Learning Spatio-Temporal Representations for Action Recognition: A Genetic Programming Approach

Li Liu, Ling Shao, *Senior Member, IEEE,* Xuelong Li, *Fellow, IEEE,* and Ke Lu

***Abstract*—Extracting discriminative and robust features from video sequences is the first and most critical step in human action recognition. In this paper, instead of using handcrafted features, we automatically learn spatio-temporal motion features for action recognition. This is achieved via an evolutionary method, i.e., genetic programming (GP), which evolves the motion feature descriptor on a population of primitive 3D operators (e.g., 3D-Gabor and wavelet). In this way, the scale and shift invariant features can be effectively extracted from both color and optical flow sequences. We intend to learn data adaptive descriptors for different datasets with multiple layers, which makes fully use of the knowledge to mimic the physical structure of the human visual cortex for action recognition and simultaneously reduce the GP searching space to effectively accelerate the convergence of optimal solutions. In our evolutionary architecture, the average cross-validation classification error, which is calculated by an support-vector-machine classifier on the training set, is adopted as the evaluation criterion for the GP fitness function. After the entire evolution procedure finishes, the best-so-far solution selected by GP is regarded as the (near-)optimal action descriptor obtained. The GP-evolving feature extraction method is evaluated on four popular action datasets, namely KTH, HMDB51, UCF YouTube, and Hollywood2. Experimental results show that our method significantly outperforms other types of features, either hand-designed or machine-learned.**

***Index Terms*—Action recognition, feature extraction, feature learning, genetic programming (GP), spatio-temporal descriptors.**

I. Introduction

HUMAN action recognition [1]–[3], as a hot research area in computer vision, has many potential applications such as video search and retrieval, intelligent surveillance systems, and human-computer interaction. Despite its popularity, how to precisely distinguish different actions still remains challenging, since variations in lighting conditions, intraclass differences and complex backgrounds all pose as obstacles for robust feature extraction and action classification.

Generally, the basic approach to action recognition contains the following two stages: 1) feature extraction and representation and 2) action classification. For the first stage, there are mainly two groups of methods: 1) local feature-based and 2) holistic feature-based.

Within local feature-based methods, unsupervised techniques (e.g., cuboid detector [4] and 3D Harris corner detector [5]) are first applied to detect interest points around which the most salient features, such as: histogram of 3D oriented gradients (3DHOG) [6], 3D scale invariant feature transforms [7], and histogram of "optical flow" (HOF) [8], are extracted. Then the bag-of-features (BOF) scheme is utilized to form a codebook and map obtained features in histogram representations which are finally fed to a classifier for action classification. The local feature-based methods tend to be more robust to complex backgrounds and occlusion in real- istic actions [9], however, this kind of sparse representation is often not precise and informative because of the quantization error during codebook construction and the loss of structural configuration among local features. Another weakness of local approaches is that the detected spatio-temporal features are usually not distinctive and invariant enough, because the 3D local feature detectors are extended from their 2-D counterparts without fully exploiting the intrinsic differences between static images and dynamic video sequences. Because of these reasons, holistic feature-based methods have recently attracted significant attention in action recognition research.

On the other hand, the holistic approaches represent actions using visual information from the whole sequence and have also been utilized in a variety of applications. Commonly, shape, intensity, and color features are used for the holis- tic representation of an action. The structure and orienta- tion information of texture and shape can be successfully extracted by mimicking the biological mechanism of visual cortex for human perception. Color features have the advantage of being invariant with respect to scaling, rotation, perspective, and partial occlusion. The classical approaches to compute the holistic features for action recognition were

Manuscript received August 28, 2014; revised December 5, 2014; accepted January 30, 2015. Date of publication February 13, 2015; date of current version December 14, 2015. This work was supported in part by the Key Research Program of the Chinese Academy of Sciences under Grant KGZD-EW-T03, and in part by the National Natural Science Foundation of China under Grant 61125106. This paper was recommended by Associate Editor S. X. Yang. *(Corresponding author: Ling Shao.)*

L. Liu and L. Shao are with the College of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K. (e-mail: ling.shao@ieee.org).

X. Li is with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China.

K. Lu is with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with Beijing Center for Mathematics and Information Interdisciplinary Sciences, Beijing, China.

developed by [10]–[12], etc., which are able to encode more visual information by preserving spatial and temporal structures of the human action occurring in a video sequence. However, holistic representations are sensitive to geometric and photometric distortions and shifting. Moreover, preprocessing steps, such as background subtraction, spatial and temporal alignments, segmentation and tracking, are often required.

The methods introduced above are all based on handcrafted techniques [13], [14] designed and tuned by human experts, which, however, may achieve "good" performance in a particular given domain and often result in poor performance on other applications. How to design an adaptive methodology to extract spatio-temporal features with discriminative recognition capabilities for any user-defined application still remains an open research question.

As an alternative to handcrafted solutions based on deep domain knowledge, genetic programming (GP), a powerful evolutionary method inspired by natural evolution, can be employed to automatically solve problems without prior knowledge of the solutions. In the present setting, we wish to identify the feature descriptor (i.e., the sequence of primitive operations, the composition and order of which are unknown) to maximize recognition performance on a human action recognition task. This is an NP-hard search problem [15] that evolutionary methods may solve in a tractable amount of computer time compared to the exhaustive enumerative search. GP has been used to address a wide range of practical problems producing human-competitive results and even patentable inventions. As a search frame- work, GP can typically escape the local minima in the optimization landscape which may trap deterministic search methods.

In this paper, we adopt GP for designing holistic descriptors that are adaptive to action domains and robust to shift, scaling and background cluttering for action recognition. Given a group of primitive 3D processing operators and a set of labeled training examples, GP evolves better-performing individuals in the next generation. Eventually, a best-so-far individual can be selected as the final solution (i.e., the near-optimal descriptor). The GP-evolved spatio-temporal descriptors can extract and fuse the meaningful information from the original sequences and the corresponding optical flow motion data. We systematically evaluate the method on the KTH, HMDB51, YouTube, and Hollywood2 datasets to demonstrate its performance and generalizability. For comparison, we also show that the proposed method is superior to some previously-published hand-crafted solutions.

The main contributions of this paper can be summarized as follows.

1) GP is used to automatically "evolve" spatio-temporal feature descriptors that are adaptive and discriminative for action recognition without profound knowledge of the action datasets.
2) The GP-learned descriptors provide an effective way to simultaneously extract and fuse the color and motion (i.e., optical flow) information into one feature representation.

This paper is organized as follows. In Section II, some related work is summarized. The detailed architecture of our method is presented in Section III, and relevant experiments and results are described in Section IV. In Section V, we conclude this paper and outline possible future work.

II. Related Work

As this paper falls in the category of holistic representations, we mainly review methods of holistic spatio-temporal representations for action recognition.

Bobick and Davis [10] presented motion templates through projecting frames onto a single image, namely motion history images (MHI) and motion energy images. This kind of motion templates can capture the motion patterns occurring in a video sequence. However, this simple representation only gives satisfactory performance where the background is relatively static. Efros *et al.* [16] proposed a motion descriptor based on smoothed and aggregated optical flows over a spatio-temporal volume, which is centered on a moving figure. This descriptor has been proven to be suitable for distant objects, but the moving figure needs to be localized quite accurately. Schindler and Van Gool [17] found that very short snippets (1–7 frames) are sufficient for basic action recognition. They applied log-Gabor filters to raw frames to extract form features and optical flow filters to extract motion features. In addition, Gorelick *et al.* [18] extracted spatio-temporal features, such as local space-time saliency, action dynamics, shape structure, and orientation, based on the properties of Poisson equation solutions. Moreover, some recent discriminant analysis methods have also shown superior performance for action recognition, such as slow feature analysis (SFA) [19] which extracts the slowly varying and relevant stable features from the quickly changed action videos. SFA has been proved to be effectively used in constructing the visual receptive fields of the cortical neurons. General tensor discriminant analysis and Gabor features originally proposed for gait recognition [20] can be also applied to action recognition. These handcrafted features usually involve a lot of engineering work to design and tune and are not adaptive to different datasets.

