Long-Short-Term Features for Dynamic Scene Classification

Yuanjun Huang, Xianbin Cao, *Senior Member, IEEE*, Qi Wang, *Senior Member, IEEE*, Baochang Zhang, Xiantong Zhen, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Dynamic scene classification has been extensively studied in computer vision due to its widespread applications. The key to dynamic scene classification lies in jointly characterizing spatial appearance and temporal dynamics to achieve informative representation, which remains an outstanding task in the literature. In this paper, we propose a unified framework to extract spatial and temporal features for dynamic scene representation. More specifically, we deploy two variants of deep convolutional neural networks to encode spatial appearance and short-term dynamics into short-term deep features (STDF). Based on STDF, we propose using the autoregressive moving average model to extract long-term frequency features (LTFF). By combining STDF and LTFF, we establish the long–short-term feature (LSTF) representations of dynamic scenes. The LSTF characterizes both spatial and temporal patterns of dynamic scenes for comprehensive and information representation that enables more accurate classification. Extensive experiments on three-dynamic scene classification benchmarks have shown that the proposed LSTF achieves high performance and substantially surpasses the state-of-the-art methods.**

***Index Terms*— Dynamic scene classification, long-short term feature, long term frequency feature.**

I. Introduction

DYNAMIC scene classification has been an active research area in computer vision because of its widespread applications in scene interpretation, traffic surveillance, and video content analysis [1]–[4]. Moreover, the significance of scene classification also stems from its fundamentally important roles in solving more challenging tasks. For instance, to find ground vehicles by the unmanned aerial vehicle (UAV), we first recognize the highway scenes, wherein the vehicle targets are appearing with high possibility, and then the target detection process can be accelerated.

Compared with the image-based scene classification [5], [6], dynamic scene classification in videos is more challenging due to the high complexity in extracting spatial and temporal features from videos. In the past decades, it has stimulated a vast amount of research works including SOE [7], SFA [8], and CSR [9], etc. However, it remains an outstanding tasks and the challenges are mainly in two folds. On the one hand, it has to deal with the difficulties shared in still image classification e.g., large variations in illumination, viewpoints and scales. As shown in Fig. 1, exemplar shots in the same category have large variations, making it very difficult for classification. On the other hand, by the nature of dynamic scenes, the addition of temporal dynamics further complicates the classification. Every coin has two sides. As evidenced in [10] exploring temporal information from motion is beneficial to dynamic scene classification. Nevertheless, it is nontrivial to obtain reliable motion clues due to complexity of motion patterns in scenes, e.g., the flicker motion lasts in very short time or motion of snowfall is too obscure to be seen. To recognize these motion patterns, previous methods often use consecutive frames or short video clips to capture temporal clues. These methods failed to fully capture motions in that only very small parts of the whole videos are used, losing long-term dynamic information.

In fact, the long-term description of the video can alleviate the incomplete information extraction problem, due to the holistic motion pattern in consideration. For example, the flicker motion in lightning scene is too swift such that most portions of the video clips or frames contain no lightning motion, where the extracted features could confuse the classification. But with long-term description, the flicker of lightning motion is included in the holistic feature than can enhance the classification process. Another example is the understanding of repetitive or periodic motion, e.g., rotating wheel of windmill and sea waves, where only incremental long-term observation can explicitly reveal its properties.

When it comes to the interpretation of long term properties, related methods have tried to employ long short-term memory (LSTM) [11]. The LSTM networks operate on frame-level

Manuscript received September 6, 2017; revised March 27, 2018; accepted March 31, 2018. Date of publication April 5, 2018; date of current version April 3, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1200100, in part by the National Science Fund for Distinguished Young Scholars under Grant 61425014, in part by the National Natural Science Foundation of China under Grant 61521091 and Grant 61773316, in part by the National Natural Science Foundation of China under Grant 61761130079, in part by the Key Research Program of Frontier Sciences, CAS under Grant QYZDY-SSW-JSC044, and in part by the National Key R&D Program of China under Grant 2017YFB1002202. This paper was recommended by Associate Editor A. K. Roy-Chowdhury. *(Corresponding author: Xianbin Cao.)*

Y. Huang, X. Cao, and X. Zhen are with the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China, and also with the Key Laboratory of Advanced Technology of Near Space Information System, Ministry of Industry and Information Technology of China, Beijing 100031, China (e-mail: xbcao@buaa.edu.cn).

Q. Wang is with the School of Computer Science and Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi’an 710072, China.

B. Zhang is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China.

X. Li is with the Xi’an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi’an 710119, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2823360

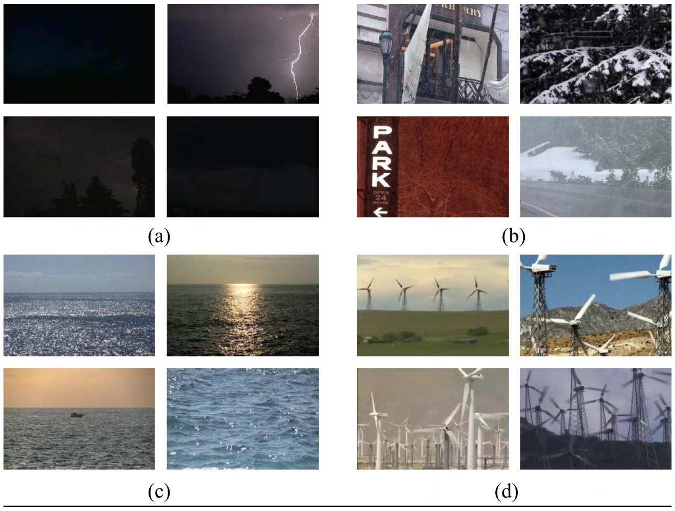


Fig. 1. Example shots for dynamic scenes. Large variation within the same category will make classification very difficult. And the existence of diverse motion patterns is another problem to be tackled.

convolutional neural network (CNN) activations and learn long term information over time. However, it usually needs a large amount of samples for training and the application to dynamic scene classification is rather restricted by inadequate samples of dynamic scene videos.

