Attentive Temporal Pyramid Network for Dynamic Scene Classification

Yuanjun Huang,[1,2,3,4] **Xianbin Cao,**[1,3,4] **Xiantong Zhen,**[1,3,4] **Jungong Han**[2]
[1]School of Electronics and Information Engineering, Beihang University, Beijing, 100191, China
[2]Lancaster University, Lancaster, LA1 4YW, UK
[3]Key Laboratory of Advanced technology of Near Space Information System (Beihang University), Ministry of Industry and Information Technology of China
[4]Beijing Advanced Innovation Center for Big Data-Based Precision Medicine

Abstract

Dynamic scene classification is an important yet challenging problem especially with the presence of defected or irrelevant frames due to unconstrained imaging conditions such as illumination, camera motion and irrelevant background. In this paper, we propose the attentive temporal pyramid network (ATP-Net) to establish effective representations of dynamic scenes by extracting and aggregating the most informative and discriminative features. The proposed ATP-Net detects informative features of frames that contain the most relevant information to scenes by a temporal pyramid structure with the incorporated attention mechanism. These frame features are effectively fused by a newly designed kernel aggregation layer based on kernel approximation into a discriminative holistic representations of dynamic scenes. The proposed ATP-Net leverages the strength of attention mechanism to select the most relevant frame features and the ability of kernels to achieve optimal feature fusion for discriminative representations of dynamic scenes. Extensive experiments and comparisons are conducted on three benchmark datasets and the results show our superiority over the state-of-the-art methods on all these three benchmark datasets.

Introduction

Dynamic scene classification has been extensively studied in the computer vision community in recent years (Feichtenhofer, Pinz, and Wildes 2016) (Huang et al. 2018)(Vasudevan et al. 2013), owing to its wide applications in video searching, autonomous driving (Yan et al. 2018b) and surveillance (Yan et al. 2018a). However, dynamic scene classification is still an unsolved problem, due to unconstrained imaging conditions such as poor illumination, abrupt camera motion, unfavorable viewpoints and irrelevant background.

In comparison to static scenes, recognition of dynamic scenes in videos captured by moving cameras is more challenging attributable to the introduced much more blurred and irrelevant video frames. In these videos, most video frames only contain background instead of meaningful label-related contents that are distractions for classification. Thus, exploiting what truly matters and selecting significant contents from video frames is the key component for dynamic scene classification.

Recently, scene classification methods mostly adopt convolutional neural networks (CNNs) (Zhou et al. 2016)(Huang et al. 2017b) that have proved to be very powerful of learning strong representations in wide range of visual tasks. CNNs such as VGGnet (Simonyan and Zisserman 2014b), Resnet (He et al. 2016), Densenet (Huang et al. 2017a) etc., are regarded as the feature extractors for obtaining local representations from different regions of images. However, these CNNs are designed to work with still image only and it is non-trivial, if not impossible, to extend these models to videos containing complex temporal clues. A straightforward way to deal with this problem is to use the average pooling strategy which simply takes the average of features along temporal frames. However, average pooling inevitably discards the temporal sequential information and introduces negative effects of irrelevant frames on the overall representations of scenes.

To achieve more accurate recognition of dynamic scene, it would be beneficial to fully explore the temporal dynamic information. There are basically two pathways to model temporal clues within CNNs. One way is to explicitly model the video as an ordered sequence of frames based on long short-term memory (LSTM) (Donahue et al. 2015) or gated recurrent unit (GRU) (Chung et al. 2014). These models usually adopt memory cells to store, modify and access internal state so as to discover the long-range sequential information. Alternatively, another way of capturing the temporal information in CNNs resorts to the two-stream architecture (Simonyan and Zisserman 2014a) which uses both RGB and dense optical flows as the inputs for CNNs. By incorporating these two sources of information, the model encodes both spatial and temporal clues in the two-stream network. Despite the success of these methods, the computational cost tends to be high, and in addition, indiscriminately using entire video frames for modeling will introduce negative effects of irrelevant and noisy frames, thereby compromising the classification performance.

