To appear in the Proc. of the Eleventh IEEE International Conference on Computer Vision

A Biologically Inspired System for Action Recognition

H. Jhuang T. Serre L. Wolf* T. Poggio
Brain & Cognitive Sciences Department
McGovern Institute for Brain Research
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139
{hueihan,serre}@mit.edu wolf@cs.tau.ac.il tp@ai.mit.edu

Abstract

We present a biologically-motivated system for the recognition of actions from video sequences. The approach builds on recent work on object recognition based on hierarchical feedforward architectures [25, 16, 20] and extends a neurobiological model of motion processing in the visual cortex [10]. The system consists of a hierarchy of spatio-temporal feature detectors of increasing complexity: an input sequence is first analyzed by an array of motion-direction sensitive units which, through a hierarchy of processing stages, lead to position-invariant spatio-temporal feature detectors. We experiment with different types of motion-direction sensitive units as well as different system architectures. As in [16], we find that sparse features in intermediate stages outperform dense ones and that using a simple feature selection approach leads to an efficient system that performs better with far fewer features. We test the approach on different publicly available action datasets, in all cases achieving the highest results reported to date.

1. Introduction

Understanding the perception of actions in both humans and animals is an important area of research crossing the boundaries between several scientific disciplines from computer science to brain science and psychology. Motion recognition is one of the most challenging recognition problems in computer vision with important applications such as surveillance and human-machine interaction. At the same time, our understanding of the brain mechanisms responsible for the recognition of actions has progressed over the past decades (see [1] for a recent review) and a body of experimental data is growing rapidly.

*Now at the School of Computer Science at Tel Aviv University, Israel

The visual cortex is organized in two different pathways: a ventral stream which is usually thought of as dealing mainly with the processing of shape information and a dorsal stream involved with the analysis of motion information. Interestingly, the organization of these two pathways is very similar [23]. Their organization is hierarchical; aiming, in a series of processing stages, to gradually increase both the selectivity of neurons along with their invariance to 2D transformations (see [10]). In parallel, the size of the receptive fields of the neurons, *i.e.* the part of the visual field that if properly stimulated may elicit a response from the neuron, increases along the hierarchy.

These two pathways originate in the primary visual cortex (V1) where one can find two populations of cells: cells which are tuned to spatial orientations (*e.g.* a static vertical bar) and project to areas V2 and V4 of the ventral stream, and cells which are sensitive to directions of motions (*i.e.* a bar at a specific orientation and moving in a direction perpendicular to its orientation) and project to area MT and MST in the dorsal stream. The analysis of motion information proceeds in MT and MST where neurons have substantial position and scale invariance and are tuned to optical flow patterns, see [10]. It has been reported in these areas that neurons are selective for complex optical flow patterns, *e.g.* translation flows or opponent motion.

In this work, we speculate that neurons in intermediate visual areas of the dorsal stream such as MT, MST and higher polysensory areas are tuned to spatio-temporal features of intermediate complexity (see Section 2), which pool over afferent input units tuned to different directions of motion. This includes, but is not limited to, the optical flow neurons described above. Finally in higher polysensory areas, one can find neurons which are tuned to short chunks of actions (see [22] for a review).

Motivated by the recent success of biologically inspired approaches for the recognition of objects in real-world applications [25, 16, 20], we here extend a neurobiological model [10] of motion processing in the dorsal stream of the visual cortex. The model has only been applied so far to simple artificial stimuli [10, 28].

Our work is motivated by the similarity in the organization of the shape and motion pathways in the visual cortex and we here try to apply computational mechanisms that have proved to be useful for the recognition of objects to the recognition of actions. The idea of extending object descriptors to actions has already been shown to be a good one, as illustrated by the excellent performance in the non-biologically motivated system by Dollar *et al.* [5].

1.1. Our approach

Our approach is closely related to feedforward hierarchical template matching architectures that have been used for the recognition of objects in still images. These systems have been around for quiet some time now, starting with the work of Fukushima [8] and LeCun *et al.* [12]. Here we follow the more recent framework using scale and position invariant C_2 features [25, 16] that originated with the work of Riesenhuber & Poggio [21].

C_2 shape features In previous work [25, 16], a still gray-value input image is first analyzed by an array of Gabor filters (S_1 units) at multiple orientations for all positions and scales. Processing is then hierarchical: feature complexity and position/scale invariance are gradually increased by alternating between a template matching and a max pooling operation. That is, at the C_1 stage some tolerance to small deformations is obtained via a local max over neighborhoods of S_1 units (in both position and scale).