In this paper, we propose long-short term features (LSTF) to incorporate temporal dynamics and spatial information in dynamic scenes in a unified framework. The LSTF is established in two major stages. Firstly, we deploy variants of deep CNNs to encode spatial appearance and short-term dynamics into short-term deep features (STDF). In our setting, VGGnet is served as frame-level CNNs for describing spatial appearance. And C3D is used to process video clips, which retain both spatial and short-term temporal cues. In these CNNs, there are multiple layers extracting features from low level to high level, representing simple edges to complex objects. We use high-level features for constructing our STDF. Secondly, based on STDF, we compute long-term frequency features (LTFF) by the autoregressive moving average (ARMA) model, which effectively capture the long-term temporal dynamics. Since these high level features are related to objects in scenes, the LTFF describes the motion of these objects from global long-term range. In comparison to neural networks based LSTM, our LTFF is directly extracted without training on a large number of samples and is therefore more effective. The spatial and short-term dynamics are retained in STDF while long-term dynamics are fully captured in LTFF. By concatenating STDF and LTFF into a holistic feature vector, we achieves long-short term features (LSTF) for comprehensive and informative representation of dynamic scenes.

We summarize the contributions of this work in three major aspects as follows:

1) We introduce a new unified framework to jointly encoding spatial and temporal features, which establishes more informative and comprehensive representation of dynamic scenes for improved recognition performance.
2) We introduce the autoregressive moving average (ARMA) model to extract long-term temporal features, which is able to effectively capture motion patterns of dynamic scenes for more informative representation.
3) We successfully apply two variants of convolutional neural networks (CNNs) to extract both spatial and short-term temporal features of dynamic scenes, which have shown improved performance.

To evaluate the effectiveness of the LSTF for scene classification, we apply it to diverse dynamic scene recognition tasks on three datasets. YUPENN [7] and Maryland [10] are two natural dynamic scene datasets, while UCF-101 [12] is the action dataset, which can be regarded as a special type of dynamic scenes. Experimental results have shown that the proposed method achieves high performance on all three benchmark datasets and substantially exceeds most of state-of-the-art methods.

II. Related Work

In this section, we will review the most related work on dynamic scene classification in terms of representations based on hand-crafted features and deep-learned features.

A. Hand-Crafted Features

In the field of scene classification, there are plenty of hand-crafted features, which are intentionally designed without the learning process involved. These kind of features describe scenes from either spatial properties or temporal properties or both. The spatial properties can be obtained from orientation, color, frequency and etc.. For example, the Scale-Invariant Feature Transform (SIFT) descriptors proposed by Lowe [13] has been widely used for scene classification. The SIFT features make use of oriented gradient histograms to capture the spatial appearance cues. Similarly, the Histogram of Oriented Gradients (HOG) [14] also deploys the orientation information. Compared to SIFT, however, HOG lacks the rotation invariant property, which is more appropriate for human detection rather than scene classification. Olive and Torralba [15] took advantage of aggregated frequency information to construct GIST feature, which calculated holistic characteristics of scenes with five spatial properties (naturalness, openness, roughness, expansion, ruggedness).

In recent years, researchers have demonstrated that encoding temporal clues can largely enhance the performance of dynamic scene classification. As stated in [10], chaotic feature is proposed for dynamic scene classification, which uses the Lyapunov exponent to characterize the level of chaos in the scene and correlation properties to estimate the complexity of scenes. And thus the dynamic attributes of motion in scenes, such as the degree of busyness, the degree of flow granularity and the degree of regularity, are calculated. Except for this kind of chaotic temporal feature, other temporal features are mostly based on the optical flow method, which can capture the velocity information between subsequent frames further used for classification. For instance, Laptev [16] and Dalal *et al.* [17] use histogram of optical flow field to represent temporal cues. Likewise, Vasudevan *et al.* also [18] build temporal 5DMFV feature to describe characteristics of motion in scenes based on optical flow field. Nevertheless, these optical flow based methods fail to generate an effective representation in some special cases, where the complex motions often violate the illumination constancy assumption, e.g., the flicker motion.

As both spatial and temporal clues are proven to be efficient for dynamic scene classification, integrating them in the spatio-temporal features becomes prevalent. One possibility to construct spatio-temporal features is to transform current 2D feature into 3D feature. As in [19], 3D SIFT feature evolved from the widespread SIFT feature, is proposed to model 3D objects or video resources. Similarly, 3D HOG [20] shows improved results for action video classification. Moreover, spatio-temporal orientation energy (SOE) features is proposed in [7], by applying 3D Gaussian third-derivative filters in videos. The spatial-temporal Laplacian pyramid (STLP) [21] also show great effectiveness in extracting spatial and temporal features from video sequences for action recognition. In [22] and [23], the integration of SOE features and improved bag of feature model have achieved state-of-the-art results in dynamic scene classification field. Spatial-temporal oriented energies have also been successfully used for action recognition [24], showing impressive performance in multiple benchmark datasets. In addition, based on handcrafted features, supervised learning has recently been incorporated in spatial-temporal local descriptor learning for action recognition, which demonstrates improved performance [25], [26].

Although the aforementioned hand-crafted features have shown remarkable performance, the discriminative power is still limited when compared with deep-learned features [27], [28]. Nowadays, major attentions are paid on the implementation of deep-learned features, showing great effectiveness in handling large scale data in contrast to kernel methods [29], [30].

B. Deep-Learned Features

Deep-learned features are obtained through convolutional neural networks (ConvNets) with large amount of training data, which has demonstrated astounding results on large scale object recognition [31], [32]. However, since most ConvNets features are designed for object detection task, it can not be directly used for scene classification. To deal with this problem, Zhou *et al.* [33], [34] introduced a new scene-centric database called Places with over 7 million pictures of scenes, based on which the newly trained ConvNets features appear more suitable for scene classification task, such as VGGnet [35], Resnet [36] and etc. Gong *et al.* [37] improved the ConvNets by applying them within local multi-scale patches and further integrated the patch-based ConvNets with global ConvNets, which can capture both detailed information and holistic characteristics in scenes.

Differently from most ConvNets with a focus on static image scene classification, many researchers pay more attention to the video based scene classification. Gangopadhyay *et al.* [38] proposed a statistical aggregation solution based on convolutional neural networks for dynamic scene classification. The convolutional neural networks (CNN) use large datasets to acquire spatial information and the resulting CNN features are further analyzed by statistical methods in the temporal domain. Tran *et al.* [39] proposed C3D feature that transforms 2D ConvNets to 3D ConvNets, which exploits deep information in both spatial and temporal domain. And it has achieved good performance on various video analysis tasks, including dynamic scene classification. Unfortunately, the original C3D model was trained on sports video datasets and contains no prior information related to dynamic scenes. Due to the huge computational cost, C3D can only handle small video clips with few frames and discard the long term information in videos.