Besides handcrafted features, there have also been a few works on learning feature representations for action recognition. Le *et al.* [21] have proposed using unsupervised feature learning as a direct way to learn invariant spatio-temporal features from unlabeled video data. Furthermore, Taylor *et al.* [12] have introduced a model that learns latent representations of image sequences from pairs of successive images. Similarly, Ji *et al.* [11] developed a 3D convolutional neural network (CNN), which is directly extended from its 2-D counterpart, for feature extraction. In a 3D CNN, motion information in multiple adjacent frames is captured through performing convolutions over spatial and temporal dimensions. The convolutional architecture of their model allows it to scale to realistic video sizes whilst using a compact parametriza- tion. Recently, deep belief network (DBN) [22] also shows its capacity to automatically learn multiple layers of nonlin- ear features from images and videos. However, the number of

parameters to be learned in those deep learning models [23] is very large, sometimes too large relative to the available number of training samples, which unfortunately restricts their applicability.

Within the area of evolutionary computation, evolution-based methods simulate biological evolution to automatically generate solutions for user-defined tasks, such as: genetic algorithms (GA), memetic algorithms (MA), particle swarm optimization (PSO), ant-colony systems (ACS), and GP. Generally, these are heuristic and population-based searching methods. They all attempt to move from one population to another population in a single iteration with probabilistic rules. In particular, GA seeks the solution of a problem in the form of a string of numbers (traditionally binary, although the best representations are usually those that reflect something about the problem being solved), by applying operators such as recombination and mutation (sometimes one, sometimes both). Bhanu *et al.* [24] have proposed an adaptive image segmentation system based on a GA. In their method, the GA is an effective way of searching the hyperspace of segmentation parameter combinations to determine the set which maximizes a segmentation quality criterion. Besides GA, PSO, which is inspired by the social behavior of migrating birds trying to reach an unknown destination, has been used for feature selection and classification in computer vision tasks. In [25], PSO is incorporated within an AdaBoost framework for face detection. Dynamic clustering using PSO has been proposed for unsupervised image classification in [26]. Additionally, a multiobjective PSO for discriminative feature selection was proposed in [27] for robust classification problems. Beyond the above methods, MA [28] and ACS [29] have been adopted in vision applications too.

However, since GA and MA are based on a fixed form of gene expression during the whole optimization procedure, the representations of the solution are relatively fixed and limited, which heavily influence the effectiveness in complex optimization problems. While, different from GA/MA, PSO considers the birds' social behavior and accordingly their movements toward an optimal destination rather than creating new solutions within each generation. Compared with other evolution-based methods, PSO achieves a final solution in a linear search space and tends to be relatively efficient. However, this kind of simple linear search cannot tackle complex optimization problems.

To enable more flexible representations, another evolutionary approach, i.e., GP, has been proposed [15], [30]. GP has been widely utilized in the computer vision domain and proved to be more powerful than GA. It is more intuitive for implementation and can effectively solve highly nonlinear optimization problems. Thus, in terms of obtaining better results, this kind of flexible, nonlinear searching mechanism can help GP achieve better solutions. Poli [31] applied GP to automatically select optimal filters for segmentation of the brain in medical images. Following the same line, Torres *et al.* [32] used GP for finding a combination of similarity functions for image retrieval. Davis *et al.* [33] have also employed GP for feature selection in multivariate data analysis, where GP can automatically select a subset of the most discriminative features without any prior information. In addition, other researchers [34]–[36] have also successfully applied GP to recognition tasks with improvements compared with previous methods.

Recently, GP has been exploited to assemble low-level feature detectors for high-level analysis, such as: object detection, 3D reconstruction, image tracking, and matching. The first work in this area employed GP to evolve an operator for detecting interest points [37]. Trujillo and Olague [38] have also used GP to generate feature extractors for computer vision applications. In addition, a GP-based detector was proposed by Howard *et al.* [39] for detecting ship wakes in synthetic aperture radar images.

One most related work using GP to automatically generate low-level features for action recognition is introduced in [40]. In this paper, some basic filters are successfully evolved to construct spatio-temporal descriptors for representing action sequences. Although this framework is regarded as the first attempt in using GP to learn holistic representations for action recognition, some aspects in this framework can still be improved. Specifically, the evolved structure is totally random rather than mimicking the structure of the human brain cortex with multiple tiers—this kind of random evolution may fail to find the best solutions in a limited number of generations. Furthermore, the previous work attempts to learn general-purpose representations, which tend to be less specific and discriminative for different action data domains. Lastly, is the method was only evaluated on "staged" action datasets rather than realistic action datasets. We expect to solve all these issues in this paper.

Inspired by the effectiveness of GP on flexible optimization tasks and successful applications mentioned above, in this paper, we use GP to automatically evolve more task-specific spatio-temporal descriptors from a set of 3D filters and operators for realistic human action recognition.

III. Evolutionary Motion Feature Extraction

Much of what is done in action recognition aims to achieve what the human vision system is capable of. This has caused many researchers to model systems and algorithms after various aspects of the human vision system. In this paper, we also attempt to simulate the human visual cortex system which is made up of hierarchical layers of neurons with feedforward and feedback connections that grow and evolve from birth as the vision system develops. The prior stages of processing in the visual cortex are sensitive to visual stimuli such as intensity and orientations, spatial motion, and even colors. In a similar way that feedforward neural connections between these visual cortex layers are created and evolved in humans, we propose a domain-independent machine learning methodology to automatically generate low-level spatio-temporal descriptors for high-level action recognition using GP. In our architecture, the original color and optical-flow sequences are regarded as the inputs and a group of 3D operators are assembled to construct an effective problem-specific descriptor which is capable of selectively extracting features from input data. The final evolved descriptor, combining the nice properties of those

TABLE I
FUNCTION SET IN GP

Operator Name	Input	Function Description	Operator Type
Filt i g ti			
G Py1	1 S q	Th fi t l l f G i py mid whi h ppli 3D G i filt with $\sigma = 2$ i p t q	Filt
G Py2	1 S q	Th d l l f G i py mid whi h ppli 3D G i filt with $\sigma = 2$ g i q bt i d f m fi t l l f G i py mid	Filt
G Py3	1 S q	Th thi d l l f G i py mid whi h ppli 3D G i filt with $\sigma = 2$ g i q bt i d f m d l l f G i py mid	Filt
L pPy1	1 S q	Th fi t l l f L pl i py mid whi h ppli bt ti b tw G Py1 d G Py2	Filt
L pPy2	1 S q	Th d l l f L pl i py mid whi h ppli bt ti b tw G Py2 d G Py3	Filt
W l t1	1 S q	Th fi t l l f w l t py mid whi h ppli CDF 9/7 w l t filt i p t q	Filt
W l t2	1 S q	Th d l l f w l t py mid whi h ppli CDF 9/7 w l t filt g i q bt i d f m fi t l l f w l t py mid	Filt
GBO 0	1 S q	M lti l m G b filt with i t ti f 0 d g	Filt
GBO 45	1 S q	M lti l m G b filt with i t ti f 45 d g	Filt
GBO 90	1 S q	M lti l m G b filt with i t ti f 90 d g	Filt
GBO 135	1 S q	M lti l m G b filt with i t ti f 135 d g	Filt
D f	1 S q	E ti g bt ti b tw th dj t f m f i p t q	Filt
b D f	1 q	E ti g bt ti b tw th dj t f m f i p t q d th t ki g th b l t l f th p d q	Filt
MED	1 q q	pplyi g m di filt with filt i g wi d w i 5 5 th i p t q	Filt
M	1	pplyi g m filt with filt i g wi d w i 5 5 th i p t q	Filt
S dd	2 S q	dd th i p t q	ithm ti
M b	2 S q	S bt t th i p t q	ithm ti
S	2 S q	M ltiply th i p t q	ithm ti
M p b	2 S q	b l t bt ti f th i p t q	ithm ti
g ti		i 5 5 th i p t q	
li g5 li g10 li g15 li g20		p li g with p li g wi d w 5 10 th i p t li g with p li g wi d w i 10 15 15 th i p t q li g with p li g wi d w i 15 20 th i p t q li g with p li g wi d w i 20 20	Filt Filt Filt

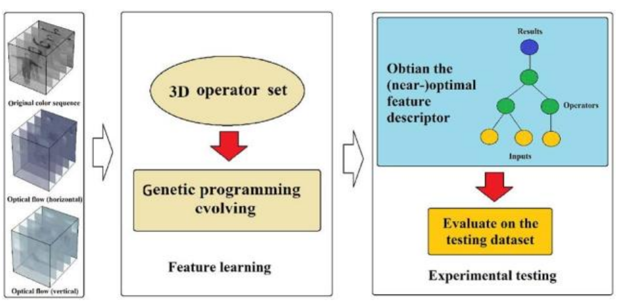


Fig. 1. Outline of our feature learning-based approach.

primitive 3D operators, can both extract meaningful features and form a compact action representation. We learn our proposed system over a training set, in which descriptors are evolved by maximizing the recognition accuracy through a fitness function, and further evaluate the GP-selected one over a testing set to demonstrate the performance of our method. The architecture of our proposed model is illustrated in Fig. 1.