Next, template matching is performed over the C_1 maps, creating thousands of S_2 maps. At each position (and scale) a patch of C_1 units centered at that position (scale) is compared to each of d_1 prototype patches. Each prototype corresponds to a vector of size $4n^2$ which is obtained by cropping an $n \times n$ ($n = 4, 8, 12, 16$) patch of C_1 units at a given location and all 4 orientations. These d_1 prototype patches represent the intermediate-level features of the model, and are randomly sampled from the C_1 layers of the training images in an initial feature-learning stage.

At the top of the hierarchy, a vector of d_1 scale and position invariant C_2 features is obtained by computing a global max for each of the d_1 (multi-scale) S_2 feature maps. These feature vectors are finally passed to a linear SVM classifier to obtain a classification label. Here we propose to extend this approach to the recognition of actions.

Motion-direction sensitive S_1 units To extend the C_2 framework to the analysis of motion, we start by empirically searching for a suitable representation for the S_1 units. We compare 3 types of motion-direction sensitive S_1 units: a) Gradient-based features (I_t/I_x, I_t/I_y); b) Optical flow based features; c) Space-time oriented filters [29], which have been shown to be good models of motion-sensitive simple cells in the primary visual cortex [15]. Details about how the 3 different S_1 unit types were computed is given in Section 2.1. Interestingly, we find that the optical flow features previously used in [10, 4, 28] lead to worse performance than the gradient-based features and the space-time oriented filters.

Learning sparse spatio-temporal motion C_2 features Recently, Mutch & Lowe showed that C_2 features can be sparsified leading to a significant gain in performance [16] on standard object recognition databases (see also [20]). In addition, they also showed that applying a simple feature selection technique to the C_2 feature responses can lead to an efficient system which can perform better with less features.

Motivated by these findings, we experiment with the zero-norm SVM [31] feature selection technique. We find that a more compact C_2 feature representation can lead to significant decrease in the computation time taken by the overall system without sacrificing accuracy.

Adding new S_3 and C_3 stages Finally we experiment with an extension of the hierarchy which is specific to motion processing, *i.e.* to include time invariant S_3 and C_3 units. Preliminary experiments suggest that these units sometimes improve performance, but not significantly.

1.2. Related work

Typically, computer vision systems for the recognition of actions have fallen into two categories. One class of approaches relies on the tracking of object parts [32, 19, 3]. While these approaches have been successful for the recognition of actions from articulated objects such as humans (see [9] for a review), they are not expected to be useful in the case of less articulated objects such as rodents [5]. The other common class of approaches is based on the processing of spatio-temporal features, either global as in the case of low-resolution videos [33, 6, 2] or local for higher resolution images [24, 5, 7, 18].

Our approach falls in the second class of approaches to action recognition. It extends an earlier neurobiological model of motion processing in the dorsal stream of the visual cortex by Giese & Poggio [10]. While this model has been successful in explaining a host of physiological and psychophysical data, it has only been tested on simple artificial stimuli such as point-light motion stimuli [10, 28]. In particular, the model of [10] is too simple to deal with real videos. It lacks translation invariance and uses a limited handcrafted dictionary of features in intermediate stages.