Meanwhile, attention mechanism has recently become increasingly popular (Ba, Mnih, and Kavukcuoglu 2014)(Mnih et al. 2014)(Vaswani et al. 2017)(Wang et al. 2017). Typically, attention modules are injected into the existing CNN architectures and guide the network to focus more on the regions of interest. By training the attention network in an end-to-end manner, it can generate attention-

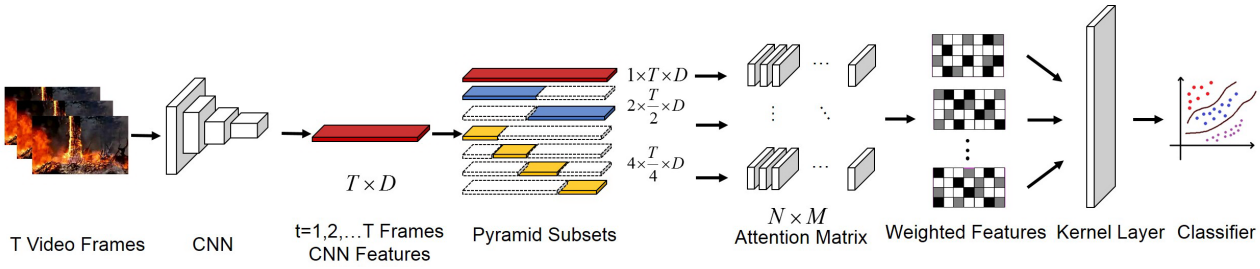

Figure 1: Framework of the ATP-Net. For a given video sample containing T frames, we extract their CNN features with $t = 1, 2, ..., T$. Then we use pyramid structure to divide these CNN features into n pyramid CNN subsets where $n = 1, 2, 4$. The total number of pyramid CNN subsets are $N = 1 + 2 + 4 = 7$. Next, these pyramid CNN subsets are fed into an attention matrix of $N \times M$ in order to obtain the weights of every frame of CNN features, where each element of attention matrix represents an attention module that is used for calculating frame weights. In the attention matrix, we have $N \times M$ attention modules in total in which N represents the pyramid CNN subsets and M is the number of attention modules that we used for calculating weights of each pyramid CNN subsets. The pyramid CNN subsets and attention matrix enable our model to calculate weights from different temporal scales and reflect diversified aspects of video. The output of attention matrix is weighted features which are then aggregated in kernel aggregation layer. In the final stage, the output features are classified by softmax function.

aware features for better representations. Inspired by the success of spatial attention modules, some researchers extend this idea to acquire temporal attentions. Neural aggregation network (NAN) (Yang et al. 2017) is proposed specifically for video face recognition, which incorporates attention aggregation module in CNNs and adaptively aggregates image-level features into video-level features. Since face images in these videos do not always have pleasant viewpoint and clear facial details, the NAN filters blurred and irrelevant face images out while focusing on essential face images that supports recognition. While comparing with video-based face recognition, dynamic scene classification is more complicated due to the great variability of contents both temporally and spatially. Therefore, it is highly desired to extract the most informative and discriminative frames from videos of dynamic scenes for compact representations.

To this end, we propose the attentive temporal pyramid network (ATP-Net) for dynamic scene classification. Since video sequences usually contain hundreds of frames, finding key features within these frames is complicated. Thus, we stack multiple independent attention modules at temporal pyramid scales. The idea behind temporal pyramid structure is simple so that binary searching key features from hundreds of frames will rapidly and accurately locate the key frames from both long-term and short-term scales. As the temporal scale being extended, multiple pyramid attention features are extracted from videos. In order to better organize these features, we design the kernel aggregation layer to achieve holistic representations. The kernel aggregation layer fuses multiple features based on kernel approximation such that the non-linear power of features can be significantly enhanced. The kernel aggregation layer and independent attention modules ensure the learning weight of these multiple attentions to keep diversified and fully describe different aspects of videos. The newly designed kernel aggregation layer enjoys the benefits of both kernels and neural networks, and is adaptively learned from the training data. Extensive experiments and comparisons are conducted on three benchmark datasets and the results show the superiority of the proposed ATP-Net over the state-of-the-art methods on all these three benchmark datasets.