Generally, GP programs can be represented as a tree structure, evolved (by selection, crossover, and mutation) through sexual reproduction with pairs of parents being chosen stochastically but biased in their fitness on the task at hand, and finally select the best performing individual as the terminal solution. In our method, each individual in GP represents a candidate spatio-temporal descriptor and is evolved continuously through generations. To establish the architecture of our model, three significant concepts: function set, terminal set, and fitness function should be first defined.

A. Function Set and Terminal Set

A key component of GP is the function set which constitutes the internal nodes of the tree and is typically driven by the nature of the problem. To make the GP evolution process fast, more efficient operators that can extract meaningful information from action sequences are preferred. Our function set consists of 19 unary operators and 4 binary ones, including processing filters and basic arithmetic functions, as illustrated in Table I.

In our GP structure, we divide our function set into two tiers: 1) filtering tier (bottom tier) and 2) max-pooling tier (top tier). The order of these tiers in our structure is always fixed. Specifically, we do not allow the filter operators in the function set to be above the max-pooling functions. In our implementation, when a descriptor is evolved, we will check whether it is a wrongly ordered descriptor or not. A wrongly ordered descriptor will be automatically discarded by our program and a new correctly-ordered descriptor would be evolved

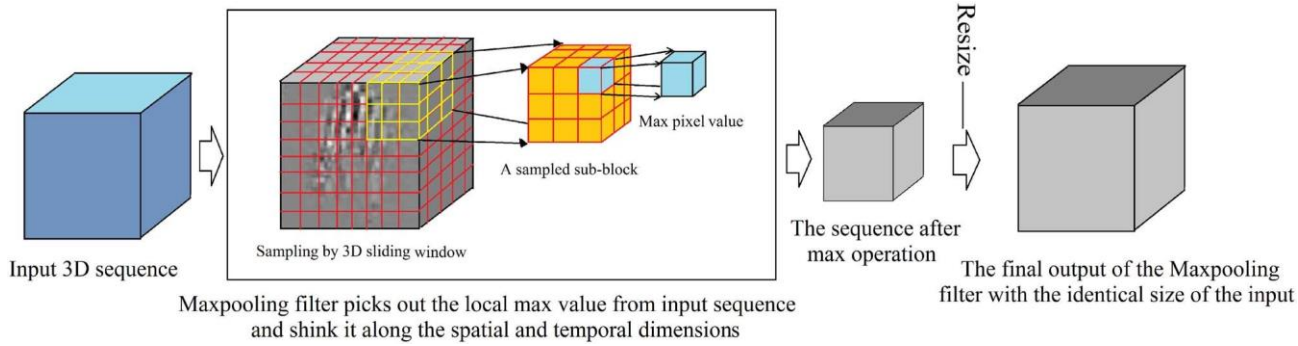


Fig. 2. Illustration of the mechanism of max-pooling filter.

to replace the discarded one. In this way, in any GP-evolved program, the operators in the filtering tier must be located below the operators in the max-pooling tier. In addition, not all the operators listed in the function set have to be used in a given tree and the same operator can be used more than once. Therefore, the topology of the tree in each tier is essentially unrestricted. This kind of tree structure makes fully use of the knowledge to mimic the physical structure of the human visual cortex [41], [42] by encoding orientation, intensity, and color information of the targets and can effectively tolerate shifting, translation, and scaling for action recognition and simultaneously reduce the GP searching space to effectively accelerate the convergence of optimal solutions.

1) Filtering Tier: In the filtering tier, aiming to extract meaningful features from dynamic actions, we adopt 3D Gaussian filters, 3D Laplacian filers, 3D Wavelet filters, 3*D Gabor filters, and some other sequence processing operators and basic arithmetic functions.

3D Gaussian filters are adopted due to their ability for denoising and 3D Laplacian filters are used for separating signals into different spectral sub-bands. Laplacian of Gaussian operators have been successfully applied to capture intensity features for action recognition in [2] and [43]. Wavelet transforms can perform multiresolution analysis and obtain the contour information of human actions by using the 3D CDF "9/7" [44] wavelet filters.

In this paper, these 3D filters are used for constructing the sequence pyramid (i.e., GauPy, LapPy, Wavelet), which is a data structure designed to support efficient scaled convolution through reducing the resolution. It consists of a sequence of copies of an original sequence in which both sampling density and resolution are decreased in regular steps. A pyramid is a multiscale representation with a recursive method. Beyond those, 3D Gabor filters are regarded as the most effective method to obtain the orientation information in a sequence. Following Riesenhuber and Poggio [41], we simulate the biological mechanism of the visual cortex to define our Gabor filter-based operators. Firstly, we convolve an input action sequence with Gabor filters at six different scales ($7 \times 7 \times 7$, $9 \times 9 \times 9$, $11 \times 11 \times 11$, $13 \times 13 \times 13$, $15 \times 15 \times 15$, and $17 \times 17 \times 17$) under a certain orientation (i.e., $0^\circ, 45^\circ, 90^\circ$, or 135°); we further apply the max operation to pick the maximum value across all six convolved sequences for that particular orientation. Fig. 3 illustrates the procedure of our

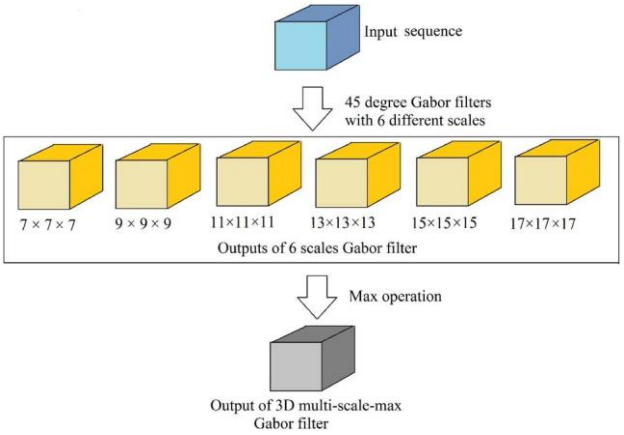


Fig. 3. Outline of multiscale-max Gabor filter. This figure illustrates an example of the multiscale-max Gabor filter with a fixed orientation of 45°.

multiscale-max Gabor filters for a certain orientation. The max operation among different scales is defined as follows:

$$I_{\mathrm{MAX}} = \max_{(x,y,z)}[I_{7\times7\times7}(x,y,z,\theta_s), I_{9\times9\times9}(x,y,z,\theta_s), \ldots, I_{15\times15\times15}(x,y,z,\theta_s), I_{17\times17\times17}(x,y,z,\theta_s)] \quad (1)$$

where I_{MAX} is the output of the multiscale-max Gabor filter. $I_{i\times i\times i}(x,y,z,\theta_s)$ denotes the convolved sequences with the scale $i \times i \times i$ and the orientation θ_s.

Moreover, several other 3D operators that are common for feature extraction are added to the function set to increase the variety of the selection for composing individuals during the GP evolution. Basic arithmetic functions are chosen to realize operations such as addition and subtraction of the internal nodes of the tree to make the whole evolution procedure more natural.

To ensure the closure property [15], we have only used functions which map one or two 3D sequences to a single 3D sequence with identical size (i.e., the input and the output of each function node have the same size). In this way, a GP tree can be an unrestricted composition of function nodes but still always produce a semantically legal structure.

2) Max-Pooling Tier: In the max-pooling tier, we include four functions listed in Table I, which are performed over local neighborhoods with windows varying from $5 \times 5 \times 5$ to $20 \times 20 \times 20$ with a shifting step of 5 pixels. This max-pooling operation (see Fig. 2) is a key mechanism for object

recognition in the cortex and provides a more robust response, successfully tolerating shift and scaling, in the case of recognition in clutter or with multiple stimuli in the receptive field [41]. Given a sequence, max-pooling functions will pick out the local max values from the input and shrink it along spatial and temporal dimensions to compose a more compact representation of the input sequence. We further resize outputs calculated from max-pooling functions to an identical size as inputs using linear interpolation [45]. In this way, the sizes of inputs and outputs of our max-pooling functions are the same.