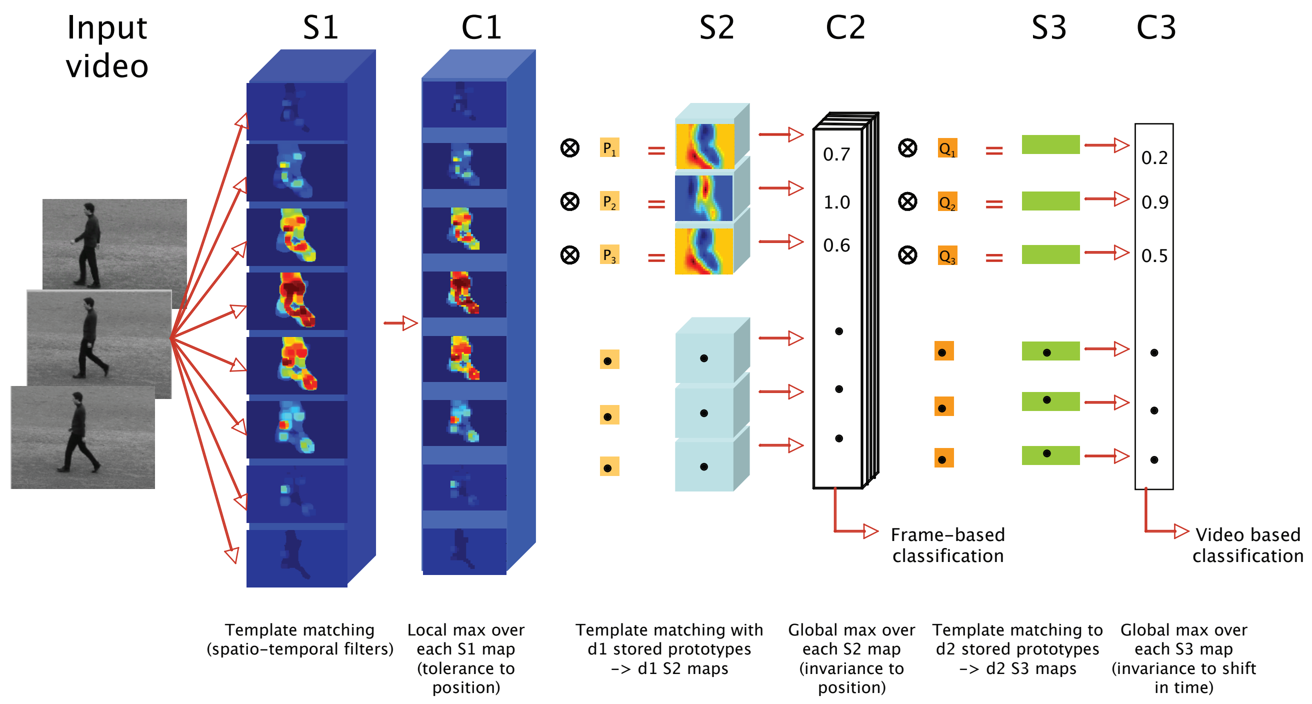
Input video
S1
C1
S2
C2
S3
C3
P1
P2
P3
0.7
1.0
0.6
Q1
Q2
Q3
0.2
0.9
0.5
Frame-based classification
Video based classification
Template matching (spatio-temporal filters)
Local max over each S1 map (tolerance to position)
Template matching with d1 stored prototypes -> d1 S2 maps
Global max over each S2 map (invariance to position)
Template matching to d2 stored prototypes -> d2 S3 maps
Global max over each S3 map (invariance to shift in time)

Figure 1. Sketch of the system (see text for details).

1.3. Main contributions

Our main contribution is the application of a neurobiological model of motion processing to the recognition of actions in complex video sequences and the surprising result that it can perform on par or better than existing systems on varying datasets. Indeed none of the existing neurobiological models of motion processing have been used on real-world data [10, 13, 4, 28, 11]. As recent work in object recognition has indicated, models of cortical processing are starting to suggest new algorithms for computer vision [25, 16, 20]. Conversely applying biological models to real-world scenarios should help constrain plausible algorithms.

In order to convert the neuroscience model of [10] into a real computer vision system, we altered it in two significant ways: We propose a new set of motion-sensitive units which is shown to perform significantly better and we describe new tuning functions and feature selection techniques which build on recent work on object recognition.

2. System overview

Our system is based on a hierarchy of increasingly complex and invariant space-time feature detectors. By alternating between a local maximum operation to increase the tolerance to local deformations (such as translation) and a template matching operation to increase the complexity of the feature detectors, the system gradually builds a representation which is tolerant to 2D changes (*e.g.* the position of the actor in the visual field) yet specific enough so as to allow fine discrimination between similar actions (*e.g.* jogging and running), see section 3.3.

2.1. S_1 units

The input to the system is a gray-value video sequence that is first analyzed by an array of S_1 units at all positions (see Fig. 1). In [25, 16] for the recognition of static objects, Gabor filters at multiple orientations were used. To extend these approaches to the analysis of motion, we experiment with three different types of S_1 units.

Space-time gradient-based S_1 units: These features are based on space and time gradients, which were used for instance, in the system by Zelnik-Manor & Irani [33]. We compute the spatial gradients along the x and y axis for each frame, denoted I_x and I_y as well as the temporal gradient of adjacent frames I_t. Motivated by optical-flow algorithms that are based on the constant-brightness assumption and by recent work on video matching [26], we consider two types of S_1 units: $|I_t/(I_x+1)|$ and $|I_t/(I_y+1)|$. The absolute value is taken to make features invariant to contrast reversal.

Optical flow based S_1 units: A second set of S_1 units was obtained by computing the optical flow of the input image sequence using Lucas & Kanade's algorithm [14]. We denote θ and ν, the direction and the magnitude of the optical flow at each pixel position in the current frame. As in [10], S_1 unit responses were obtained by applying the following equation:

$$b(\theta, \theta_p) = \{\frac{1}{2}[1 + cos(\theta - \theta_p)]\}^q \times exp(-|\nu - \nu_p|), \quad (1)$$

where θ_p determines the preferred direction of the S_1 unit and ν_p is its preferred speed. We use 4 directions $\theta_p = 0^o, 90^o, 180^o, 270^o$ and two speeds, both an intermediate speed ($\nu_p = 3$) and a higher speed ($\nu_p = 6$). The constant q, which controls the width of the tuning curve, was set to $q = 2$ as in [10, 4]. Taking all the possible combinations of ν_p and θ_p, we obtained 8 types of S_1 units.