In summary, the major contributions of this work lie in the following three aspects:

–We propose the attentive temporal pyramid network (ATP-Net) for dynamic scene classification. The ATP-Net is able to filter out defected and irrelevant frames while focusing on essential contents detected from long-term to short-term temporal scales.

–We develop a kernel aggregation layer derived from kernel approximation to effectively aggregate multiple features from the temporal pyramid, achieving discriminative and compact feature representations.

– We evaluate the ATP-Net on three diverse benchmark datasets, which achieves new state-of-the-art performance on the benchmarks, substantially outperforming previous methods.

The rest paper is organized as follows. In methodology, the proposed ATP-Net is presented in details with theoretical analysis. The experimental section shows the results and comparisons with the state-of-the-arts baselines on three benchmark datasets. Finally, we draw some conclusions.

Methodology

Network overview

The overall structure of the proposed ATP-Net is shown in Figure.1. It can be decomposed into three parts: the first part is the CNN architecture to extract spatial local features from video frames. For a given video sample of T frames, we will obtain $T \times D$ CNN features from the activation of the last feature layer in CNN architectures, where D is the dimension of CNN features. In the second part, these $T \times D$ CNN features are divided into n pyramid CNN subsets and fed into the attention matrix to calculate the weights of frames.

The pyramid CNN subsets contain $N = 7$ CNN subsets of size $1 \times T \times D$, $2 \times \frac{2}{T} \times D$ and $4 \times \frac{4}{T} \times D$. In the attention matrix, we have $N \times M$ attention modules in total in which N represents the pyramid CNN subsets and M is the number of attention modules that we used for calculating weights of each pyramid CNN subsets. The pyramid CNN subsets and attention matrix enable our model to rapidly locate where the key frames are and reflect diversified aspects of videos. In the final part, the kernel aggregation layer is employed to fuse features from the temporal pyramid. In the next following sections, we will detail each component of our model and illustrate how these elements work in our model.

CNNs for local features

For dynamic scene classification, it is very common to obtain spatial local features with various CNNs available. Since most CNN models are already pretrained on large dataset, e.g., Places2, imagenet and proved to be very powerful, we can take advantage of these pre-trained models and use them directly for feature extraction. To be specific, we use Resnet-50 for spatial feature extraction networks.

The spatial features are defined as a set of ordered frame-level features extracted from each frame of a video. We use X of size $T \times D$ to represent spatial features extracted from T frames of a given video sample with dimension of D. The frame-level CNN features x_t in X can be defined as :

$$X = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T, \tag{1}$$

where T is the total frames of the video data and D is the dimension of CNN features. And the average pooling method can be denoted as:

$$\bar{X} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t. \tag{2}$$

Attentive Temporal Pyramid Network

In our videos captured by moving cameras, lots of frames are blurred with changing viewpoints all the time which is not favorable for classification. Moreover, some videos are even untrimmed with only a small portion of video clips that are related to the scene class. Thus, exploiting what truly matters in videos is very important in our task. To deal with the aforementioned problem, we propose the attentive temporal pyramid network (ATP-Net) which can be divided into two main modules: the single attention module and the temporal pyramid attention matrix.

Attention module The attention module aims to filter defected or irrelevant frames out and only focus on the essential contents. The network structure of attention module is depicted in Figure.2. We use 2 batch normalization layers, 2 Relu activations, 1 attention layer and 1 dropout layer to construct the attention module network. The input of attention module is added on the attention layer, which is commonly used as a residual component in Resnet.