Each function in the function set is regarded as a tree node in evolved programs, which connects the outputs of lower level functions or inputs. Note that, in our proposed GP architecture, not all the functions listed in the function set have to be used in a given structure and the same function can be used more than once. The topology of our GP structure is essentially unrestricted. Besides, functions in the function set are also highly related to our problem domain. For this paper, we aim to construct novel discriminative feature descriptors for action recognition. So, what we choose in the function set are effective filters for feature extraction, i.e., 3D Gaussian filters, 3D Gabor filters, 3D Laplacian filters, etc. We expect the whole learned architecture is consistent with the physical structure of the human visual cortex.

3) Terminal Set: In addition, the terminal set is also a significant component of GP. For action recognition, we consider the following aspects of our task: 1) the terminal set must capture the holistic information of each action sequence and 2) during the evolution process, the evaluation of the fitness function must be efficient. In our implementation, to simulate the human visual cognition system, which makes decisions relying on both color and motion information of moving objects from a viewpoint, we expect to obtain informative spatio-temporal features by fusing original color and motion information. Particularly, the motion information is extracted by computing the optical flow [46] along horizontal and ver- tical directions. Given an original action sequence, the optical flow maps: F_x and F_y are computed between adjacent tth frame and $(t + 1)$th frame. The final optical flow data are further obtained by piling the F_x and F_y into sequences V_{F_x} and V_{F_y}. We further normalize all data in the terminal set into the same size by applying bicubic interpolation [47]. In addition, the terminal of the GP structure is data-dependent, which means that each video sequence V_i from the training set has a corresponding T_i defined by: $T_i = \{V_{\text{color}}, V_{F_x}, \text{and } V_{F_y}\}$ (See Fig. 4).

Note that, in our implementation, a color sequence is merely a grayscale intensity sequence, as we believe that the use of three colors (red, green, and blue) will not bring much extra information for action recognition. For each tree-based genetic structure, an action sequence is located at the bottom leaf of the entire tree and connects with the higher function nodes directly. The data types used for the terminal of program nodes are listed in Table II.

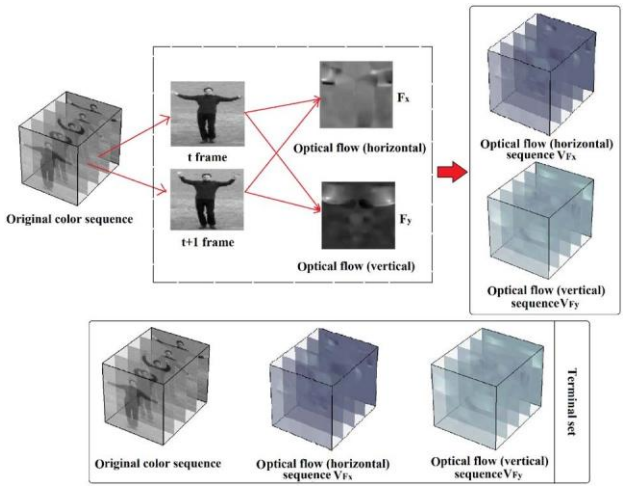


Fig. 4. Example of input data types in the terminal set.

TABLE II
STATEMENT OF TERMINAL NODES

Terminal	Type	Description
V_I	S q	O igi l l q
V_F	S q	Opti l fl w q l g h i t l di ti
V_F	S q	Opti l fl w q l g ti l di ti

B. Fitness Function

The objective of evolutionary methods is to maximize the performance of individual solutions as gauged by some appropriate fitness functions. To evaluate the candidate GP-evolved feature descriptors here, we estimate their recognition accuracies using a linear support-vector-machine (SVM),[1] which is a popular classifier for computer vision tasks because of its high accuracy and efficiency. To avoid the redundancy and make a more compact feature representation, we take the $m \times n \times t$ output of the GP tree and divide it into $10 \times 10 \times 5$ sub-blocks.[2] The mean values of each sub-block are concatenated into a 500D vector which comprises the input of a SVM, as shown in Fig. 6. To obtain a more reliable fitness evaluation, for each new GP tree we estimate the recognition accuracy with the SVM using ten-fold cross-validation. We divide the GP training set randomly into ten equal parts and perform ten repetitions of training the SVM on nine-tenths of the set and testing on the remaining tenth. The overall fitness of the candidate GP tree is taken as the average of the ten SVM test-fold accuracies. The corresponding fitness function is defined as follows:

$$E_r = \left(1 - \sum_{i=1}^{n} (\text{SVM}[\text{acu}_i])/n\right) \times 100\% \qquad (2)$$

where $\text{SVM}[\text{acu}_i]$ denotes the recognition accuracy of fold i by the SVM and n indicates the total number of folds executed with cross-validation. Here n is equal to 10.

[1] The classifier used during descriptor learning and at the testing stage should be consistent.

[2] In our experiments, we selected the optimal size of the sub-blocks from the set of $\{5 \times 5 \times 5, 10 \times 5 \times 5, 10 \times 10 \times 5, 10 \times 10 \times 10, \text{and } 15 \times 15 \times 10\}$ by 5-fold cross-validation. The relevant results show $10 \times 5 \times 5$ and $10 \times 10 \times 5$ both achieve better performance for final classification. Due to the consideration of complexity, we chose $10 \times 10 \times 5$ in our experiments.

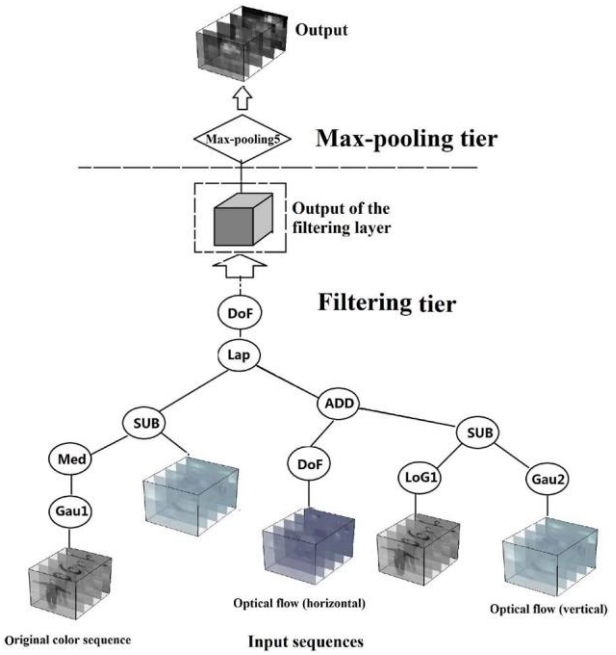


Fig. 5. Illustration of a possible learned-structure. All the 3D operators and pooling functions used in the structure are randomly selected through the GP evolution.

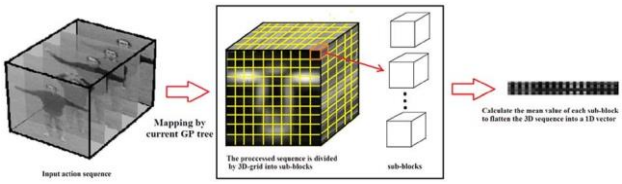


Fig. 6. Procedure of spatio-temporal sequence representation.

In our method, each training sample is a video sequence containing a large number of pixels and the fitness function has to be evaluated over the training set many times for whole population in each GP generation. Meanwhile, for getting good results, a large number of generations are usually required. All these will lead to heavy computation. To speed-up the GP learning algorithm, we used multiple processors which can evaluate many fitness measures at the same time, giving a tremendous reduction in the training time.