Space-time oriented S_1 units: These units constitute the most direct extension to the object recognition systems by [25, 16]. In these approaches, S_1 units corresponded to $2D$ Gabor filters at multiple orientations. A natural way to extend these filters to the analysis of motion is to add a third temporal dimension to their receptive fields. Such models have been shown to agree quantitatively with the tuning properties of simple cells in the primary visual cortex which are motion-direction sensitive [15].

Several specific implementations of such motion-direction sensitive cells have been proposed (see [10] for references). Here we used the popular implementation by Simoncelli & Heeger which uses (3^{rd}) derivatives of Gaussians [29]. As for the optical-flow S_1 units, we used 8 space-time filters tuned to 4 directions ($0^o, 90^o, 180^o, 270^o$) and 2 speeds (3 and 6 pixels/frame). The size of the receptive fields of the corresponding S_1 units was $9(pixels) \times 9(pixels) \times 9(frames)$. The filter outputs were half-wave rectified.

2.2. C_1 units

Beyond the S_1 stage our system follows closely the approach by [25]. At the C_1 stage, down-sampling is performed for each S_1 type by computing a local max over an 8×8 grid of S_1 units with the same selectivity (*e.g.* same preferred motion direction and speed in the case of the space-time oriented S_1 units). C_1 units are computed every 4 pixels (each C_1 layer thus has $1/4$ the size of the input frame). As a result some tolerance to small shifts is gained during the transition from the S_1 to the C_1 stages.

2.3. S_2 units

The S_2 stage matches the C_1 maps of the current frame with d_1 stored templates that were extracted during a training phase (see below). At every position in the C_1 layer, we perform a template matching operation between the current patch of C_1 units centered at that position and each of the d_1 stored S_2 templates. These stored templates constitute the intermediate-level features of the model, and are randomly sampled from the C_1 layers of frames of training videos in an initial feature-learning stage. As in [25, 16], we used $n \times n$ templates (with $n = 4, 8, 12, 16$). We denote by l the number of S_1 types ($l = 2, 8, 8$ for the gradient based, optical flow based and space-time oriented S_1 units respectively). The matching is done across all C_1/S_1 types and each template thus has ln^2 coefficients.

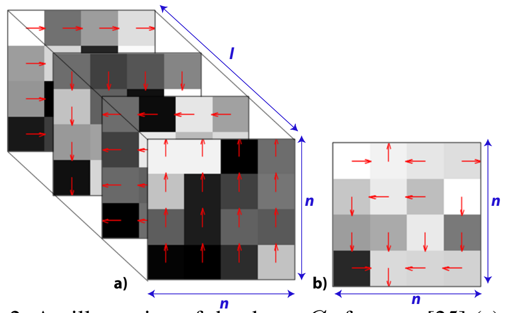

Figure 2. An illustration of the dense C_2 features [25] (a) *vs*. the sparse C_2 features [16] (b).

To obtain the d_1 S_2 templates, we randomly extract 500 C_1 patches from training videos for each of the action category and for each of the 4 template sizes. The extraction is completely random, *i.e.* we selected patches extracted from random frames and random training sequences and at random X, Y positions within the frame. This leads to 2,000 stored templates per action category and a total number of templates d_1 = 10,000 - 18,000 extracted during training (the datasets used contains 5-9 action categories).

In our experiments (Section 3.3), we compare two alternative S_2-unit response functions for the template matching operation: the dense Euclidean distance adapted from [25] and the sparse normalized dot-product by [16]. For the dense features [25], the template matching is computed over all ln^2 coefficients of the template. Conversely, in the sparse case, only the strongest coefficients are stored for each pixel location of the template. Thus only n^2 sparse coefficients are stored. The difference between dense and sparse C_2 features is illustrated in Fig. 2.

Another difference is in terms of the form taken by the template matching procedure. Let x denote a C_1 patch from the current frame and let $\mathbf{w}^k$ denote the k-th stored template (prototype). For the sparse features of [16], we use the normalized dot-product, where the response y_k of the S_2 unit, which corresponds to the C_1 patch $\mathbf{x}$ is given by:

$$y_k = \frac{\mathbf{x} \cdot \mathbf{w}^k}{||\mathbf{x}|| \times ||\mathbf{w}^k||}. \quad (2)$$

For the dense features of [25], we simply compute the Euclidean distance, *i.e.* the response y_k is given by:

$$y_k = -||\mathbf{x} - \mathbf{w}^k||. \quad (3)$$

2.3.1 C_2 units

In this stage, a global max across all positions is taken for each S_2 feature maps. With d_1 stored templates during training, the corresponding d_1 S_2 features maps obtained after matching between the C_1 maps and the d_1 stored templates thus lead to d_1 spatio-temporal C_2 units. More specifically, for each one of the d_1 S_2 feature maps we obtain one scalar, which is the maximum value in the map.