Having obtained frame-level features $X = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T$ for a given video sample of T frames, the next step is to find the most proper weights a_t that linearly aggregate these

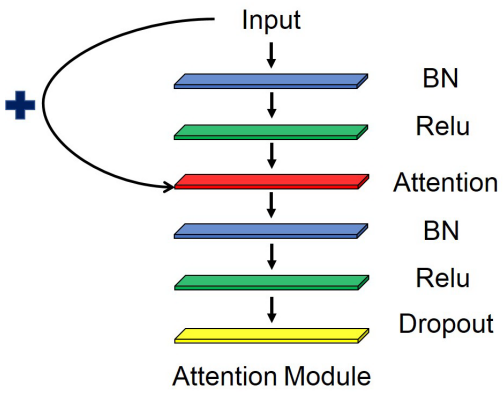

Figure 2: The network structure of attention module. 2 batch normalization layers, 2 Relu activations, 1 attention layer and 1 dropout layer are used in a single attention module.

frame-level features and yield best performance in the classification stage:

$$f = \sum_{t=1}^{T} a_t \mathbf{x}_t, \tag{3}$$

where f is the weighted features. If we define $a_t = \frac{1}{T}$, then it will degrade to averaging pooling method. As in the attention module, we try to adaptively learn a_t in a neural network. The attention module consists of a fully-connected layer to learn the weights of frame-level features that can be defined as:

$$\mathbf{a} = \text{softmax}(\mathbf{w}\mathbf{x}^T + b), \tag{4}$$

where w and b are parameters of dimensionality T and D, respectively.

Inspired by the success of Resnet, we add a residual component in Eq. 3, and it will be changed into:

$$f' = \sum_{t=1}^{T} (1 + a_t)\mathbf{x}_t. \tag{5}$$

The shortcut in Eq. (5) will accelerate the fluent of gradient and achieve better performance.

Temporal pyramid attention matrix The original attention module is very simple and not enough to deal with our problem. Since we don't know where the exact locations of label related frames and blurred as well as unfavorable viewpoints frames are, efficiently filtering out noisy frames while keeping essential features is very difficult for a simple single attention module. Therefore, to be able to rapidly locate the key frames and accurately generate effective features, we propose a temporal pyramid attention matrix. Normally, previous attention methods tend to find the optimal solution from entire input data, which is very slow and inaccurate. While in our model, we aim to use pyramid structure with attention matrix to exploit the key frames from divided input data and concatenate these solutions into final features. Similar to binary searching, we divide the entire videos into

n subsets without overlapping, where $n = 1, 2, 4$ in our setting.Therefore, the total number of the pyramid subsets are $N = 1+2+4 = 7$. Then these N subsets of data are fed into the corresponding independent attention modules to search the key frames from different temporal scales. In the pyramid attention structure, the solutions will be exploited from coarser to finer scales and such binary searching scheme can largely improve the efficiency and accuracy and find what truly matters in our videos. This process can be defined as:

$$\begin{pmatrix} f_1 = Attention(X_1) \\ f_2 = Attention(X_2) \\ ... \\ f_N = Attention(X_N) \end{pmatrix}, \tag{6}$$

where X is the N divided pyramid subsets and f is the generated temporal pyramid attention features, reflecting attentions from both long-term scale and short-term scale.

Although we employ temporal pyramid structure to find optimal solutions of key frames, we can not expect one single attention module is enough for reflecting all aspects of the specific subsets of videos. As illustrated from previous research (Vaswani et al. 2017)(Lin et al. 2017), one single attention module can only reflect very limited aspects of the video, which is quite similar to convolution layers that have to use multiple kernels for better representation. Therefore, we need multiple attention modules to focus on the diversified aspects of video subsets. In each pyramid subsets in Eq. 6, we stack $M = 32$ independent attention modules to fully exploit essential frames of specific video subsets. Therefore, we have $N \times M$ attention modules to describe video. And the Eq. 6 can be extended to an attention matrix F in which each elements of the matrix is a single attention module:

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & \cdots & f_{1M} \\ f_{21} & \ddots & & & \\ f_{31} & & \ddots & & \\ \vdots & & & \ddots & \\ f_{N1} & & & & f_{NM} \end{bmatrix} \tag{7}$$

where N represents the pyramid subsets and M is the number of independent attention modules for each specific pyramid subsets. As presented in Eq. (7), we have attention matrix of $N \times M$, which can exploit key frames from pyramid subsets of videos and employing multiple independent attention modules to better reflect diversified aspects of each pyramid parts.