C. Genetic Programming Framework

GP [15] is one of a number of population-based evolutionary algorithms inspired by natural evolution and is widely used in machine learning. It allows a computer to automatically solve predefined tasks without requiring users to know or specify the form or structure of the solution in advance. In GP, we randomly generate an initial population of operation sequences which are regarded as candidate solutions. This population is then allowed to evolve (by selection, crossover, and mutation) through sexual reproduction with pairs of parents being chosen stochastically but biased in their fitness on the task at hand. In this way, the general fitness of the population tends to improve over time. Finally, the best performing individual obtained is taken as the final solution. It should be noted that

Algorithm 1 Genetic Programming

```
Start
  Initialization
  for size of population do
  Randomly create an initial population of operation sequences from the
  available primitives (terminal set & function set)
  end for
  for number of generations do
    for each individual do
      (1) Process action sequences with evolved individual feature descriptor
      (2) Evaluate the fitness of the individual via recognition error rate
      (3) Choose individuals from the population with a particular probability biased

      in their fitness
      (4) Create a new generation of individuals applying genetic operations
      (crossover & mutation)
    If  An acceptable solution is found or the maximum number of generations
     (defined by user) exceeded
    end for
  end for
  Return  The best feature descriptor is selected
end for
```

evolutionary methods do not guarantee to find any mathematical optimum, but, in practice, usually find a good solution to an NP-hard problem in an acceptable amount of computer time. The relevant GP algorithm is shown in Algorithm 1.

The action sequences form the terminal set and each sequence with the size of $m \times n \times t$ is taken, in turn, as the input to a GP individual. Each GP candidate feature descriptor is formulated as a tree structure, the output of which is still a $m \times n \times t$ block. A representative GP tree is illustrated in Fig. 5.

IV. Experiments and Results

In this section, we describe the details of our GP implementation and the relevant experimental results we obtain by our approach.

A. GP Implementation

We implement our proposed method using MATLAB 2011a (with the GP toolbox GPLAB[3]) on a server with a 12-core processor and 54GB of RAM running the Linux operating system. The total runtime was around three weeks. The user-defined GP parameters are as follows.

1) Population Size: According to some previous relevant experiments, the larger population we define in GP running, the better solution we can potentially obtain. In this case, considering the high computational cost, we set a population size of 200 individuals with the initial population generated with the ramped half-and-half method [15]. The number of generations is defined as 70.

2) Genetic Operators: We use both tree crossover and mutation [15] as our genetic operators. Following the standard setting in [15], we fix their probabilities during the GP evolution at 90% and 10%, respectively.

3) Selection for Reproduction: The selection method we apply is lexicographic parsimony pressure [48] which is similar to tournament selection in choosing parents from a random sub-set of individuals in the population. However, the unique feature of lexicographic parsimony pressure is that the smallest individual (i.e., fewest tree nodes) will be selected if more

[3]http://gplab.sourceforge.net/download.html, a GP Toolbox for MATLAB.

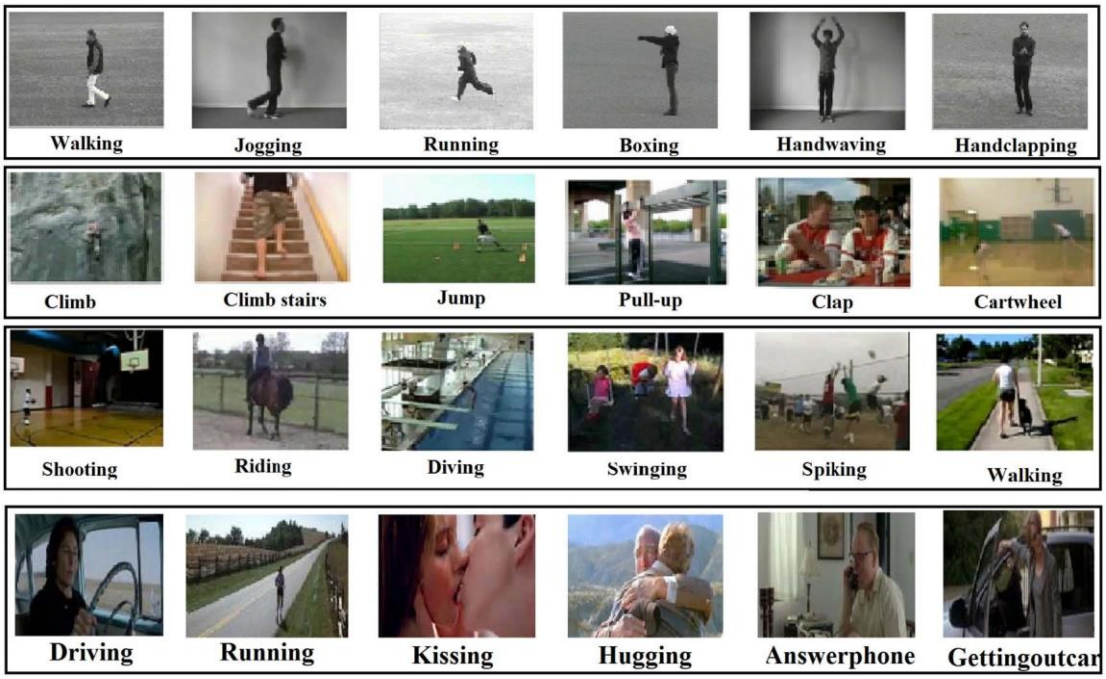


Fig. 7. Some example frames of four datasets. Images in the top row are from the KTH dataset, images in the second row are from the HMDB51 dataset, images in the third row are from the YouTube dataset, and images in the bottom row are from the Hollywood2 dataset.

than one individual has the same best fitness in the selection competition.

4) Survival Method: We adopt the "total elitism" scheme for GP running. In this scheme, all the individuals from both parents and children populations are ordered by fitness alone, regardless of being parents or children. Consequently, the best individuals can be kept and inherited generation by generation. This scheme has been demonstrated leading to promising results in many applications.

5) Stopping Conditions: We set the GP termination criterion as the error rate falling to $\leq$ 2% or the number of generations exceeding 70.

As GP is a stochastic approach, we run our method three times on each dataset and select the best performing descriptor. Once a descriptor is learned and selected, its structure is fixed and can be used on new data the same as a hand-crafted descriptor.

B. Datasets

We systematically test our proposed method on four popular action datasets: KTH [49], HMDB51 [50] YouTube [9], and Hollywood2 [51]. Some example frames from these four datasets are visualized in Fig. 7.

The KTH dataset is a commonly used benchmark action dataset with 599 video clips. Six human action classes, including walking, jogging, running, boxing, hand-waving, and handclapping, are performed by 25 subjects in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3), and indoors with lighting variation (s4). Following the preprocessing step mentioned in [52], the coarse 3D bounding boxes are extracted from all the raw action sequences and further normalized into an equal size of $100 \times 100 \times 60$. We follow the original experimental setting of the authors, i.e., divide the data into a test set (9 sujects: 2, 3, 5, 6, 7, 8, 9, 10, 22 for final testing) and a training set (the remaining 16 subjects for GP training). As in [49], we train and evaluate a multiclass classifier and report the average accuracy over all classes as the performance measure.

The HMDB51 dataset collects 6849 action sequences from various movies and online videos. In our case, we adopt 2241 sequences from 19 general body action categories (i.e., cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, and wave) as our research data. In our experiments, coarse bounding boxes have been extracted from all the sequences through masks released with the dataset and initialized into the size of $100 \times 120 \times 50$. We further randomly divide these 2241 sequences into three subsets and adopt the first two subsets as the training set and the rest as the testing set.

The YouTube dataset contains 1168 video sequences collected from 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. This dataset is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, and illumination conditions. For this dataset, we deliberately use the full-sized sequences without any bounding boxes as the input to evaluate our method's robustness against complex and noisy backgrounds. Each sequence is further normalized into the size of $100 \times 100 \times 60$. We take the first 2/3 sequences from each category to compose our training set, and the rest of the data is defined as the testing set.

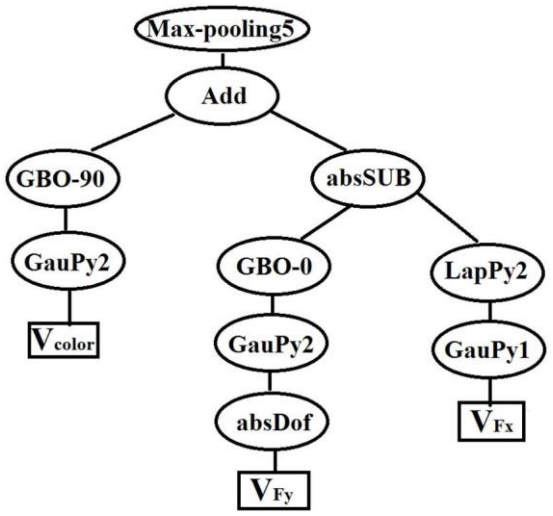


Fig. 8. Near-optimal feature descriptor generated through GP on the KTH dataset.