2.3.2 S_3 and C_3 stage

While the C_2 units achieve a high degree of spatial invariance, they lack temporal invariance. We experimented with the addition of a new layer that encodes the content of the video with a higher degree of temporal invariance. These C_3 features are computed, as before, by performing a template matching stage to create S_3 features, followed by a pooling stage that creates C_3 features.

The prototypes used to compute the S_2 features are confined to a small patch of video sequence. This cannot be transferred directly to form prototypes for the S_3 units since, unlike the retinotopic C_1 features, the C_2 features are unordered. Instead, we randomly select, for each S_3 prototype, a random subset of the C_2 features (half the total number of C_2 features). S_3 units are then obtained by considering 7 consecutive frames at a random location in the input sequence and for each frame, storing the coefficients of the top 50% input units). In our experiments we sample $d_2 = 300$ S_3 templates.

For a new input video, the S_3 unit responses are obtained by computing the similarity between the d_2 stored prototypes and the video for all frames (temporal matching). The C_3 units are then computed, for each video sequence, as the maximum response over the duration of the video. This results in d_2 C_3 units per video regardless of the length of the video. The location of the maximum response is discarded as in the C_2 unit computation.

2.3.3 Classification stage

The final classification stage is a linear multi-class SVM classifier trained using the all-pairs method. When using C_2 features as input to the classifier, we first compute the C_2 features over a random set of $n = 500$ frames for each action category in the training set. For a test video, we thus obtain a classification label for each frame. A classification label for the entire video was obtained by majority voting.

When using the C_3 features as input to the classifier, we compute the C_3 units for the entire video for all the videos in the training set. For a test video, a single label is obtained for each one of the videos.

2.3.4 C_2 feature selection with zero-norm SVM

As in [16], we have performed experiments with the zero-norm SVM [31] of Weston *et al.* Instead of regularizing the norm of the hyperplane $||\mathbf{w}||^2$, the classifier tries to optimize the objective function:

$$||\mathbf{w}||^0 + \mathbf{C}\sum_{i=1}^{N}\zeta_i, \text{ such that } (\mathbf{w}^T\mathbf{x_i} + b) > 1 - \zeta_i \quad (4)$$

The zero norm $||w||^0$ here indicates the count of the feature used. In practice this is done in multiple rounds. At each round an SVM classifier is trained on the pool of C_2 features and the training set is re-weighted using the weights of the trained SVM. Typically this leads to sparser SVM weights at each stage. We compare the performance of both the full and compact feature sets in Section 3.3.

3. Experiments

We have conducted an extensive set of experiments to evaluate the performance of the proposed action recognition system on three publicly available datasets: two human action datasets (KTH and Weizmann) and one mice action dataset (UCSD). Details about the datasets are given below.

3.1. Datasets

KTH human : The KTH human action dataset [24] contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping. These actions are performed several times by twenty-five subjects in four different conditions: outdoors ($s1$), outdoors with scale variation ($s2$), outdoors with different clothes ($s3$) and indoors with lighting variation ($s4$). The sequences average about 4 seconds in length. We down-sampled the sequences to a spatial resolution of 160×120 pixels. We split the dataset as: actions of 16 randomly drawn subjects for training and actions of the remaining 9 subjects for testing. The system performance is based the average of five random splits.

Weizmann human: The Weizmann human action dataset [2] contains eighty-one low resolution (180×144 pixels) video sequences with nine subjects performing nine actions: running, walking, jumping-jack, jumping forward on two legs, jumping in place on two legs, galloping-sideways, waving two hands, waving one hand, and bending. For training we use the actions from 6 random subjects (6 videos per action category), the actions of the remaining 3 subjects are then used for testing. The size of the subject in this dataset is about half the size of the subject in the KTH human action dataset. However, we run experiments on the two sets using the same parameters. The system performance is evaluated by the average of five random splits.