This paper is partly inspired by deep-learned feature presentation of scene classification. Compared with current ConvNets that only focus on the short term motion or spatial properties, the proposed method pays more attention to the long term motion information which combines long term information with short term deep information. In this way, these two complementary representation can make a better understanding toward dynamic scenes.

III. Long-Short Term Features

A. Overview

Due to distinct spatial and temporal characteristics in dynamic scenes, understanding scenes from multiple perspectives can be very beneficial. Moreover, both short and long term temporal information are essential for comprehensive representation of dynamic scenes and will be fully explored in our framework. As illustrated in Fig. 2, for the given video clips or frames we firstly employ CNNs to extract high-level features from frames and short video clips. We choose VGGnet [35] and C3D [39] to extract spatial and short-term temporal motions, which achieves short-term deep features (STDF). These STDF are then arranged in the temporal order and serve as inputs to extract long-term temporal dynamic features. We introduce the autoregressive moving average (ARMA) model to calculate the long-term frequency features (LTFF), which captures the long-term temporal motion patterns of dynamic scenes. Since both STDF and LTFF are essential and capture complementary scene properties, these two features are concatenated into a holistic feature vector called long short term features (LSTF) for final scene representation which is fed into linear support vector machines (SVM) for classification.

B. Short-Term Deep Features

The STDF mainly focuses on static spatial appearance and short-term motion properties. In the context of dynamic scenes, both handcrafted and learned features could be used. Among them, the deep-learned features are preferred because of their highly discriminative power compared to hand-crafted features [27]. We adopt two variants of CNN architectures, VGGnet [35] and C3D [39], to build up STDF. VGGnet mainly focuses on extracting spatial features from each frame of the video, while C3D can extract both spatial and short-term temporal features from short video clips.

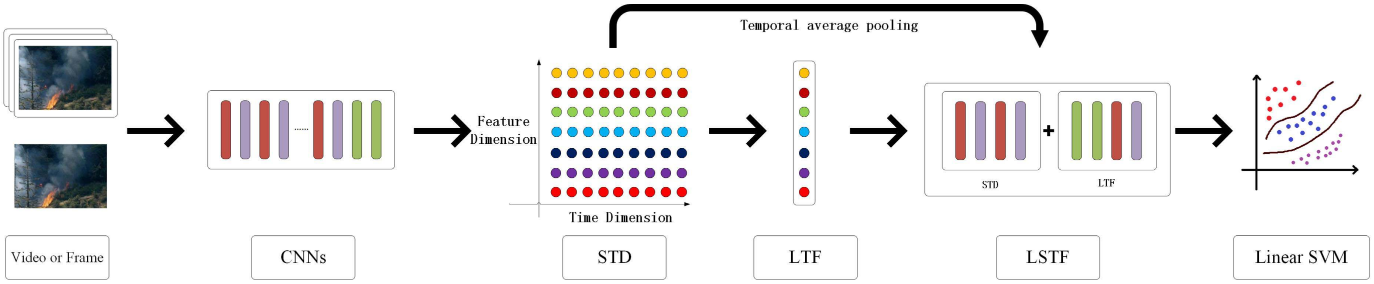


Fig. 2. Framework of the long-short term feature (LSTF). The LSTF reflects the diverse aspects of dynamic scenes, including long term frequency aspect as well as short term deep aspect. Short term deep feature is responsible for the static properties such as background and short term motion cues. While long term frequency feature is capable of representing long term properties, e.g., regularity or periodicity. Together they yield complementary representation for dynamic scene classification.

1) VGGnet Architecture: The VGG network takes frames of the size of 224×224 as inputs and passes through a stack of convolutional layers of size 3×3 with stride 1. Spatial pooling is conducted by 5 max-pooling layers with stride 2, followed by some of the convolutional layers. After a series of convolutional layers and max-pooling layers, there are three fully-connected layers, which output a 4096-d feature vector.

2) C3D Architecture: In contrast to 2D convolutional neural networks, C3D uses 3D convolutional layers and extracts both spatial and temporal information to generate spatio-temporal representation. The network architecture of C3D has 8 convolution layers, 5 pooling layers and 2 fully connected layers. Within each convolution layer, the 3D convolution filters are applied with the size of $3 \times 3 \times 3$ and stride $1 \times 1 \times 1$. All 3D pooling layers are $2 \times 2 \times 2$ with stride $1 \times 1 \times 1$, except $pool1$ with kernel size of $1 \times 1 \times 1$ to preserve the temporal information in the early phase. Due to the nature of three-dimensional convolution, C3D is well suited for extracting short-term dynamic of scenes. We have 4096-d output units for each fully connected layer as the feature vector.

3) Pre-Training and Feature Extraction: The VGGnet is pre-trained on the Places dataset [34], which contains over 10 million scene images, labeled with 476 scene semantic categories covering numerous types of environments encountered in the world. With the help of large scene dataset, these CNNs can well capture scene properties that are useful for scene classification. In VGGnet, the given video frames are processed by CNN networks and the outputs of $fc7$ layer in VGGnet extracted from each frames are averaged to form the representation. Different from these image based CNNs, C3D with 3D convolutional layers is trained on the sports-1M dataset, which has about 1 million sports videos downloaded from YouTube. In C3D, a given video is split into 16-frame long clips with a 15-frame overlap between two consecutive clips. Then, these video clips are passed into the C3D network to extract $fc6$ activation. As we have multiple $fc6$ activations obtained from each video clip, the average temporal pooling is finally used resulting in a 4096-d vector. In our method, the dense overlap setting is deployed for two reasons. Firstly, it can preserve more information within STDF. Secondly, since we use STDF as the inputs for constructing our LTFF, more samples are required to generate more effective representation.

C. Long-Term Frequency Features

Long-term motion patterns carry discriminative information to distinguish different scenes. Previous attempt tried long short-term memory (LSTM) [11] to explore long term temporal relationships in human action recognition. The LSTM network processes CNN activations as inputs and learns long term properties from these frame-level CNNs. However, since training such long term deep networks requires a very large of samples, its application in dynamic scene field is rather limited due to inadequate amount of dynamic scene videos. To overcome this problem, we introduce the autoregressive moving average (ARMA) model to extract long term frequency features (LTFF).

