Knives are Picked before Slices are Cut: Recognition through Activity Sequence Analysis

Ahmet Iscen
Computer Engineering Department
Bilkent University
ahmet.iscen@bilkent.edu.tr

Pinar Duygulu
Computer Engineering Department
Bilkent University
duygulu@cs.bilkent.edu.tr

ABSTRACT

In this paper, we introduce a model to classify cooking activities using their visual and temporal coherence information. We fuse multiple feature descriptors for fine-grained activity recognition as we would need every single detail to catch even subtle differences between classes with low inter-class variance. Considering the observation that daily activities such as cooking are likely to be performed in sequential patterns of activities, we also model temporal coherence of activities. By combining both aspects, we show that we can improve the overall accuracy of cooking recognition tasks.

Categories and Subject Descriptors

I.2.10 [**ARTIFICIAL INTELLIGENCE**]: Vision and Scene Understanding—*video analysis*; I.5 [**PATTERN RECOGNITION**]: Applications—*Computer vision*

Keywords

Activity recognition, Action Recognition, Cooking activities

1. INTRODUCTION

With the advancement of technology and internet, research in human activity recognition has improved dramatically over the recent years. The early research was focused on basic activities that were easily distinguishable, such as human body movements like walking, bending, punching etc. On the other hand, need for recognition of more specific activities that can be similar to each other, or *fine-grained* human activities, has gained big demand due to new possible applications, such as elderly care. As the elderly population is increasing, monitoring of individual's homes becomes an important issue to reduce the cost of care. This requires the recognition of a subject's daily activities accurately, and these activities are usually very similar to each other with low inter-class variance.

Cooking activities, given in [12] can be viewed as *fine-grained* activities; specific activities with very low inter-class class variability. These activities, such as *cut slices, cut stripes, cut dice*, are not very different from each other, and can be found hard to be distinguished between not only by computers, but even by humans. Therefore, solving fine-grained activity recognition task remains an important challenge in activity recognition domain.

Classification of fine-grained activities is a challenging task. Usually different classes are very similar to each other with only subtle differences that can be hard to be represented by spatio-temporal visual descriptors. Consider some of the cooking activities in our dataset, such as *cut apart, cut dice, cut off ends* and *cut slices* in Figure 1. These activities look very similar to each other, and it is often difficult to decide which feature descriptor to use in order obtain the best classification result. Furthermore, some feature descriptors can work well in some subset of activities, while others give better results in other activities.

We can also argue that when someone enters a kitchen, they follow a certain sequence of activities when they are cooking. In that sequence, certain activities are more likely to come after other certain activities. For example, when someone performs the activity *cut dice*, just by considering a normal cooking process, our intuition tells us that the subject might want to put whatever they have cut into a bowl, and the next activity is likely to be *put in bowl.*

In our work, we propose a classification model that considers both the preceding activities, and spatio-temporal visual information of the observed activity, as shown in Figure 2. We train separate models for each component, and combine them to obtain our final decision.

This paper is organized as follows. In Section 3 and its subsections, we introduce each component of our model and show how we connect them to each other. In Section 4 we give implementation details and conduct various experiments using our model.

2. RELATED WORK

There have been many research in computer vision and multimedia domains that have focused on activity recognition. The reader can refer to [1] for extensive survey on past activity recognition research. Some of the research, such as [9] focused on using *interest points* in order to recognize activities. Wang et al. [18] have shown that the activity recognition can be improved by extracting dense trajectories from frames. Sener et al. [15] have shown that it is also possible to recognize human activity by only looking at still images.

Other works showed that human activity can be recognized with sequential approaches. These approaches have

CEA'13, October 21, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-2392-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2506023.2506025.