Kernel Aggregation Layer

Incorporating the kernel aggregation layer brings three advantages. Firstly, features from the attention matrix can be of very high dimensionality if being simply concatenated. The kernel aggregation layer can reduce the dimensionality of features. Secondly, the kernel aggregation layer is based on kernel approximation to leverage the great strength of kernels for non-linear feature learning, which maintains high non-linear discriminative ability of features. Thirdly, the kernel aggregation layer can be incorporated in an end-to-end neural network easily, which enjoys the benefits of back propagation and improves performance. The kernel aggregation layer is derived from the approximation of shift invariant kernels based on random Fourier features underpinned by the well-known Bochner's theorem.

Theorem 1 (Bochner) *A continuous shift-invariant kernel function $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$ on $\mathbb{R}^d$ is positive definite if and only if it is the Fourier transform of a unique finite non-negative measure on $\mathbb{R}^d$. Defining $\zeta_\omega(\mathbf{x}) = e^{j\boldsymbol{\omega}^\top \mathbf{x}}$, for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$,*

$$k(\mathbf{x} - \mathbf{x}') = \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}^\top (\mathbf{x}-\mathbf{x}')} d\boldsymbol{\omega} = \mathbb{E}_{\boldsymbol{\omega}}[\zeta_{\boldsymbol{\omega}}(\mathbf{x})\zeta_{\boldsymbol{\omega}}(\mathbf{x}')^*] \tag{8}$$

*where * is the conjugate and $p(\boldsymbol{\omega})$ is the Fourier transform of the kernel.*

If the kernel $k(\delta)$ is properly scaled, then its Fourier transform $p(\mathbf{w})$ will also be a proper probability distribution. Defining $\xi_{\mathbf{w}}(x) = e^{j\mathbf{w}^T \mathbf{x}}$, for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, we have:

$$k(\mathbf{x} - \mathbf{x}') = \int_{\mathbb{R}^d} p(\mathbf{w}) e^{j\mathbf{w}^T (\mathbf{x}-\mathbf{x}')} d\mathbf{w} = E[\xi_{\mathbf{w}}(\mathbf{x})\xi_{\mathbf{w}}(\mathbf{x}')^*], \tag{9}$$

where * is the conjugate and $\xi_{\mathbf{w}}(\mathbf{x})\xi_w(\mathbf{x}')^*$ is an unbiased estimate of $k(\mathbf{x} - \mathbf{x}')$ when $\mathbf{w}$ is drawn from $p(\mathbf{w})$.

The kernel $k(\mathbf{x}, \mathbf{x}')$ can be approximated by drawing d random samples as:

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{i=1}^{d} \left\langle \sqrt{\frac{2}{d}} \cos(\mathbf{w}_i^T \mathbf{x} + b_i), \sqrt{\frac{2}{d}} \cos(\mathbf{w}_i^T \mathbf{x}' + b_i) \right\rangle, \tag{10}$$

where $\mathbf{w}$ is sampled from the probability distribution $p(\mathbf{w})$, and b is uniformly sampled over $[0, 2\pi]$.

Thus, the corresponding approximated feature map takes the following form

$$\phi(\mathbf{x}) = \sqrt{\frac{2}{d}} [\cos(\mathbf{w}_i^T \mathbf{x} + b_i)]_{i=1:d}, \tag{11}$$

where $\phi(\mathbf{x})$ is called the random Fourier feature, and has been successfully used in various kernel methods.