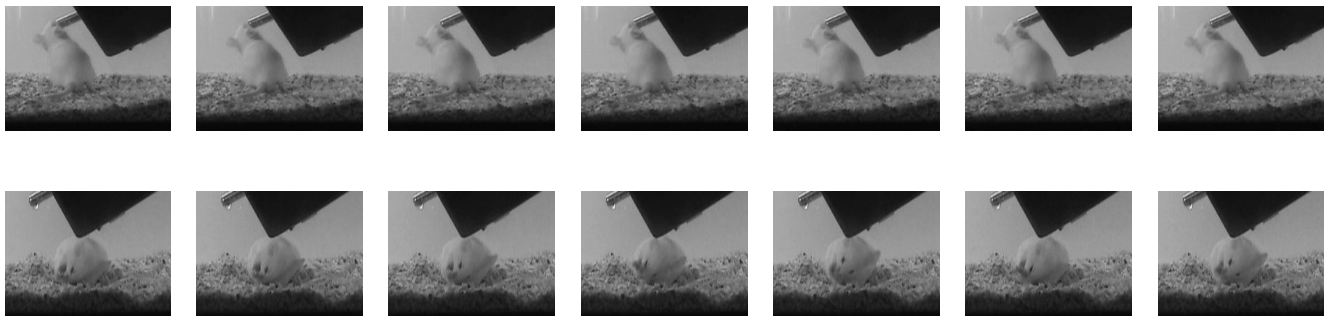
Figure 3. Sample videos from the mice dataset (1 out 10 frames displayed with a frame rate of 15 Hz) to illustrate the fact that the mice behavior is minute.

UCSD mice: The UCSD mice behavior dataset [5] contains seven subsets, each being recorded at different points in a day such that multiple occurrences of actions within each subset vary substantially. There are five actions in total: drinking, eating, exploring, grooming and sleeping. The sequences have a resolution of 240×180 pixels and a duration of about 10 seconds. This dataset presents a double challenge. First the actions of the mice are minute (see Fig. 3 for examples) and second the background of the video is typically noisy (due to the litter in the cage). Each split, we randomly choose 4 subsets for training and the remaining 3 subsets for testing. The system performance is the average of five random splits.

Preprocessing We preprocessed the datasets to speed-up our experiments: On the KTH human and UCSD mice datasets we used the openCV GMM background subtraction technique based on [30]. In short, a mixture of Gaussians model was used to identify the foreground pixels of each frame. From the foreground mask, we extracted a bounding box (full height and half the width of the frame) for each frame. For the Weizmann Human dataset, the bounding boxes were extracted directly from the foreground masks provided with the dataset.

3.2. Benchmark algorithms

For benchmark we use the algorithm by Dollar *et al.* which has been compared favorably to several other approaches [33, 6, 18] on the KTH human and UCSD mice datasets described earlier. In short, the approach detects interest points in the spatio-temporal domain and extracts *cuboids*, *i.e.* spatio-temporal windows of pixel values, around each point detected. These cuboids are further matched to a dictionary of cuboid-prototypes learned from sequences in the training set. Finally, a vector description is obtained by computing the histogram of cuboid-types of each video, and a SVM classifier is used for classification. The code for [5] was graciously provided by Piotr Dollar.

3.3. Results

We have studied several aspects and design alternatives for the system. First we show that zero-norm feature selection can be applied to the C_2 units and that the number of features can be reduced from $12,000$ down to ≈ 500 without sacrificing accuracy. We then proceeded to apply feature selection for all the remaining experiments and compare different types of motion-direction sensitive input units. We also compared the performance of sparse *vs*. dense C_2 features and present initial preliminary results with the addition of a high-level C_3 stage.

3.3.1 Selecting C_2 features with the zero-norm SVM

The following experiment looks at feature selection and in particular how the performance of the system depends on the number of selected features. For this experiment, we used space-time oriented S_1 units and sparse C_2 features. Performance is evaluated on the four conditions of the KTH dataset.[1] In the first iteration, all $12,000$ patches of C_1 units extracted from the training set of images were used to compute the C_2 features. In each of the following iteration, only features with a weight $|w_i| > 10^{-3}$ were selected.

Table 1 compares the performance of each round. In agreement with previous results on object recognition [16], we found that it is possible to reduce the number of C_2 features quiet dramatically (from $\sim 10^4$ down to $\sim 10^2$) with minimal loss in accuracy. This is likely due to the fact that during learning, the S_2 prototypes were extracted at random locations from random frames. It is thus expected that most of the prototypes should belong to the background and should not carry much information about the category of the action. In the following, feature selection was performed on the C_2 features for all the results reported.

[1]For computational reason the performance reported is based on a single split of the KTH dataset.

		1	5	10	15	20
$s1$	No. feat.	12000	3188	250	177	158
	accu.	91.7	91.7	89.3	88.9	90.3
$s2$	No. feat.	12000	4304	501	340	301
	accu.	86.6	86.6	85.2	87.0	85.7
$s3$	No. feat.	12000	3805	392	256	224
	accu.	90.3	90.7	89.4	88.4	88.0
$s4$	No. feat.	12000	3152	313	217	178
	accu.	96.3	96.3	96.3	95.3	95.0
Avg	accu.	91.2	91.3	90.1	90.0	89.8

Table 1. Selecting features: System performance for different numbers of selected C_2 features at rounds 1, 5, 10, 15 and 20 (see text for details).