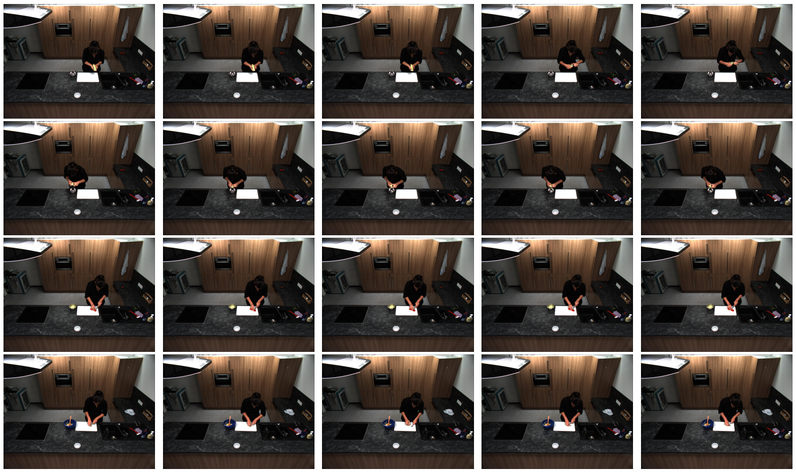

Figure 1: Frames of the subject performing very similar actions.The subject is performing *cut apart* on the top row, *cut dice* on the second row, *cut slices* on the third row, and *cut-off ends* on the last row.

modeled activity sequences using probabilistic models. One example is [7], where Ikizler and Forsyth model human activities by HMMs. Other works such as [6, 14] have also used sequential and visual information for human activity recognition. One clear distinction of our work from previous sequential approaches is that we do not model frames as sequences, but rather we model sequence of activities where each activity is a *collection of frames*.

Classification combination for different feature spaces have been proposed in [8]. More recently, Hashimoto et al. [5] have shown that it can be applied to current problems that require multiple feature spaces, but they ignore setting the *reliability term* and weigh each classifier the same. Also, cooking related activities have been previously studied using other datasets like [3] and [16].

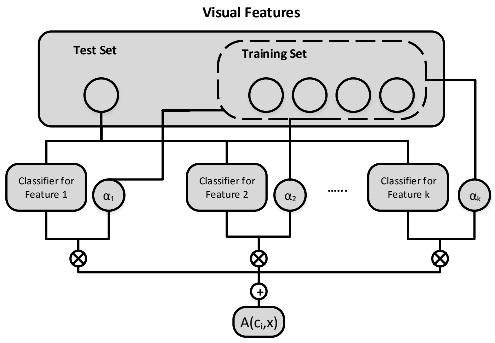

Figure 3: The framework for visual model explained in Section 3.1

3. OUR METHOD

In this section, we give a detailed explanation of our method, which is the combination of two parts; visual model and temporal coherence model. In visual based model, we combine different spatio-temporal visual feature descriptors that give information about the visual appearance of the current activity, and in temporal coherence model we consider the preceding activities that come before the observed activity in a sequence.

3.1 Visual Model With Multiple Feature Descriptors

The simplest idea of combining different features is to concatenate their feature vectors. Although this approach is extremely simple, Rohrbach et al.[12] have shown that it actually yields to better results in classification of cooking activities than using any of the individual feature descriptors.

However, concatenation of feature vectors has one large drawback; curse of dimensionality. As we concatenate more and more feature descriptors, the dimensionality of our feature vectors will also increase, which is not desirable. In fact, when we concatenate the feature vectors of the cooking activities dataset in [12] by using four feature descriptors (HOG, HOF [10], MBH [2], and trajectory information) with bag-of-word representations of 4000 bins, we obtain a 16000 dimensional representation for each observation in our data, which is clearly very high dimensional. This approach limits the number of feature descriptors that we can use only to a few, and still introduces large dimensional feature vectors which would not be efficient when performing other operations on them, such as training an SVM model with a non-linear kernel.

Nevertheless, we must be able to use multiple feature descriptors in our visual model for fine-grained activities. Each feature descriptor looks at an activity from a different *perspective*, and since these activities can be very similar to each other, we must be able to combine the *views* from these perspectives to obtain a better classification result than just looking at one feature descriptor. However, we must also pay attention to efficiency, and avoid issues like the curse of dimensionality that is caused by just concatenating different features.