LTFF focuses on the interpretation of power spectrum in a long-term temporal range. In the context of dynamic scenes, the energy of power spectrum in the frequency domain is directly correlated with the statistical long term properties in the temporal domain, such as periodicity or regularity. As a result, the distribution of such power spectrum can be utilized for distinguishing diverse motion patterns. The power spectrum may be calculated directly from time series of raw pixels [40]. However, in the case of scene understanding, the raw pixels would not be suitable in that noises and camera movement can largely affect the construction of LTFF. To circumvent this problem, we work on CNN features instead of raw pixels. These CNNs extract features from the low level to high levels, representing simple edges to complex objects. The proposed LTFF features use high-level features from CNNs as inputs to measure how the motion patterns of these objects change over long periods of time. Compared with raw pixels, high-level features contain less noises and are more suitable for the construction of LTFF. For each dimension of high level features from CNNs, the distribution of power spectrum is calculated to discriminate diverse motion patterns in scenes.

Specifically, the short term deep features by C3D and VGGnet are chosen as the input to compute the LTFF. For each video clips or frames, we first obtain features from CNNs denoted as X with dimension of 4096 in VGGnet and C3D. We then align these features along temporal range as $X(t), t = 1, 2, 3 \dots T$, where T is the total number of video clips or video frames. In the end, these $X(t)$ samples can be taken as input data for the LTFF.

Traditionally, the power spectrum can be estimated via Fourier based methods, such as periodogram. However, those methods are highly sensitive to signal noises [40], which are consequently not suitable for dynamic scene data that contains much interference between background and target object. Moreover, due to the finite frames of video resources, the resolution of power spectrum estimated by Fourier based methods is very limited. To overcome those shortcomings, the latest time series models are used for spectral analysis [41], including autoregressive (AR), moving average (MA) and autoregressive moving average (ARMA) models. In what follows, we will firstly introduce the principles of time series models and then explain how to compute those models for power spectrum estimation.

1) Temporal Modeling: We model dynamic scenes by time series models. Given a stationary time series signal $\{X_t\}$, $t \in N$, it can be represented by a discrete-time autoregressive moving average ARMA(p, q) process, written as in Eq. (1) [41]:

$$x_t + a_1 x_{t-1} + \cdots + a_p x_{t-p} = \varepsilon_n + b_1 \varepsilon_{n-1} + \cdots + b_q \varepsilon_{n-q}, \tag{1}$$

where ε_t is a purely random white noise process of independent identically distributed stochastic variables with zero mean and variance σ_ε^2, $\{a_1, \cdots, a_p\}$, and $\{b_1, \cdots, b_q\}$ are the parameters in the ARMA model and p, q denote the order of the ARMA model.

This time series model in Eq. (1) is purely AR for $q = 0$ and purely MA for $p = 0$. It represents a parametric estimate of the power spectral density for the given time series data. The power spectral density $h(\omega)$ by the ARMA model is fully determined by the parameters in Eq. (1) and the variance σ_ε^2, which can be defined as

$$h(\omega) = \frac{\sigma_\varepsilon^2}{2\pi} \frac{\left|1 + \sum_{i=1}^{q} b_i e^{-j\omega i}\right|^2}{\left|1 + \sum_{i=1}^{p} a_i e^{-j\omega i}\right|^2}, \tag{2}$$

where ω denotes the frequency.

In modern time series models, it is essential to find the most suitable model with a proper order. For the same data source, the power spectrum estimated by AR, MA and ARMA with diverse order can be very different. In other words, if the model type and orders are selected properly, the time series models can provide the best solution [41]. Fortunately, Broersen [41] established an ARMASA model to deal with the model selection problem, which can automatically select the best model type and model order to estimate the power spectrum of the measured data. It calculates a number of candidate AR, MA and ARMA models and uses a statistical criteria to select the best fitting one.

2) Power Spectrum Estimation: The construction of appropriate time series models involves a number of interrelated problems, including the parameter estimation ($\{a_1, \cdots, a_p\}$ and $\{b_1, \cdots, b_q\}$), order selection (choosing the order p and q), and model identification (deciding among AR, MA and ARMA models).

In ARMASA, the problem mentioned above can be solved by following previous work [41]. The ARMASA algorithm consists of three main parts, i.e., AR model estimation, MA model estimation and ARMA model estimation. For the AR model, frequently-used estimators include the Burg method, Yule-Walker method and Forward-Backward Least Squares estimators. Since under the context of dynamic scenes, we only have limited data samples for estimation, the Burg estimator deserves a preference [40], [42]. And the order of the AR process can be measured by the Combined Information Criterion CIC(p) defined as:

$$\begin{aligned}\mathrm{CIC}(p) &= \log(\mathrm{RES}(p)) \\ &\quad + \max\left[\frac{1 + \frac{p}{N-p+1}}{1 - \frac{p}{N}} - 1, 3\sum_{i=1}^{p} \frac{1}{N - i + 1}\right],\end{aligned} \tag{3}$$

where RES(p) is the residual variance.

For the MA model, the Durbins method [43] is used for estimating MA parameters, which calculates a long AR model with the Burg method to approximate the MA process. The performance of the Durbins method can be improved by selecting a proper order [44]. The MA order q is selected with $GIC(q, 3)$, defined as:

$$\mathrm{GIC}(q, 3) = \log(\mathrm{RES}(q)) + \frac{3q}{N}. \tag{4}$$

The ARMA(p,q) model is estimated with the Durbins method [45], which calculates the parameters by minimizing

$$\begin{aligned}\sum_{n=\max(p,q)+1}^{\mathrm{N}} &\{x_n + a_1 x_{n-1} + \cdots + a_p x_{n-p} \\ &- \varepsilon_n - b_1 \varepsilon_{n-1} - \cdots - b_q \varepsilon_{n-q}\}^2.\end{aligned} \tag{5}$$

Finally, having the AR, MA and ARMA models, the prediction error of these three resulting models is estimated with the given data [46]. This step is to find the most suitable model for calculating the long term information. For MA and ARMA models, the prediction error is calculated by

$$\mathrm{PE}(m) = \{\mathrm{RES}(m)\} \frac{1 + \frac{m}{N}}{1 - \frac{m}{N}}, \tag{6}$$

where m is the number of estimated parameters in the model. For $AR(p)$ model, the prediction error is given by the expression [46]:

$$\mathrm{PE}(p) = \{\mathrm{RES}(p)\} \prod_{m=1}^{p} \frac{1 + \frac{1}{N+1-m}}{1 - \frac{1}{N+1-m}}. \tag{7}$$

The model type with the smallest estimate for the prediction error is selected. In this way, the most suitable model type with proper order can be determined for the given time series samples.