The Hollywood2 dataset has collected 1707 action samples from 69 different Hollywood movies with 12 action classes: answering the phone, driving car, eating, fighting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up. To meet the closure requirement of GP, we further resize all sequences in this dataset to the identical size $100 \times 100 \times 80$. In our experiments, we utilize our GP method on a training set with 823 sequences and a test set with 884 sequences following the original paper.

C. Results

For the KTH dataset, our approach automatically selects and fuses color-motion information and generates machine-learned descriptors for action recognition. We select the best GP-evolved feature descriptor to represent all the action sequences and train a linear-SVM classifier on the training set and test on the remaining following the experimental setting in [49]. We finally achieve the recognition accuracy of 95% on the testing set. Compared with other results listed in Table III, we can easily conclude that our result is comparable to [53] and significantly outperforms other methods. Note that using the "leave-one-out" experimental setting should yield higher accuracies than using the "split" setting as mentioned in [49]. Fig. 8 shows the tree structure of the best-performed GP program(feature descriptor), in which 3D Gaussian, 3D Laplacian, and 3D Gabor operators were automatically selected by GP evolving at the filtering layer to extract the orientation and intensity features, and several scales of pooling operators can get the most robust and distinctive responses to different data resolutions on the top layer. The whole learned architecture is indeed consistent with the physical structure of the human visual cortex. In addition, the detailed information of error rate during the genetic evolution can be seen in Fig. 12. Beyond those, as illustrated in Table IV and Fig. 11, we have further evaluated our method with different classifiers in the fitness function to prove that our evolved system, which comprises the evolved descriptor and the used classifier, gives better classification performance

TABLE III
COMPARISON OF ACTION RECOGNITION ACCURACIES IN PERCENTAGE (%) ON THE KTH DATASET WITH DIFFERENT METHODS

Methods \ Accuracy	Experimental setting	Recognition rate
our method	Split	**95.0**
Dense HOG3D [ ]	Split	92.7
Dense HOG/HOF [8]	Split	92.3
HOF [4]	Split	92.0
HMHI [55]	Split	90
Dense SURF3D [5]	Split	89.3
Motion and Structure Features [43]	Split	92.7
3D Gabor based	Split	91.7
CNN [57]	Split	93
DBN [22]	Split	94.2
Liu *et al.* [40]	Split	93.5
Schindler and van Gool [58]	Split	92.7
Wang *et al.* [59]	Split	92.1
Laptev *et al.* [8]	Split	91.8
Jhuang *et al.* [0]	Split	91.7
Klaser *et al.* [1]	Split	91.4
Fathi and Mori [2]	Split	90.5
Nowozin *et al.* [3]	Split	87.04
Schuldt *et al.* [53]	Split	71.71
Ke *et al.* [4]	Split	2.79
Chen *et al.* [54]	Leave one out	
Liu and Shah [5]	Leave one out	
Niebles *et al.* [ ]	Leave one out	
Dollar *et al.* [4]	Leave one out	

TABLE IV
CLASSIFICATION PERFORMANCE OF OUR GP-BASED TECHNIQUE USING DIFFERENT CLASSIFIERS IN THE FITNESS FUNCTION ON THE KTH DATASET (WITH THE SPLITTING SETTING)

Methods \ Accuracy	Classifier	Recognition rate
Gabor based	SVM	95.0%
Gabor based	Nearest Neighbor	87.2%
Gabor based	Naive Bayes (Gaussian distribution)	85.5%
Dense HOG3D	SVM	92.7%
Dense HOG3D	Nearest Neighbor	82.9%
Dense HOG3D	Naive Bayes (Gaussian distribution)	80 %

independently of the selected classifier compared with the hand-crafted feature descriptors. In addition, Table IV shows that a more powerful classifier such as SVM will lead to better performance of the system consisting of the evolved descriptor and the adopted classifier.

The HMDB51 dataset is one of the most complex datasets for action recognition. In our experiments, the proposed method still works well to assemble a (near-)optimized feature descriptor by using GP. The LISP format of the evolved descriptor is shown in Fig. 9. Combining with the linear-SVM classifier, the GP-evolved descriptor achieves excellent performance on these action sequences with noisy and complex backgrounds. As a result, the obtained best feature descriptor achieves the final recognition accuracy rate of 48.4% on the testing set. Due to different experimental settings and a different focus of attention, we only compare with state-of-arts

```
Max-pooling5(absSUB(SUB(SUB(GauPy2
(Vcolor),GauPy2(Vcolor)),MED(Wavelet2
(GauPy2(Vcolor)))),Add(GBO-90(LapPy1(Mean
(VFx))),GBO-135(LapPy2(Dof(VFy))))))
```

Fig. 9. LISP format of the (near-)optimal feature descriptor generated through GP on the HMDB51 dataset.

```
Max-pooling10(SUB(absSUB(GBO-45
(Dof(LapPy1(Vcolor))),GauPy2
(VFx)),GBO-90(LapPy2
(Vcolor))))
```

Fig. 10. LISP format of the (near-)optimal feature descriptor generated through GP on the YouTube dataset.

```
Maxpooling10(MED(ADD(LapPy2(Wave
let2(absDof(LapPy2(VFx)))),SUB(M
ean(Vcolor),GauPy2(MED(GauPy1(Vcolo
r)))))))
```

GP+Nearest Neighbor

```
Maxpooling5(GauPy1(SUB(GBO-90(Ga
uPy1(Vcolor)),GBO-0(Dof(GauPy2(Ga
uPy1(VFy)))))))
```

GP+Naive Bayes

Fig. 11. LISP format of the (near-)optimal feature descriptors generated through GP with nearest neighbor classifier and naive Bayes classifier, respectively in fitness function on the KTH dataset.

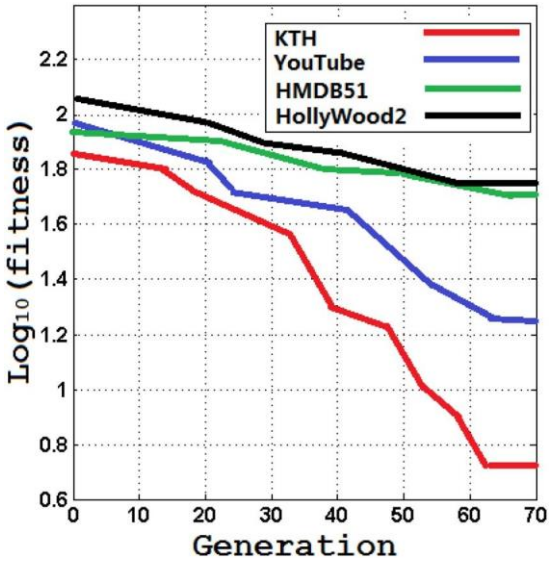


Fig. 12. Evolved best-so-far values of fitness on four datasets.

handcrafted feature descriptors rather than other action recognition systems. The relevant results are shown in Table V, from which, our method shows higher performance than other handcraft and machine learned features.

The results on the YouTube dataset are shown in Table VI. As expected, the best GP-evolved feature descriptor achieves a recognition accuracy rate of 82.3% on the testing set using the SVM classifier, since this collection represents a natural pool

TABLE V
COMPARISON OF ACTION RECOGNITION ACCURACIES IN PERCENTAGE (%) ON THE HMDB51 DATASET WITH DIFFERENT METHODS

Accuracy / Methods	Experiment setting	Recognition rate
our method	Split	**48.4**
Dense HOG3D [ ]	Split	42.7
Dense HOG/HOF [8]	Split	43.5
HOF [4]	Split	42.2
HMHI [55]	Split	41.5
Dense SURF3D [5]	Split	39.8
Motion nd Structure Features [43]	Split	45.2
3D G bor b nk	Split	44.0
CNN [57]	Split	45.8
DBN [22]	Split	4 9

TABLE VI
COMPARISON OF ACTION RECOGNITION ACCURACIES IN PERCENTAGE (%) ON THE YOUTUBE DATASET WITH DIFFERENT METHODS

Accuracy / Methods	Experiment setting	Recognition rate
our method	Split	**82.3**
Dense HOG3D [ ]	Split	7 4
Dense HOG/HOF [8]	Split	7 0
HOF [4]	Split	74.7
HMHI [55]	Split	72
Dense SURF3D [5]	Split	72.3
Motion nd Structure Features [43]	Split	77.7
3D G bor b nk	Split	75.8
3DCNN [57]	Split	79.8
DBN [22]	Split	82.1