By considering these constraints, we train an individual classifier for each feature space. By performing cross-validation on the training set, we also find a *confidence factor* for each individual classifier, which gives an idea about how each single classifier would perform generally, and is used to weigh

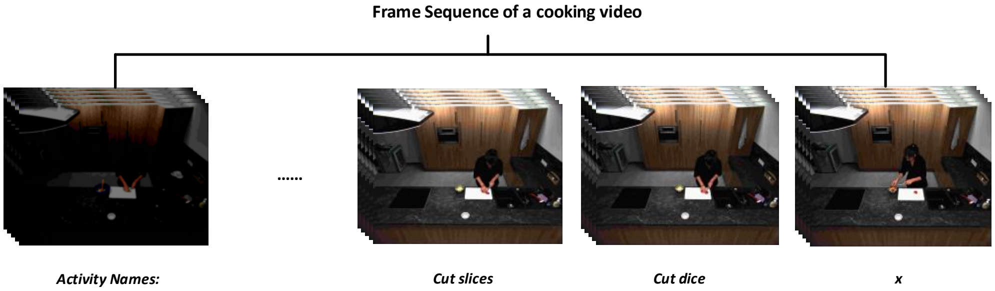

Figure 2: In our classification framework, we classify a newly observed activity x by considering its preceding activities and spatio-temporal visual descriptors. For the example above, we consider only 2 previous activities which are *cut slices* and *cut dice*. Our model classifies x as *put in bowl*, which is an accurate decision.

the results of that particular classifier in the test stage. Finally, we combine the results of each individual classifier. Figure 3.1 shows the framework for this model.

3.1.1 Training Individual Classifiers for Each Feature Space

Our goal is to bring out the best of each feature space, and consider each of them in order to make the best decision for the final classification. Therefore, we consider each group independently in the beginning. That is, we assume that the individual classification performance of a concept group has no effect on another, and should therefore be treated completely separately. This also allows us to have an agnostic classification method that can be used with any type of feature descriptors.

In order to implement this idea, we train a separate, individual discriminative classifier P_j for each feature space j. For a given activity representation in feature space j, the role of each individual classifier is to give a score for query activity belonging to each class. Our choice of classifier here is a multi-class (one vs all) SVM for each feature space. Since SVM classifiers do not output probabilities, but rather confidence scores, we convert scores to probabilities using Platt et al.'s method described in [11].

3.1.2 Finding a Confidence Factor For Each Classifier

One of the contributions of our work is to introduce the *confidence factor*, which gives a different weight for each classifier. After training a separate classifier for each concept group, we must be able to combine them properly before making a final decision. Hashimoto et al. [5] use a similar classification combination framework to combine multimodal data, however they use the same value to weigh each individual classifier. The drawback of this approach is that poor-performing classifiers would have the same contribution in the final decision making process as a well-performing classifier, and effect it negatively. Therefore, we try to come up with a value for each classifier that would weigh its decision confidence.

With the aim of generating a generalized estimation of each classifier, we introduce the notation of *confidence factor* to our framework. A *confidence factor* is a measure for weighing the decisions made by a certain classifier. In order to calculate this value, we divide the training set of each classifier into 10-folds, and perform cross-validation. We take the average of all accuracy values for each fold, and assign it as the confidence factor.

3.1.3 Combining Individual Classifiers

With the introduction of the confidence factor α, the probability results obtain from each classifier is multiplied by its confidence. This step adds the required weighting measure for our individual classifiers. Combination of results from each classifier can be expressed with the following formula:

$$A(ci, x) = \frac{\sum_{j=1}^{F} \alpha_j \cdot P_j(f_j = c_i)}{p(x)}. \qquad (1)$$

where x is an instance, c_i is the ith activity class, F is the number of different feature spaces, f_j is the representation of x in feature space j, $P_j(f_j = c_i)$ is the probability of f_j belonging to class c_i by using the individual classifier of jth feature space, and α_j is the confidence factor for the jth classifier.