To sum up, the ARMASA can automatically select the best-fitting model (AR, MA or AMRA) as well as a proper order for estimating the parameters in Eq. (1), which can be further used to extract the power spectrum of time series as our LTFF in Eq. (2). The power spectrum in the frequency domain is directly correlated with the energy of periodic motions in temporal domain, which enables distinguishing distinctive motion patterns of scenes from different categories.

TABLE I
PERFORMANCE OF STATE OF THE ART METHODS ON THE MARYLAND DATASETS

	HOF [17]+GIST [15]	SOE [7]	SFA [8]	CSO[47]	BoSE [22]	CSR [9]	DPCF [23]	C3D [39]	**LSTF**
Avalanche	20 %	40 %	60 %	60 %	60 %	80 %	90 %	100%	100%
Boiling Water	50 %	50 %	70 %	80 %	70 %	100%	60 %	90 %	90 %
Chaotic Traffic	30 %	60 %	80 %	90 %	90 %	90 %	100%	90 %	90 %
Forest Fire	50 %	10 %	10 %	80 %	90 %	90 %	90 %	80 %	100%
Fountain	20 %	50 %	50 %	80 %	70 %	80 %	80 %	90 %	100%
Iceberg Collapse	20 %	40 %	60 %	60 %	60 %	90 %	80 %	80 %	100%
Landslide	20 %	20 %	60 %	30 %	60 %	80 %	80 %	80 %	80 %
Smooth Traffic	30 %	30 %	50 %	50 %	70 %	90 %	80 %	80 %	90 %
Tornado	40 %	70 %	70 %	80 %	90 %	90 %	80 %	80 %	90 %
Volcanic Eruption	20 %	10 %	80 %	70 %	80 %	100%	90 %	90 %	100%
Waterfall	20 %	60 %	50 %	50 %	100%	80 %	70 %	70 %	100%
Waves	80 %	50 %	60 %	80 %	90 %	90 %	100%	100%	100%
Whirlpool	30 %	70 %	80 %	70 %	80 %	60 %	80 %	90 %	90 %
Average	33 %	43 %	60 %	68 %	78 %	86 %	80 %	86 %	95 %

D. Long-Short Term Features

In order to achieve complementary and comprehensive feature representation of dynamic scenes, we integrate both STDF and LTFF into long-short term features (LSTF). Since we have employed two different CNN architectures, i.e., VGGnet and C3D, our LSTF can be computed based on VGGnet, C3D and their combination. In the end, the LSTF concatenates STDF and LTFF into one single holistic representation. Because the dimensionality of LTFF is very high, feature reduction is necessary before concatenation.

1) PCA and Normalization: In our method, STDF and LTFF are used to construct our LSTF. Compared with 4096d STDF, LTFF is of very high dimensionality and needs reduction. Specifically, for each dimensions in the features extracted from CNNs, the corresponding long term frequency features are constructed as a 16-d vector. As in C3D and VGG networks which have produced 4096-d feature vectors, the corresponding LTFF are $4096 \times 16 = 65536$-d vectors. It is computationally expensive to handle such high-dimensional features for classifiers, which would also lead to overfitting. Therefore, the principal component analysis (PCA) is deployed to reduce the dimensionality of LTFF. After PCA, the dimension of LTFF can be approximately reduced to 100, which can further improve the computational efficiency.

Since we use multiple views to describe scene properties and construct our LSTF with STDF and LTFF, normalization is required to avoid bias for proper concatenation of these features. Specifically, the min-max normalization is introduced in Eq. (8)

$$\bar{x} = \frac{2(x - x_{\min})}{x_{\max} - x_{\min}} - 1, \tag{8}$$

where x is the original input feature and $\bar{x}$ is the normalized output feature.

IV. EXPERIMENTS AND RESULTS

We conduct extensive experiments on three publicly available datasets, i.e., YUPENN [7], Maryland and UCF101 [12], in which actions in UCF-101 are treated as a special type of dynamic scenes. Our method consistently achieves high performance on all three datasets and largely surpasses most of state-of-the-art methods. We have also provided comprehensive experimental analysis to show the effectiveness of each component in our method.

A. Implementation Details

For each video sample, VGGnet takes individual video frames with a stride of 1 frame as the input and generates spatial features, which are then averaged and normalized; while C3D takes 16-frames video clips with overlap 15 as the input and generates spatio-temporal features, which are also averaged and normalized; the outputs from VGGnet and C3D build STDF. The LTFF are computed based on these STDF. Since both STDF and LTFF are essential for classification, these two complementary features are normalized by minmax normalization and concatenated into the long-short term features (LSTF). For a fair comparison, we use the linear SVM classifier, where parameters of SVM are empirically set to $C = 0.01, e = 0.001$ by following previous work [39]. To keep the consistency with common protocols [7], [8], [18], we use the leave-one-out cross-validation (LOOCV) rule for dynamic scene classification and 3-fold cross validation for UCF-101.

B. Results on Maryland

1) Dataset: The Maryland dataset contains 13 dynamic scene categories, with 10 samples in each category. The video samples are collected from Internet sharing sites, e.g., YouTube. This dataset is very challenging that the videos have large variation in illumination, image scale, view point and video length. Also, the camera motions as well as scene cuts are confounded with object motions, which further increase the complexity of dynamic scenes. The Maryland dataset is very small with only 130 available video samples. Thus, we use leave-one-out strategy for our experiments. We conduct the experiments 10 times and each time 9 video samples from each category are used for training and 1 video sample for testing. The average accuracy rates are calculated from these 10 times experiments.