Nevertheless, the power of kernel approximation based on random Fourier features remains largely underdeveloped, and only recently attracted attention. In the original kernel approximation methods, no adaptive learning is involved and the approximate features would be of high redundancy and of low discriminant ability, which deteriorate the performance while inducing unnecessary computational cost. In addition, approximating the kernel with a fixed configuration does not necessarily lead to high performance since it is still a very challenging task of how to choose the best kernel configuration.

Since random sampling from data-independent distributions is not optimal, we propose to approximate kernels from data in a supervised way. Specifically, we design the kernel aggregation layer, which learns parameters of $\mathbf{w}, b$ to generate more compact but highly discriminative feature representations. W and $\mathbf{b}$ are defined as $W = [\mathbf{w}_1, ..., \mathbf{w}_c] \in \mathbb{R}^{d \times c}$ and $\mathbf{b} = [b_1, ..., b_d]$, respectively. In this way, we build a

Fourier imbedding layer within neural networks and achieve nonlinear ability with cosine activations:

$$\Phi(\mathbf{x}) = \cos(W\mathbf{x} + \mathbf{b}), \tag{12}$$

where $\cos$ is the element-wise function, i indicates the i-th attention feature in the attention matrix, and W is the weight matrix of the nonlinear layer. The proposed kernel aggregation layer can be seamlessly integrated with other deep learning architectures and trained via back-propagation, which enjoys the benefits of neural networks for end-to-end learning. The outputs of kernel aggegration layers are concatenated to generate the final holistic representation for classification. In the final stage, a softmax function is used for classification.

Experiments

In this section, we first conduct our experiments on two benchmark dynamic scene datasets, i.e., YUPENN++ (Feichtenhofer, Pinz, and Wildes 2017) and Maryland (Shroff, Turaga, and Chellappa 2010) and further experiment our model on a large activity dataset called ActivityNet. Then, we discuss our experimental setups on these three datasets. We report comprehensive experimental analysis on each component in our method. This allows us to observe the performance gain of each component under various experimental conditions and achieve a deeper understanding of how these components work in our model. In the final part, the proposed work is compared with state-of-the-art methods on three datasets mentioned above, which further verifies the effectiveness of the proposed attentive temporal pyramid network (ATP-Net).

Datasets

We evaluate our proposed method on three benchmark datasets: YUPENN++ dataset (Feichtenhofer, Pinz, and Wildes 2017) , Maryland dataset (Shroff, Turaga, and Chellappa 2010) and AcivityNet dataset (Heilbron et al. 2015).

YUPENN++: YUPENN++ is a newly proposed dynamic scenes dataset which samples 20 scene classes and encompasses a wide range of conditions including variations of illumination, view-points and seasonal changes. It is extended from the earlier YUPENN dataset that has only 14 scene classes and all videos are captured without camera motions. The 20 scene classes in newly proposed YUPENN++ dataset are as follows: beach, city street, elevator, forest fire, fountain, highway, lightning storm, ocean, railway, rushing river, sky clouds, snowing, waterfall, windmill farm, building collapse, escalator, falling trees, fireworks, marathon, waving flags. And the last six classes are the newly added classes in YUPENN++. In each scene classes of YUPENN++, there are 60 color videos with half of them captured by a static camera and the other half are captured by a moving camera. The camera motions include pan, tilt, zoom and jitter. Normally, the videos captured with moving camera are much more difficult to be classified, compared with videos captured by static cameras. Except for camera motion, there still exists other varying conditions that make this dynamic scene datasets even more challenging, encompassing illuminations, seasonal, scale and viewpoint. Each video sample lasts about 5 seconds and has approximately 100-150 frames. All these samples have been resized to width of 480 pixels while preserving their original aspect ratio. Since this video dataset is much more challenging and includes samples from previous used YUPENN dataset, we use this new dataset for evaluating our method.