	[5]	GrC_2	OfC_2	StC_2
KTH $s1$	88.2	**94.3** / 92.7	92.8 / **93.3**	89.8 / **96.0**
s.e.m. $s1$	±1.9	±1.7 / ±3.2	±2.8 / ±2.9	±3.1 / ±2.1
KTH $s2$	68.3	86.0 / **86.8**	80.7 / **83.1**	81.3 / **86.1**
s.e.m. $s2$	±2.1	±3.9 / ±3.9	±4.0 / ±3.9	±4.2 / ±4.6
KTH $s3$	78.5	85.8 / **87.5**	89.1 / **90.0**	85.0 / **88.7**
s.e.m. $s4$	±2.9	±2.7 / ±3.3	±3.8 / ±3.5	±5.3 / ±3.2
KTH $s4$	90.2	91.0 / **93.2**	92.9 / **93.5**	93.2 / **95.7**
s.e.m. $s4$	±1.8	±2.0 / ±1.9	±2.2 / ±2.3	±1.9 / ±2.1
Avg	81.3	89.3 / **90.0**	88.9 / **90.0**	87.3 /**91.6**
s.e.m. Avg	±2.2	±2.6 / ±3.1	±3.2 / ±3.1	±3.6 / ±3.0
UCSD	75.6	78.9 / **81.8**	**68.0** / 61.8	76.2 / **79.0**
s.e.m.	±4.4	±4.3 / ±3.5	±7.0 / ±6.9	±4.2 / ±4.1
Weiz.	86.7	91.1 / **97.0**	**86.4** / **86.4**	87.8 / **96.3**
s.e.m.	±7.7	±5.9 / ±3.0	±9.9 / ±7.9	±9.2 / ±2.5

Table 2. Comparison between three types of C_2 features (gradient based GrC_2, optical flow based OfC_2 and space-time oriented StC_2). In each column, the number on the left *vs*. right corresponds to the performance of dense *vs*. sparse C_2 features (see text for details). $s_1, \ldots s_4$ corresponds to different conditions of the KTH database (see Section 3.1) and *Avg* to the mean performance across the 4 sets. Below the performance on each dataset, we indicate the standard error of the mean (s.e.m.).

3.3.2 Comparing different C_2 feature-types

Table 2 gives a comparison between all three types of C_2 features: gradient based GrC_2, optical flow based OfC_2 and space-time oriented StC_2 features. In each column, the number on the left *vs*. the right corresponds to the performance of dense [25] *vs*. sparse [16] C_2 features (see Section 2 for details). $s_1, \ldots s_4$ corresponds to the different conditions of the KTH database (see Section 3.1).

Overall the sparse space-time oriented and the gradient-based C_2 features (GrC_2 and StC_2) perform about the same. The poor performance of the OfC_2 features on the UCSD mice dataset is likely due to the presence of the litter in the cage which introduces high-frequency noise. The superiority of sparse C_2 features over dense C_2 features is in line with the results of [16] for object recognition.

	GrC_3	OfC_3	StC_3
KTH $s1$	**92.1** / 91.3	84.8 / **92.3**	89.8 / **96.0**
KTH $s2$	81.0 / **87.2**	80.1 / **82.9**	81.0 / **86.1**
KTH $s3$	89.8 / **90.3**	84.4 / **91.7**	80.6 / **89.8**
KTH $s4$	86.5 / **93.2**	84.0 / **92.0**	89.7 / **94.8**
Avg	87.3 / **90.5**	83.3 / **89.7**	85.3 / **91.7**
UCSD	73.0 / **75.0**	**62.0** / 57.8	71.2 / **74.0**
Weiz.	70.4 / **98.8**	79.2 / **90.6**	83.7 / **96.3**

Table 3. Comparison between three types of C_3 units (gradient based GrC_3, optical flow based OfC_3 and space-time oriented StC_3). In each column, the number to the left *vs*. the right corresponds to the performance of C_3 features computed from dense [25] *vs*. sparse [16] C_2 features. The results are based on the performance of the model on a single split of the data.

3.3.3 Comparing different C_3 feature-types

We have started to experiment with high-level C_3 features. Table 3 shows some initial results with three different types of motion-direction sensitive input units (see caption). Overall the results show a small improvement using the C_3 features *vs*. C_2 features on two of the datasets (KTH and Weiz) and a decrease in performance on the third set (UCSD).

3.3.4 Running time of the system

A typical run of the system takes a little over 2 minutes per video sequence (KTH human database, 50 frames, Xeon 3Ghz machine), most of the run-time being taken up by the $S_2 + C_2$ computations (only about 10 seconds for the $S_1 + C_1$ or the $S_3 + C_3$ computations). We have also experimented with a standard background subtraction technique [30]. This allows us to discard about 50% of the frame thus cutting down processing time by a factor of 2 while maintaining a similar level of accuracy. Finally, our system runs in Matlab but could be easily implemented using multi-threads or parallel programming as well as General Purpose GPU for which we expect a significant gain in speed.