2) Performance: In Table I, we compare our LSTF with state-of-the-art methods on the Maryland dataset. The top-performing algorithm is our newly proposed LSTF which

TABLE II
PERFORMANCE OF STATE OF THE ART METHODS ON THE YUPENN DATASET

	HOF [17] + GIST [15]	SOE [7]	SFA [8]	CSO [47]	BoSE [22]	CSR [9]	DPCF [23]	C3D [39]	**LSTF**
Beach	87 %	93 %	93 %	100%	100%	100%	100%	97 %	97 %
Elevator	87 %	100%	97 %	100%	97 %	100%	100%	100%	100%
Fire	63 %	67 %	70 %	83 %	93 %	93 %	97 %	100%	97%
Fountain	43 %	43 %	57 %	47 %	87 %	97 %	93 %	83 %	90 %
Highway	47 %	70 %	93 %	73 %	100%	100%	100%	97 %	100%
Lightning	63 %	77 %	87 %	93 %	97 %	90 %	100%	93 %	90 %
Ocean	97 %	100%	100%	90 %	100%	97 %	100%	100%	100%
Railway	83 %	80 %	93 %	93 %	100%	100%	100%	97 %	100%
River	77 %	93 %	87 %	97 %	97 %	100%	100%	100%	100%
Sky	87 %	83 %	93 %	100%	97 %	90 %	100%	97 %	97%
Snowing	47 %	87 %	70 %	57 %	97 %	87 %	97 %	93 %	100%
Street	77 %	90 %	97 %	97 %	100%	100%	100%	100%	100%
Waterfall	47 %	63 %	73 %	77 %	83 %	67 %	97 %	97 %	97 %
Windmill	53 %	83 %	87 %	93 %	100%	97 %	100%	100%	100%
Average	68 %	81 %	85 %	86 %	96 %	94 %	99 %	97 %	98 %

achieves 95% accuracy rates on average and outperforms the state-of-the-art methods by a large margin, e.g., 9% in DPCF [23] and 15% in C3D [39]. On a single class level, it is particularly interesting to note that most previous methods exhibit weaker performance on scenes containing different patterns of water, e.g. boiling water with DPCF, waterfall with C3D and whirlpool with CSR. Distinguishing diversified water motions tends to be the most difficult for previous algorithms, suggesting that capturing differences between water patterns based on their dynamics remains unsettled. Compared with previous methods, our newly proposed LSTF present outstanding performance on these scenes, with 90% in boiling water, 100% in waterfall and 90% in whirlpool. The improvement is largely benefit from the introduce of long-short term motion properties that emphasize the differences of dynamics in both long and short term scale, indicating the effectiveness of long-short information for dynamic scene classification.

C. Results on YUPENN

1) Dataset: The YUPENN dataset contains 14 different categories of videos related to dynamic scenes, with 30 samples in each of these categories. These video samples are captured with fixed camera and last about 5 seconds having 150 frames. Because our dynamic scene dataset is rather small, in the following experiment the leave-one-out strategy is adopted to generate more precise and convincing results. To be more specific, we conduct 30 times experiments, each of which uses 29 samples in each category for training the SVM classifier and the remaining 1 sample for testing. The average accuracy rates are calculated from these 30 times experiments.

2) Performance: Since videos in the YUPENN dataset are captured with fixed cameras, the task is relatively easy in comparison to the Maryland dataset. As can be seen in Table II, the proposed LSTF model achieves state-of-the-art performance on the YUPENN dataset, outperforming previous hand-crafted features and other CNN based methods. Compared with hand-crafted features such as SOE and SFA, the improvements is over 8%. While for CNN based methods such as VGGnet and C3D, our LSTF can still improve the performance. The performance gain mainly stems from the consideration of long term properties and the integration of diverse CNN architectures with complementary representation. Outstanding performances (over 95%) are acquired by most algorithms, indicating that performance is saturated on this dataset.

TABLE III
PERFORMANCE OF STATE OF THE ART METHODS ON THE UCF-101 DATASET

Method	**Accuracy**
C3D [39]	82.3%
VGG [35]	79.9%
VGG+C3D	82.9%
Spatiotemporal ConvNet [48]	65.4%
LRCN [49]	82.9%
LSTM composite model [12]	84.3%
Two-Stream ConvNet [50]	88.0%
LSTF	85.2%

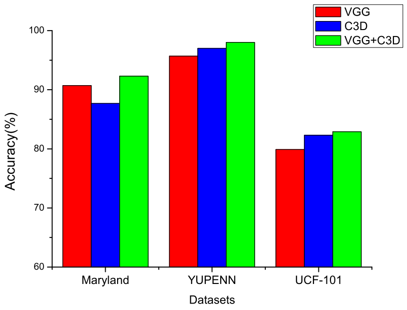


Fig. 3. The performance comparison of VGG, C3D and their combinations on three datasets.

D. Results on UCF-101

1) Dataset: The UCF-101 [12] contains 13320 videos with 101 action categories covering a broad set of dynamic scenes. We follow the 3-fold evaluation protocol conducting 3 times experiments, each of which uses 2/3 samples for training and 1/3 samples for testing. The average accuracy rates are calculated based on these three times experiments. Note that we treat actions in this dataset as a special type of scenes, we further apply our method on UCF-101 dataset and test how our LSTF model perform in this dataset.

2) Performance: We compare our LSTF with state-of-the-art methods on the UCF-101 dataset. The results are reported in Table III. The performance of our LSTF is better to baseline

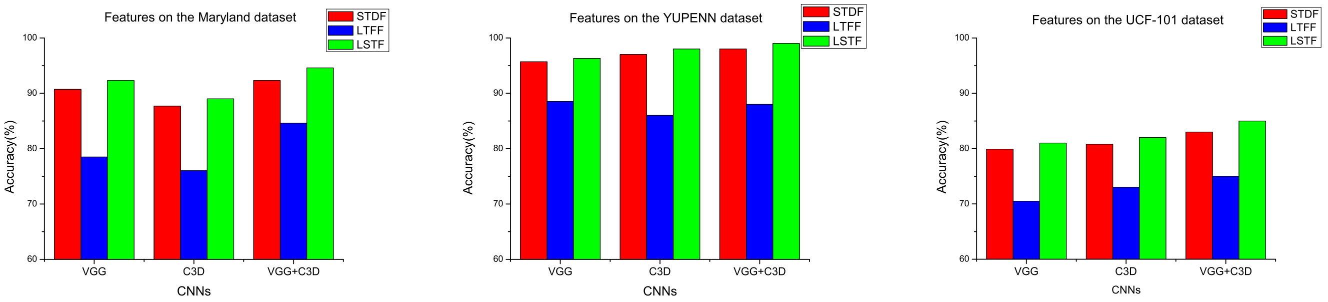


Fig. 4. Performance comparison of STDF, LTFF and LSTF, on Maryland, YUPENN and UCF 101 datasets, respectively.