TABLE VII
COMPARISON OF ACTION RECOGNITION ACCURACIES IN PERCENTAGE (%) ON THE HOLLYWOOD2 DATASET WITH DIFFERENT METHODS

Accuracy / Methods	Experiment setting	Recognition rate
our method	Split	**46.8**
Dense HOG3D [ ]	Split	43.7
Dense HOG/HOF [8]	Split	44.8
HOF [4]	Split	42.5
HMHI [55]	Split	40.8
Dense SURF3D [5]	Split	42.0
Motion nd Structure Features [43]	Split	43.3
3D G bor b nk	Split	42.2
CNN [57]	Split	45.3
DBN [22]	Split	**46.8**

of actions featured in a wide range of scenes and viewpoints with large intraclass variability. Fig. 10 shows the LISP format of the corresponding GP program. Note here, due to the computational cost of GP, we cannot do cross-validation following the original experimental setting. All the comparable results are calculated under our data division. From Table VI, it is obvious that our GP-evolved motion feature is competitive with the DBN-learned one but significantly outperforms other features.

To demonstrate the generalizability of the proposed method, we evaluate it on the Hollywood2 dataset as well. Since the actions of the Hollywood2 dataset are collected from films

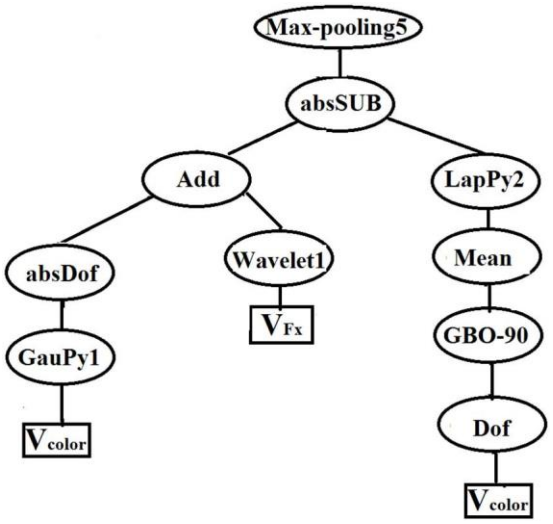


Fig. 13. Near-optimal feature descriptor generated through GP on the Hollywood2 dataset.

presenting realistic scenarios, the results shown in Table VII on this dataset are not as promising as those on other datasets. Apparently, our GP-evolved method has consistently achieved significantly better results (46.8%) than other hand-crafted and is competitive with CNN and DBN. The evolved tree-based structure can be found in Fig. 13.

For comparison, we also list the recognition rates calculated on all datasets by some prevalent hand-crafted 3D descriptors including: hierarchical MHI (HMHI), dense-HOG/HOF, dense-3DHOG, 3D-SURF, HOF, motion and structure features (MSF) [43], and 3D-Gabor-bank. Under the same experimental setting, we use the HMHI as a holistic 3D descriptor to extract the motion information for later recognition. 3D-Gabor-bank, which is considered as an effective and efficient way to obtain the orientation information, simulates the biological mechanism of the human visual cortex by applying 3D Gabor filtering with 4 orientations at 6 different scales. The output of each filter is then averaged on a $10 \times 10 \times 10$ grid to form a vector. Additionally, MSF encodes one motion plane and five image planes to capture the structure information. The Gaussian pyramid and center-surround operations are performed on each of the six obtained feature maps, decomposing each feature map into a set of sub-band maps, on which biologically inspired features are then extracted. As the other 3D descriptors are usually used as local descriptors, dense sampling is first applied on each sequence in a dense grid with the block size of $10 \times 10 \times 10$ pixels and an overlap of 5 pixels in each dimension, and the final representation vector is the concatenation of the descriptor calculated on all blocks. For fair comparison, all the above features are respectively extracted from the original sequence and the optical flow sequences, and then concatenated into a long representation which is fed to a linear SVM.

In addition, we have also utilized two popular deep learning methods, i.e., DBN [22] and CNN [56], to learn hierarchical architectures for feature extraction on the combined learning and evaluation sets. For DBN, we train a hierarchical architecture on the training sets with neuron numbers in the hidden layers: $500 - 500 - 2000$ with backpropagation fine-tuning and then utilize the learned architecture (with associated parameters) to extract features on the test sets combined with the linear SVM classifier for recogni- tion. Similarly, a 5-layer feature extraction structure has been trained using the CNN and further adopt the same recognition mechanism to compute the final accuracy. In our experiments, we use DeepLearnToolbox[4] with default parameter settings according to previous publications by Hinton *et al.* [22], to implement relevant tasks. To make the comparison fair, all the sequences used as the inputs of the architectures are the combinations of the color and optical flow components of the original sequence.

TABLE VIII
TIME COSTS OF FEATURE LEARNING ON THE FOUR DATASETS WE USED BY THE PROPOSED GP METHOD (MATLAB 2011A IS USED FOR CODING)

Time cost / Dataset	Training Phase (hours)	Testing Phase (seconds)
KTH	87 8h	0 32s per sequence
HMDB51	152 1h	0 51s per sequence
YouTube	124 6h	0 37s per sequence
Hollywood2	139 6h	0 45s per sequence
A erage	126 0h	0 41s per sequence

Additionally, to illustrate time complexity of the feature learning process, we show the evolving time costs of the method on four datasets in Table VIII. For video datasets, each training sample would be very large size. However, the fitness function must be evaluated over all the training set many times for all populations within one GP generation. Meanwhile, to getting good results, a large number of generations are usually required, which leads to heavy computation. In our experi- ments, we actually implement parallel processing to speed-up the GP learning algorithm. In our implementation, the large number of fitness evaluation can be performed by multiple processors at the same time, giving a tremendous reduction in the training time.

As many other learning algorithms, the training of the descriptors is time-consuming, but it can be performed offline. Once the optimal descriptor is obtained from the GP training phase, the classification phase will be very efficient, as the optimized descriptor can be just used as a handcrafted descriptor. Of course, with the rapid development of silicon technologies, future computers will be much faster and even the training will become less a problem.

In this paper, we aim to introduce a novel adaptive method to learn discriminative descriptors. Our GP-learned solutions are just descriptors like SIFT. So, we mainly compare our GP-evolved descriptors with other state-of-the-art descriptors rather than a whole action recognition system composed with different feature descriptors and classifiers. As our contribution is the learning of features, comparing with other handcrafted and learned features using the same classifier with exactly the same setting is the fairest way. If combining our GP-learned motion features with more advanced classification models, it is possible to reach higher recognition results on these datasets,

[4]https://github.com/rasmusbergpalm/DeepLearnToolbox

but it is not the core of this paper. So, for the HMDB51, YouTube, and Hollywood2 datasets, it is meaningless to compare with other entire recognition systems under different experimental settings.

V. Conclusion

In this paper, we have developed an adaptive learning methodology using GP to evolve discriminative spatiotemporal representations, which simultaneously fuse the color and motion information, for high-level action recognition tasks. Our method addresses feature learning as an optimization problem, and allows a computer to automatically assemble holistic feature extraction by using a pool of primitive operators, which are devised according to the general knowledge of feature extraction. We have systematically evaluated our method on four public datasets: KTH, HMDB51, YouTube, and Hollywood2 with accuracies of 95%, 48.4%, 82.3%, and 46.8% using the learned descriptors. In all four datasets, experimental results manifest that our GP feature learning approach achieves significantly higher recognition performance compared with state-of-the-art hand-crafted and machine-learned techniques. In future work, we will mainly focus on the parallel and GPU computation to speed-up our methods. Besides, other more recent evolutionary methods (e.g., PSO) will be taken into consideration for leaning discriminative features.

References

[1] L. Liu, L. Shao, X. Zhen, and X. Li, "Learning discriminative key poses for action recognition," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1860–1870, Dec. 2013.

[2] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 817–827, Jul. 2013.

[3] L. Liu, L. Shao, F. Zheng, and X. Li, "Realistic action recognition via sparsely-constructed Gaussian processes," *Pattern Recognit.*, vol. 47, no. 12, pp. 3819–3827, 2014.