4. Conclusion

We have applied a biological model of motion processing to the recognition of human and animal actions. The model accounts only for part of the visual system, the dorsal stream of the visual cortex, where motion-sensitive feature detectors analyze visual inputs. It has also been suggested [10] that another part of the visual system, the ventral stream of the visual cortex, involved with the analysis of shape may also be important for the recognition of motion (consistent with recent work in computer vision [17] which has shown the benefit of using shape features in addition to motion features for the recognition of actions). Future work will extend the present approach to integrate shape and motion information from the two pathways. Another extension is to incorporate top-down effects, known to play an important role for the recognition of motion (*e.g.* [27]), to the present feedforward architecture.

5. Acknowledgements

The authors would like to thank Piotr Dollar for providing the code for the benchmark system [5] in Section 3.3 as well as for his help and assistance with using the code.

This research was sponsored by grants from: DARPA Contract No. HR0011-04-1-0037, DARPA Contract No. FA8650-06-7632, National Science Foundation-NIH (CRCNS) Contract No. EIA-0218506, and National Institutes of Health (Conte) Contract No. 1 P20 MH66239-01A1. H. Jhuang was supported by the Taiwan National Science Council (TMS-094-1-A032).

Additional support was provided by: Central Research Institute of Electric Power Industry (CRIEPI), Daimler-Chrysler AG, Eastman Kodak Company, Honda Research Institute USA, Inc., Komatsu Ltd., Merrill-Lynch, NEC Fund, Oxygen, Siemens Corporate Research, Inc., Sony, Sumitomo Metal Industries, Toyota Motor Corporation, and the Eugene McDermott Foundation.

References

[1] R. Blake and M. Shiffrar. Perception of human motion. *Annu. Rev. Psychol.*, 58:47–73, 2007.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[3] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *IJCV*, 56(3):179–194, 2004.

[4] A. Casile and M. Giese. Critical features for the recognition of biological motion. *J. Vis.*, 5:348–360, 2005.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal feature. In *VS-PETS*, 2005.

[6] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

[7] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid models for human motion recognition. In *CVPR*, 2005.

[8] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cyb.*, 36:193–202, 1980.

[9] D. Gavrilla. The visual analysis of human movement: A survey. *CVIU*, 73(1):82–98, 1999.

[10] M. Giese and T. Poggio. Neural mechanisms for the recognition of biological movements and action. *Nat. Rev. Neurosci.*, 4:179–192, 2003.

[11] J. Lange and M. Lappe. A model of biological motion perception from configural form cues. *J. Neurosci.*, 26(11):2894–2906, 2006.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, Nov. 1998.

[13] J. Lee and W. Wong. A stochastic model for the detection of coherent motion. *Biol. Cyb.*, 91:306–314, 2004.

[14] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, 1981.

[15] J. A. Movshon, I. D. Thompson, and D. J. Tolhurst. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *J. Physiol.*, 283:53–77, 1978.

[16] J. Mutch and D. Lowe. Multiclass object recognition using sparse, localized hmax features. In *CVPR*, 2006.

[17] J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[18] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[19] D. Ramanan and D. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2004.

[20] M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies, with application to object recognition. In *CVPR*, 2007.

[21] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2:1019–1025, 1999.

[22] G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nat Rev Neurosci*, 2(9):661–670, 2001.

[23] H. Saito. *Brain Mechanisms of Perception and Memory*, pages 121–140. Oxford Univ. Press, 1993.

[24] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

[25] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex cortex. In *CVPR*, 2005.

[26] E. Shechtman and M. Irani. Space-time behavior based correlation. In *CVPR*, 2005.

[27] M. Shiffrar and J. Freyd. Apparent motion of the human body. *Psychol. Sci.*, 1:257–264, 1990.

[28] R. Sigala, T. Serre, T. Poggio, and M. Giese. Learning features of intermediate complexity for the recognition of biological motion. In *ICANN*, 2005.

[29] E. Simoncelli and D. Heeger. A model of neural responses in visual area MT. *Vis. Res.*, 38:743–761, 1998.

[30] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[31] J. Weston, A. Elisseeff, B. Scholkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *JMLR special Issue on Variable and Feature Selection*, 3:1439–1461, 2002.

[32] Y. Yacoob and M. Black. Parameterized modeling and recognition of activities. *CVIU*, 73(2):232–247, 1999.

[33] L. Zelnik-Manor and M. Irani. Event-based video analysis. In *CVPR*, 2001.