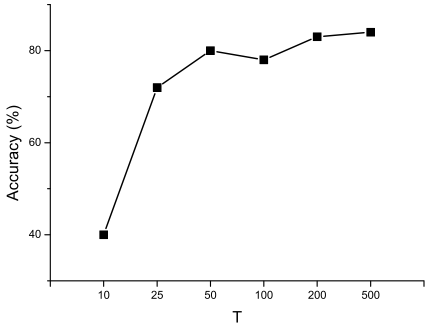


Fig. 5. Long term effect (LTFF) on the Maryland dataset.

methods including VGGnet and C3D. The results show that the integration of two CNN architectures can improve 1% performance. And with the introduce of long-term clues, our newly proposed LSTF further increase the accuracy rates by nearly 2%. Consistent with previous two datasets, the results demonstrate the effectiveness of our long-short term features. Because in our method we only use intensity information for representation, the performance is still lower than methods that use optical flow, e.g. Two-Stream ConvNet. To summarize, the performance for action recognition shows the great generality of the proposed LSTF for video representation.

E. Ablation Studies

We conduct three sets of experiments to separately evaluate the contribution of each components in our model to the overall performance. In our model, we have three main components, namely, STDF, LTFF and LSTF. For STDF, we evaluate its performance with different CNN architectures and their combinations. For LTFF, we test the impact of the temporal scale T in LTFF on the overall performance and demonstrate that long-term information is beneficial for dynamic scene classification. For LSTF, we evaluate the contributions of STDF, LTFF and LSTF to the performance, which shows that all of them are indispensable in our model.

1) CNN Architectures in STDF: Fig. 3 shows the results obtained using VGGnet, C3D and their combination on the Maryland, YUPENN and UCF-101 datasets. It is interesting to find that VGGnet with only spatial scene properties can achieve better performance than C3D on the Maryland dataset. This could be due to that VGGnet is trained on large scene dataset and can well capture scene properties, while C3D is trained on the sports-1M dataset which is good at extracting motion information rather than scene properties. Note that both scene properties and motion clues are indispensable. We concatenate features from VGGnet and C3D to better describe dynamic scenes. The combination of features from VGG and C3D achieves 5% improvement on the Maryland dataset.

On the YUPENN dataset, VGGnet, C3D and their combination all delivers high performance with the recognition rates over 95%. The results have shown that different CNN architectures, i.e., VGG and C3D, provide complementary information for comprehensive representation of scenes, which improve the overall performance.

Similarly, with regard to the action recognition task, we also use multiple CNN architectures for evaluation. The VGGnet is pre-trained on the Imagenet dataset [31], which uses single frame of videos for representation. C3D is pre-trained on the sports-1M dataset, using 16-frame video clips for representation. As shown in Fig. 3, C3D performs better than VGGnet. This is because VGGnet is trained on the static image dataset without any motion clues. While C3D is trained on large quantities of sports videos and is very good at extracting human action information. Furthermore, the integration of different CNNs extracts complementary properties and the experiment demonstrates that integrated CNNs can greatly improve the performance.

2) Evaluation of Components in LSTF: We show the performance of STDF, LTFF and LSTF in Fig. 4, which aims to demonstrate the significance of long term properties and the complementary effect of these features. In the left column of Fig. 4, we evaluate each components on the Maryland dataset. Since our STDF can be multiple choices such as VGGnet and C3D, these individual STDF as well as the corresponding LTFF and LSTF are compared in the experiment. We can observe from Fig. 4 that by integrating STDF with LTFF in the LSTF, the performance can be enhanced for both VGGnet and C3D architectures. This indicates that long term properties are essential for the understanding of dynamic scenes. Therefore, the combination of STDF and LTFF used in the algorithm generates a complementary and informative representations of dynamic scenes. And with the help of integrated CNN architectures, the performance can be further improved.

In the middle column of Fig. 4, we conduct experiment on YUPENN dataset. Consistent with Maryland dataset, by integrating STDF with LTFF in the LSTF, the performance can be enhanced under different CNN architectures on YUPENN dataset. Considering the fact that these CNNs have achieved an accuracy of over 95%, the performance is saturated on this dataset, as described in [23] and the performance gain is not as obvious as the Maryland dataset.

The performance comparison of STDF, LTFF and LSTF on the UCF-101 dataset is shown in the rightmost column of Fig. 4. Similarly, from the results, we find that by integrating STDF with LTFF in the LSTF, the performance can be enhanced under various CNN architectures, showing the great effectiveness of the LSTF for scene representation.

3) Temporal Scales in LTFF: The temporal scales in LTFF also show varied effects on the performance, which have also been exploited in our experiments. The temporal range T controls how many frames we utilize to obtain long term temporal features. We evaluate the effect of the temporal range on the Maryland dataset. In this dataset, the video samples have large variation in length, ranging from 40 frames to over 2000 frames, and thus we test the performance with $T = 25, 50, 100, 200$ and all frames. The classification performance under various temporal ranges of T is shown in Fig. 5. The results indicate that a larger temporal range leads to higher performance, and this would be due to that long term observation obtains more information such as regularity or periodicity. To summarize, the optimal temporal range is reached when using all video frames. We therefore use all frames of the video as input in our experiments.

V. Conclusion

In this paper, we have presented a new unified framework to build long-short term feature (LSTF) presentation of dynamic scenes. Our LSTF jointly encodes spatial and temporal information of dynamic scenes in one single framework. Two variants of deep convolutional neural networks (CNN) have been adopted to extract spatial and short-term temporal features, which establishes short-term deep features (STDF). Based on STDF, autoregressive moving average (ARMA) model is introduced into dynamic scenes to extract long-term frequency features (LTFF). By combining STDF and LTFF, we achieve long-short term features (LSTF) for comprehensive and informative representation of dynamic scenes. We have evaluated the great effectiveness of LSTF for dynamic scene classification by extensive experiments on three challenging datasets, i.e., YUPENN, Maryland and UCF101. Results have shown that LTSF consistently achieves high performance which is competitive to state-of-the-art methods.

References

[1] T. Huynh-The, O. Banos, S. Lee, B. H. Kang, E.-S. Kim, and T. Le-Tien, "NIC: A robust background extraction algorithm for foreground detection in dynamic scenes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp. 1478–1490, Jul. 2017.

[2] J. Shao, C. C. Loy, K. Kang, and X. Wang, "Crowded scene understanding by deeply learned volumetric slices," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 613–623, Mar. 2017.

[3] 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.

[4] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3241–3253, Aug. 2014.

[5] S. Guo, W. Huang, L. Wang, and Y. Qiao, "Locally supervised deep hybrid model for scene recognition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 808–820, Feb. 2017.

[6] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao, "Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2055–2068, Apr. 2017.

[7] K. G. Derpanis, M. Lecce, K. Daniilidis, and R. P. Wildes, "Dynamic scene understanding: The role of orientation features in space and time in scene classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1306–1313.