3.2 Temporal Coherence Model of Activities

Up to this point, we have only considered how each activity *looks like* by using the feature descriptors that were extracted from its frames. Although this piece of information captures important aspects of the current activity, it is usually not enough to classify an activity only based on this information. We need to find other ways to distinguish the current activity from the others, and combine it with the visual information to come up with a final decision.

We assume that natural daily activities usually follow patterns, therefore knowledge of preceding activities would help us guess what the current activity is. This idea is a Markov Assumption and can be represented by a Markov Chain [4] mathematically. Markov Chains have a property such that, given the current event, the future event is conditionally independent of the events of the past. It can be formulated as:

$$P(x_i | x_1, x_2, ..., x_{i-1}) = P(x_i | x_{i-1}) \qquad (2)$$

The formulation above considers only the previous element before making a decision. We can extend it to consider n previous elements, and re-write it as:

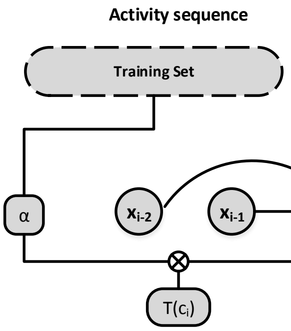

Figure 4: The framework for temporal model explained in Section 3.2

$$P(x_i|x_{i-n},...,x_{i-1}) = \frac{P(x_{i-n},...,x_{i-1},x_i)}{P(x_{i-n},...,x_{i-1})} \quad (3)$$

This is also called an *n-order Markov Process*, where the future event is conditionally independent on n previous events given the current event. Using the sequence of activities that each subject performs during their cooking course, we model our temporal coherence model using an *n-order Markov Process*, where each x_i is the name of the ith activity.

Additionally, we use the *confidence factor* idea introduced in Section 3.1.2, and multiply it with the probability obtained from the Markov Chain in order scale its efficiency by how much we expect it perform well generally. We find the *confidence factor* using the same way explained in Section 3.1.2. Our temporal coherence model with the confidence factor is:

$$T(c_i) = P(x_i|x_{i-n},...,x_{i-1}) \cdot \alpha \quad (4)$$

The framework for our temporal coherence model is shown in Figure 3.2.

3.3 Making a Final Decision

Now that we have modeled both aspects of our classification system, we must be able to combine them in order to make a final decision. We want our temporal coherence model to effect the result of the visual model, therefore we assign the output of temporal coherence model like a prior probability value for our visual model to find the final decision y:

$$P(c_i|x) = \frac{T(c_i) \cdot A(ci,x)}{p(x)}. \quad (5)$$

$$y = \underset{i}{\operatorname{argmax}}\, P(c_i|x)$$

where x is an observation, c_i is the ith activity class, $T(c_i)$ is the prior probability of class c_i based on the result from the temporal coherence model described in Section 3.2, and $A(ci,x)$ is the result of visual model from Section 3.1.

4. EXPERIMENTS

In this section, we give details about the dataset and implementation details that were used for our experiments, and analyze the results of our model and its variations for classification tasks.

4.1 Dataset

The dataset that we have used is MPII Cooking Activities Dataset [12], which contains cooking activities that were performed by 12 subjects. Each subject was asked to prepare a dish in a realistic environment, and their actions from one frame to another during the preparation were labeled as one of the 65 cooking activities. During our experiments we did not consider the frames that were labeled as *Background Activity*, like the original paper [12], so our evaluation actually consisted of 64 classes.

For evaluation, we followed the same process described in the original paper of the dataset. The activities of 5 subjects were always used for training, and for the remaining 7 subjects, one subject was used as test set and others were added to the training set in each round. In the end, we have 7 different evaluations, one for each subject used as the test set.

4.2 Implementation Details and Settings

For all experiments, same settings were used. As our visual feature descriptors for Section 3.1, we have used four feature descriptors, HOG,HOF,MBH and trajectory speed, that are available to be used with the dataset[1].

