0.358	5
hugging	0.214	0.217	0.273	0.235	0.264	25
sitting up	0.142	0.179	0.138	0.17	0.163	10
standing up	*0.721*	*0.758*	*0.612*	*0.273*	*0.598*	*5*
sitting down	*0.587*	*0.649*	*0.442*	*0.33*	*0.654*	*10*
hand shaking	*0.395*	*0.38*	*0.229*	*0.172*	*0.232*	*5*
answering the phone	*0.201*	*0.243*	*0.139*	*0.08*	*0.225*	*10*

dependency between the video sample shot length and the system best accuracy, as well as the figure showing the number of frames needed to achieve 90% of state-of-the-art accuracy.

It should be noted that the first group of results (*driving car, running, fighting, eating*) corresponds well to the expectation that the accuracy will be increasing with the length of the shot. The second group (*kissing, getting out of the car, hugging, sitting up*) showed approximately constant accuracy depending on the length. This was probably due to the fact that the actions in these shots are recognized based on some short motions inside the actions. The final group (*standing up, sitting down, hand shaking, answering the phone*) showed decreased accuracy depending on the length. The reason is that the actions were too short (length shown using markers in Figure 3) and so increasing the length of the shot only "increased noise" and did not bring any additional information. The expectations were also not fulfilled for generally poorly recognized actions where the experiments showed that the shot length is not correlated with accuracy.

Based on our experiments, for example, the running activity can be recognized in 10 frames of space-time features with 0.765 accuracy (90% of the state-of-the-art) which corresponds to the 18 frames in total and approximately 0.72s of real-time.

4 CONCLUSIONS

In this paper, we presented novel results showing dependency between the length of a video sequence containing certain action and the accuracy of the action recognition. For this purpose, we used the Hollywood2 dataset and we demonstrated the results on the human action recognition.

The results indicate that the idea suggesting the accuracy of the action recognition being dependent on the length of the video sequence is generally right with the exception of some short and poorly recognized classes. Our research also indicates that significant differences exist between the sizes of video sequences to recognize different actions.

The results of our work can be useful in the design the real-time action recognition systems as well as in applications, such as a video database search, where real-time is not critical but where the computation performance is the bottleneck.

Future research includes extension of the classier creation process, where an algorithm for automatic optimal combination of input feature channels should be constructed in order to minimize the processing time, while preserving accuracy through minimization of the input channel count. Additionally, more efficient feature extraction and processing approaches from both the point of view of frames processed and the point of view of the computational time needed will be considered.

ACKNOWLEDGEMENTS

The presented work has been supported by Center of Excellence, Ministry of Education, Youth and Sports Czech Republic, "IT4Innovations" ED1.1.00/02.0070, Center of competence, Technology Agency of the Czech Republic, "V3C - Visual Computing Competence Center" TE01020415, and

CRAFTERS ConstRaint and Application driven Framework for Tailoring Embedded Real-time Systems, Artemis JU, project 7H12006.

REFERENCES

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *Proceedings of the 9th European conference on Computer Vision - Volume Part II*, ECCV'06, pages 428–441, Berlin, Heidelberg. Springer-Verlag.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Hsu, C.-W., chung Chang, C., and jen Lin, C. (2010). A practical guide to support vector classification.

Jain, M., Jégou, H., and Bouthemy, P. (2013). Better exploiting motion for better action recognition. In *CVPR - International Conference on Computer Vision and Pattern Recognition*, Portland, États-Unis.

Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., and Schmid, C. (2012). Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716.

Laptev, I. and Lindeberg, T. (2003). Space-time interest points. In *IN ICCV*, pages 432–439.

Le, Q., Zou, W., Yeung, S., and Ng, A. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Marszalek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936.

Reznicek, I. and Zemcik, P. (2013). Action recognition using combined local features. In *Proceedings of the IADIS Computer graphics, Visulisation, Coputer Vision and Image Processing 2013*, pages 111–118. IADIS.

Ullah, M. M., Parizi, S. N., and Laptev, I. (2010). Improving bag-of-features action recognition with non-local cues. In *Proceedings of the British Machine Vision Conference*, pages 95.1–95.11. BMVA Press. doi:10.5244/C.24.95.

Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176.

Wang, H., Ullah, M. M., Klser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *University of Central Florida, U.S.A*.

Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2006). Local features and kernels for classification of texture and object categories: A comprehensive study. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 13–13.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10.