[8] C. Thériault, N. Thome, and M. Cord, "Dynamic scene classification: Learning motion descriptors with slow features analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2603–2610.

[9] L. Du and H. Ling, "Dynamic scene classification using redundant spatial scenelets," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2156–2165, Sep. 2016.

[10] N. Shroff, P. Turaga, and R. Chellappa, "Moving vistas: Exploiting motion for describing scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1911–1918.

[11] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4694–4702.

[12] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," CRCV-TR-12-01, Nov. 2012.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[15] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[16] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[17] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. ECCV*, Graz, Austria, 2006, pp. 428–441.

[18] A. B. Vasudevan, S. Muralidharan, S. P. Chintapalli, and S. Raman, "Dynamic scene classification using spatial and temporal cues," in *Proc. IEEE Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 803–810.

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

[20] C. Hua, Y. Makihara, and Y. Yagi, "Pedestrian detection by using a spatio-temporal histogram of oriented gradients," *IEICE Trans. Inf. Syst.*, vol. E96.D, no. 6, pp. 1376–1386, 2013.

[21] 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, Jun. 2014.

[22] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Bags of spacetime energies for dynamic scene recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2681–2688.

[23] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Dynamic scene recognition with complementary spatiotemporal features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 12, pp. 2389–2401, Dec. 2016.

[24] X. Zhen, L. Shao, and X. Li, "Action recognition by spatio-temporal oriented energies," *Inf. Sci.*, vol. 281, pp. 295–309, Oct. 2014.

[25] M. Yu, L. Shao, X. Zhen, and X. He, "Local feature discriminant projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1908–1914, Sep. 2016.

[26] X. Zhen, F. Zheng, L. Shao, X. Cao, and D. Xu, "Supervised local descriptor learning for human action recognition," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2056–2065, Sep. 2017.

[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] B. Zhang *et al.*, "Latent constrained correlation filter," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1038–1048, Mar. 2018.

[29] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 221–228.

[30] X. Zhen, M. Yu, X. He, and S. Li, "Multi-target regression via robust low-rank learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 497–504, Feb. 2018.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25. 2012, pp. 1097–1105.

[33] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, vol. 27. 2014, pp. 487–495.

[34] B. Zhou *et al.*, “Places: A 10 million image database for scene recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Comput. Sci.*, 2014.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[37] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in *Computer Vision* (Lecture Notes in Computer Science), vol. 8695. 2014, pp. 392–407.

[38] A. Gangopadhyay, S. M. Tripathi, I. Jindal, and S. Raman, “SA-CNN: Dynamic scene classification using convolutional neural networks,” in *Proc. Comput. Vis. Pattern Recognit.*, 2015.

[39] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.

[40] P. M. T. Broersen, “ARMAsel for detection and correction of outliers in univariate stochastic data,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 3, pp. 446–453, Mar. 2008.

[41] P. M. T. Broersen, *Automatic Autocorrelation and Spectral Analysis*. Berlin, Germany: Springer, 2006.

[42] P. M. T. Broersen, “Finite sample criteria for autoregressive order selection,” *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3550–3558, Dec. 2000.

[43] J. Durbin, “Efficient estimation of parameters in moving-average models,” *Biometrika*, vol. 46, nos. 3–4, pp. 306–316, 1959.

[44] P. M. T. Broersen, “Facts and fiction in spectral analysis,” *IEEE Trans. Instrum. Meas.*, vol. 49, no. 4, pp. 766–772, Aug. 2000.

[45] J. Durbin, “The fitting of time-series models,” *Revue l’Institut Int. Statistique*, vol. 28, no. 3, pp. 233–244, 1960.

[46] R. Shanmugam, “Introduction to time series and forecasting,” *Technometrics*, vol. 39, no. 4, p. 426, 2016.

[47] C. Feichtenhofer, A. Pinz, and R. Wildes, “Spacetime forests with complementary features for dynamic scene recognition,” in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 56.1–56.11.

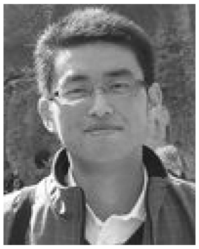
[48] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[49] J. Donahue *et al.*, “Long-term recurrent convolutional networks for visual recognition and description,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.

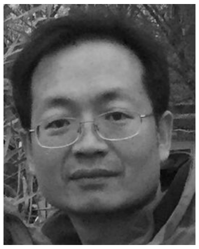
[50] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, vol. 27. 2014, pp. 568–576.

Yuanjun Huang received the B.S. degree in electronics and information engineering from Beihang University, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree with the National Key Laboratory of CNS/ATM, School of Electronics and Information Engineering. His research interests include computer vision, applied machine learning, and dynamic scene classification.

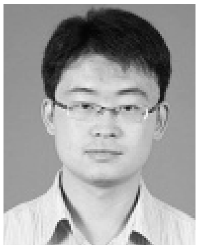
Xianbin Cao (M’08–SM’10) received the Ph.D. degree in information science from University of Science and Technology of China, Beijing, China, in 1996. He is currently the Dean and a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing. His current research interests include intelligent transportation systems, airspace transportation management, and intelligent computation.



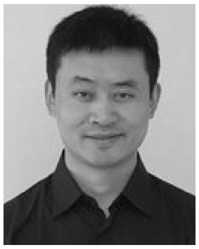
Qi Wang (M’15–SM’15) received the B.E. degree in automation and Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi’an, China. His research interests include computer vision and pattern recognition.



Baochang Zhang received the B.S., M.S., and Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1999, 2001, and 2006, respectively, all in computer science. From 2006 to 2008, he was a Research Fellow with The Chinese University of Hong Kong, Hong Kong, and Griffith University, Brisbane, QLD, Australia. He is currently an Associate Professor with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His current research interests include pattern recognition, machine learning, face recognition, and wavelets.



Xiantong Zhen received the B.S. and M.E. degrees from Lanzhou University, Lanzhou, China, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K., in 2013. He was a Post-Doctoral Fellow with University of Western Ontario, London, ON, Canada, and The University of Texas at Arlington, TX, USA, from 2013 to 2017. He is currently an Associate Professor with Beihang University, Beijing, China. His research interests include machine learning, computer vision, and medical image analysis.



Xuelong Li (M’02–SM’07–F’12) is a Full Professor with the Xi’an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi’an, Shaanxi, China, and with the University of Chinese Academy of Sciences, Beijing, China.