To train each individual feature descriptor model explained in Section 3.1.1, we train a one-vs-all SVM for each class using mean SGD [13] with a χ^2 kernel approximation [17] with $C = \frac{10}{N}$ where N is the size of the training set. While this is the same kind of classifier that was used in the original paper of the dataset [12], our results were slightly lower, probably due to not being able to select the optimal parameter value. For temporal coherence model explained in Section 3.2, we experimented with Markov Chains of different order, and concluded that $n = 2$ gives us the best results.

4.3 Classification Experiments

4.3.1 Visual Information Only

In this experiment, we do not use any temporal coherence information explained in Section 3.2. We first perform classification only by using each of the four feature descriptors described in Section 4.2, then combine their result to observe if our method from Section 3.1 has any effect on improving the classification.

As we can see from the accuracy results on Figure 5, accuracy scores obtained by the combination of feature descriptors increase accuracy for almost all subjects. We can also see that our combination method outperforms simple feature concatenation method for all subjects except for *Subject 18.*

4.3.2 Controlled Temporal Coherence Only

For this experiment, we avoid all the visual information from the feature descriptors, and train a model only by considering the sequence of activities as described in 3.2 using a *controlled environment.* This means that for each new observation, we retrieve the previous class labels from ground truth values. The result of this experiment can be seen in Table 1. Not surprisingly, this model does not perform very well when used only by itself, even in a controlled environment. This shows that activities cannot be classified by only

[1] http://www.d2.mpi-inf.mpg.de/mpii-cooking

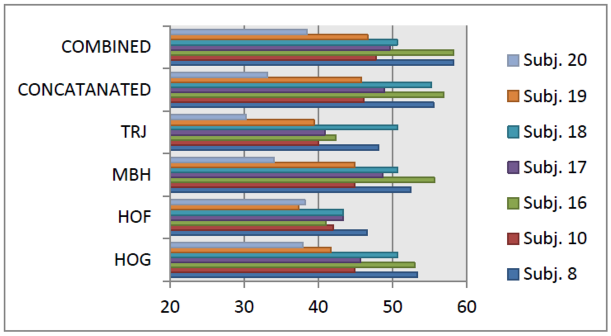

Figure 5: Classification by using visual information only.

Table 1: Classification by using only the temporal coherence information.

Subject	Accuracy
8	24.23
10	34.62
16	29.14
17	41.57
18	16.45
19	32.17
20	38.85

using the sequence information, or *temporal coherence*, and we need to make use of visual information as well.

4.3.3 *Visual Information + Controlled Temporal Coherence*

This experiment combines both visual and temporal coherence models before making a final decision as explained in Section 3.3. In a controlled environment, we use the ground truth values for previous actions that are used with the temporal coherence model. Therefore, results obtained by these experiments would give us the top results we can achieve using our model. Results of this experiment can be seen in Table 2. As we can see, by combining visual and temporal information we obtain higher classification accuracy for all subjects.

4.3.4 *Visual Information + Semi-Controlled Temporal Coherence*

This experiment is same as Section 4.3.3, except that it is not performed in a controlled environment. Class labels for previous activities that are used with temporal coherence are obtained by running a *visual only* classification on them. Results of this experiment are on Table 3.

Table 2: Visual Information + Controlled Temporal Coherence

Subject	Visual	Temp. Coh.	Combined
8	58.28	24.23	61.04
10	47.76	34.62	69.23
16	58.28	29.14	62.25
17	49.65	41.57	68.82
18	50.66	16.45	51.97
19	46.67	32.17	55.65
20	38.54	38.85	58.60

Table 3: Visual Information + Semi-Controlled Temporal Coherence

Subject	Accuracy
8	58.90
10	48.39
16	58.94
17	50.58
18	51.32
19	44.93
20	40.45

Table 4: Visual Information + Automatic Temporal Coherence

Subject	K=3	K=5	K=7
8	59.98	59.35	59.16
10	47.96	47.88	47.32
16	58.75	59.39	61.32
17	51.26	51.82	52.36
18	51.90	52.11	51.92
19	48.42	48.89	48.66
20	40.51	40.39	40.11

4.3.5 *Visual Information + Automatic Temporal Coherence*

This is the purest, most automatic version of our experiments. In this experiment everything is automatic, once a classification is made for the new observation, that classification value is used as the class label for temporal coherence model of future observation. We perform this experiment in windows of size K, and report the results. Results can be seen in Table 4.