[4] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recogni- tion via sparse spatio-temporal features," in *Proc. IEEE Int. Workshop Vis. Surveill. Perform. Eval. Track. Surveill.*, Beijing, China, 2005, pp. 65–72.

[5] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2, pp. 107–123, 2005.

[6] A. Klaser and M. Marszalek, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. 19th Brit. Mach. Vis. Conf.*, Leeds, U.K., 2008, pp. 995–1004.

[7] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia*, Augsburg, Germany, 2007, pp. 357–360.

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.

[9] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, 2009, pp. 1–8.

[10] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[11] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[12] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Heraklion, Greece, 2010, pp. 140–153.

[13] J. Han *et al.*, "An object-oriented visual saliency detection framework based on sparse coding representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 12, pp. 2009–2021, Dec. 2013.

[14] J. Han *et al.*, "Representing and retrieving video shots in human-centric brain imaging space," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2723–2736, Jul. 2013.

[15] R. Poli, W. Langdon, and N. McPhee, *A Field Guide to Genetic Programming*. Morrisville, NC, USA: Lulu Press, 2008.

[16] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proc. IEEE Conf. Comput. Vis.*, Nice, France, 2003, pp. 726–733.

[17] K. Schindler and L. Van Gool, "Action snippets: How many frames does human action recognition require?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.

[18] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[19] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 1810–1818, Jul. 2012.

[20] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[21] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 3361–3368.

[22] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[23] J. Han *et al.*, "Background prior based salient object detection via deep reconstruction residual," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[24] B. Bhanu, S. Lee, and J. Ming, "Adaptive image segmentation using a genetic algorithm," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 12, pp. 1543–1567, Dec. 1995.

[25] A. W. Mohemmed, M. Zhang, and M. Johnston, "Particle swarm optimization based AdaBoost for face detection," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 2494–2501.

[26] M. Omran, A. Salman, and A. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in unsupervised image classification," in *Proc. 5th World Enformatika Conf. (ICCI)*, Irvine, CA, USA, 2005, pp. 199–204.

[27] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[28] V. Di Gesù, G. L. Bosco, F. Millonzi, and C. Valenti, "A memetic algorithm for binary image reconstruction," in *Combinatorial Image Analysis*. Berlin, Germany: Springer, 2008, pp. 384–395.

[29] D. Dolkar and B. Saha, "Optimal face recognition method using ant colony based back propagation network," in *Proc. Int. Conf. Comput. Devices Commun.*, Kolkata, India, 2009, pp. 1–4.

[30] J. Koza and R. Poli, "Genetic programming," *Search Methodologies*. New York, NY, USA, 2005, pp. 127–164.

[31] R. Poli, "Genetic programming for image analysis," in *Proc. 1st Annu. Conf. Genet. Program.*, Stanford, CA, USA, 1996, pp. 363–368.

[32] R. Torres *et al.*, "A genetic programming framework for content-based image retrieval," *Pattern Recognit.*, vol. 42, no. 2, pp. 283–292, 2009.

[33] R. Davis, A. Charlton, S. Oehlschlager, and J. Wilson, "Novel feature selection method for genetic programming using metabolomic HNMR data," *Chemometr. Intell. Lab. Syst.*, vol. 81, no. 1, pp. 50–59, 2006.

[34] J. Kishore, L. Patnaik, V. Mani, and V. Agrawal, "Application of genetic programming for multicategory pattern classification," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 242–258, Sep. 2000.

[35] M. Zhang and W. Smart, "Multiclass object classification using genetic programming," *Appl. Evol. Comput.*, vol. 14, no. 3, pp. 369–378, 2004.

[36] H. Guo, L. Jack, and A. Nandi, "Feature generation using genetic programming with application to fault classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 1, pp. 89–99, Feb. 2005.

[37] M. Ebner and A. Zell, "Evolving a task specific image operator," *Evol. Image Anal. Signal Process. Telecommun.*, vol. 14, no. 7, pp. 74–89, 1999.

[38] L. Trujillo and G. Olague, "Synthesis of interest point detectors through genetic programming," in *Proc. 8th Annu. Conf. Genet. Evol. Comput.*, Seattle, WA, USA, 2006, pp. 887–894.

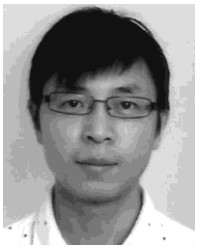
[39] D. Howard, S. Roberts, and R. Brankin, "Target detection in SAR imagery by genetic programming," *Adv. Eng. Softw.*, vol. 30, no. 5, pp. 303–311, 1999.

[40] L. Liu, L. Shao, and P. Rockett, "Genetic programming-evolved spatio-temporal descriptor for human action recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Surrey, U.K., 2012.

[41] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nat. Neurosci.*, vol. 2, no. 11, pp. 1019–1025, 1999.
[42] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 300–312, Feb. 2007.
[43] X. Zhen, L. Shao, D. Tao, and X. Li, "Embedding motion and structure features for action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1182–1190, Jul. 2013.
[44] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, 1992.
[45] E. Meijering, "A chronology of interpolation: From ancient astronomy to modern signal and image processing," *Proc. IEEE*, vol. 90, no. 3, pp. 319–342, Mar. 2002.
[46] B. Lukas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA Image Und. Workshop*, Vancouver, BC, Canada, 1981, pp. 674–679.
[47] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
[48] S. Luke and L. Panait, "Lexicographic parsimony pressure," in *Proc. Genet. Evol. Comput. Conf.*, New York, NY, USA, 2002, pp. 829–836.
[49] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. IEEE Conf. Pattern Recognit.*, vol. 3. Cambridge, U.K., 2004, pp. 32–36.
[50] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 2556–2563.
[51] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, 2009, pp. 2929–2936.
[52] A. Yao, J. Gall, and L. Van Gool, "A Hough transform-based voting framework for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 2061–2068.
[53] M. Chen, L. Mummert, P. Pillai, A. Hauptmann, and R. Sukthankar, "Exploiting multi-level parallelism for low-latency activity recognition in streaming video," in *Proc. 1st Annu. ACM SIGMM Conf. Multimedia Syst.*, Phoenix, AZ, USA, 2010, pp. 1–12.
[54] J. Davis, "Hierarchical motion history images for recognizing human motion," in *Proc. IEEE Workshop Detect. Recognit. Events Video*, Vancouver, BC, Canada, 2001, pp. 39–46.
[55] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Und.*, vol. 110, no. 3, pp. 346–359, 2008.
[56] M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, 2007, pp. 1–8.
[57] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, London, U.K., 2009.
[58] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. IEEE Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–8.
[59] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.
[60] S. Nowozin, G. Bakir, and K. Tsuda, "Discriminative subsequence mining for action classification," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–8.
[61] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Beijing, China, 2005, pp. 166–173.
[62] J. Liu and M. Shah, "Learning human actions via information maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.
[63] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vis.*, vol. 79, no. 3, pp. 299–318, 2008.

Li Liu received the B.Eng. degree in electronic information engineering from Xi'an Jiaotong University, Xi'an, China, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K., in 2011 and 2014, respectively.

He is currently a Research Fellow with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne, U.K. His current research interests include computer vision, machine learning, and data mining.



Ling Shao (M'09–SM'10) is a Professor with the Department of Computer Science and Digital Technologies at Northumbria University, Newcastle upon Tyne, U.K., and a Guest Professor with the Nanjing University of Information Science and Technology, China. Previously, he was a Senior Lecturer (2009–2014) with the Department of Electronic and Electrical Engineering at the University of Sheffield and a Senior Scientist (2005–2009) with Philips Research, The Netherlands. His research interests include Computer Vision, Image/Video Processing and Machine Learning. He is an associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CYBERNETICS, and several other journals. He is a Fellow of the British Computer Society and the Institution of Engineering and Technology

Xuelong Li (M'02–SM'07–F'12) is a Full Professor with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

Ke Lu was born in Ningxia, China, in 1971. He received the master's and Ph.D. degrees from the Department of Mathematics and Department of Computer Science, Northwest University, Kirkland, WA, USA, in 1998 and 2003, respectively.

He was a Post-Doctoral Fellow at the Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 2003 to 2005. He is currently a Professor with the University of the Chinese Academy of Sciences, Beijing. His current research interests include computer vision, 3D image reconstruction, and computer graphics.