5. CONCLUSION

In this work, we have shown that the temporal coherence information can be combined with visual information in order classify and recognize activities. As we can see from our experiment results in Table 5, overall classification scores for most subjects improves when visual and temporal information is used together.

In the controlled environment, where we obtain the labels for previous actions from ground truth values, we can see that combining visual and temporal information together can help improve the recognition accuracy.

In experiments in semi-automatic, and automatic environments, where the previous action labels are not obtained by ground truth values now but they are also classification results, we can see the actual effect of our model.These experiments also show that the overall accuracy is improved by our model, especially for *visual and automatic temporal coherence* experiment, where the obtained accuracy percentages are greater than the scores we obtain just by using *visual information.*

This gives us evidence to conclude that, we can model activity sequences to classify future activities, and *temporal coherence* model does improve the overall classification score.

Table 5: Comparison of All Experiments

Subject	Visual(Conc.)	Visual(Comb.)	Cnt. T.C.	Vis + Cnt. T.C.	Vis + Semi-Cnt. T.C.	Vis + Auto. T.C. (K=5)
8	55.52	58.28	24.23	61.04	58.90	59.35
10	46.15	47.76	34.62	69.23	48.39	47.88
16	56.95	58.28	29.14	62.25	58.94	59.39
17	48.96	49.65	41.57	68.82	50.58	51.82
18	55.26	50.66	16.45	51.97	51.32	52.11
19	45.80	46.67	32.17	55.65	44.93	48.89
20	33.12	38.54	38.85	58.60	40.45	40.39

6. ACKNOWLEDGMENTS

This study is partially supported by TUBITAK grant number 112E174.

7. REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16, 2011.

[2] N. Dalal, B. Triggs, and C. Schmid. 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, 2006. Springer-Verlag.

[3] F. De la Torre, J. K. Hodgins, J. Montano, and S. Valcarcel. Detailed human data acquisition of kitchen activities: the cmu-multimodal activity database (cmu-mmac). Technical report, RI-TR-08-22h, CMU, 2008.

[4] C. W. Gardiner et al. *Handbook of stochastic methods*, volume 3. Springer Berlin, 1985.

[5] A. Hashimoto, J. Inoue, K. Nakamura, T. Funatomi, M. Ueda, Y. Yamakata, and M. Minoh. Recognizing ingredients at cutting process by integrating multimodal features. In *Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating activities*, CEA '12, pages 13–18, New York, NY, USA, 2012. ACM.

[6] M. Hoai, Z. zhong Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[7] N. İkizler and D. Forsyth. Searching for Complex Human Activities with No Visual Examples. *International Journal of Computer Vision*, 80(3):337–357, Dec. 2008.

[8] Y. Ivanov, T. Serre, and J. Bouvrie. Error weighted classifier combination for multi-modal human identification. In *In Submission*, 2004.

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

[10] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, I. Rennes, I. I. Grenoble, and L. Ljk. B.: Learning realistic human actions from movies. In *In: CVPR. (2008*.

[11] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[12] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, United States, June 2012. IEEE, IEEE.

[13] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1641–1648, Washington, DC, USA, 2011. IEEE Computer Society.

[14] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1036–1043, Washington, DC, USA, 2011. IEEE Computer Society.

[15] F. Sener, C. Bas, and N. Ikizler-Cinbis. On recognizing actions in still images via multiple features. In *Proceedings of the 12th international conference on Computer Vision - Volume Part III*, ECCV'12, pages 263–272, Berlin, Heidelberg, 2012. Springer-Verlag.

[16] M. Tenorth, J. Bandouch, and M. Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009*, 2009.

[17] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[18] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, June 2011. MSR - INRIA.