Maryland: Compared with YUPENN++ that has 60 samples over 20 classes, the Maryland dataset is quite small that contains 13 dynamic scene categories with 10 samples in each categorie and has 130 video samples in total. Though the number of video samples are quite small, each video samples from Maryland dataset contains much longer frames and the video length varies from 80 frames to thousands of frames. Because of the varying video length as well as camera motions and even scene cuts confounded with object motions in most samples, the Maryland dataset is still very challenging and different from the YUPENN++ dataset.

ActivityNet: Different from the previous two dynamic scene datasets, the ActivityNet mainly depicts human action or activities. It consists of 10024, 4926 and 5044 videos in training, validation and test sets, respectively. These videos are untrimmed and contain activities belonging to one of the 200 different classes. In total, it amounts to 849 hours of untrimmed videos with only 68.8 hours that are truly related to the content of activities. For each video samples, it lasts from tens seconds to up to 10 minutes and we extract video frames at 5 frames per second for training and testing. Since the video samples and classes are much larger than YUPENN++ and Maryland dataset and videos are untrimmed with lots of irrelevant background, this ActivityNet is much more difficult for classification and assigned as the benchmark dataset for Activity challenge competition from 2015 to now.

Implementation details

For constructing our model, we use the Pytorch language and all experiments are conducted on a workstation with an Intel Core I7 CPU and a NIVDIA Titan X GPU.

Data augmentation: All video samples are firstly conducted by data augmentation before feature extraction process. These video samples are resized to 256×256, cropped at random position into 224×224 and through random horizontal flipping. Since the frames of video samples vary from tens to thousands, staking all frames in the neural network is not possible and not necessary. For each video sample, we randomly sample 100 consecutive frames during training and use all frames for testing. If video frames are less than 100, we repeat the video until its frames are more than 100 for selection.

CNNs features: In order to collect frame-level features from CNNs, we extract local features at every frame of video samples after data augmentation. We use Resnet-50 model (Feichtenhofer, Pinz, and Wildes 2017) to extract our frame-level features. In two dynamic scene datasets, these Resnet-50 model are pretrained on Places 365 dataset (Zhou et al. 2016) which consist of numerous natural scene images and

Table 1: Accuracy (%) on the YUPENN++ dataset and Maryland dataset to show the effect of different attention matrix settings with $N = 1, 3, 7$ and $M = 1, 8, 32$.

	YUPENN++			Maryland		
Avg	89.53			89.23		
M	M=1	M=8	M=32	M=1	M=8	M=32
N=1	88.05	89.63	90.64	89.23	90.77	90.77
N=3	87.77	90.83	92.03	90.77	92.30	92.30
N=7	89.63	91.38	**92.03**	90.77	92.30	**93.85**

Table 2: Accuracy (%) on the YUPENN++ dataset and Maryland dataset to show the effect of kernel aggregation layer.

Datasets	YUPENN++	Maryland
Concatenation	90.05	90.77
FC layer	92.03	93.85
Kernel layer	**92.37**	**95.38**

in ActivityNet dataset that contains more human activities, the Imagenet dataset (Deng et al. 2009) is more suitable for pre-training which has large quantities of human activity images. We extract features before the last fully-connected layer for spatial representation. The extracted features are with 2048-dimension in Resnet-50 model.

ATP-Net: The training procedure of ATP-Net follows standard ConvNet training (Huang et al. 2017a) (Zhou et al. 2016), with $learningrate = 0.001$,$batchsize = 32$ and momentum learning algorithm. We adopt the commonly-used cross entropy as the loss function. To keep consistency with previous research, different training and testing strategies are used in our experiments. In the YUPENN++ dataset, we use 10% for training and 90% for testing, which is a very strict train test ratio applied in (Feichtenhofer, Pinz, and Wildes 2017). In Maryland dataset, because of very limited video samples for both training and testing, we use 50% for training and 50% for testing that is more challenging than "leave-one-out" strategy used in the previous methods (Feichtenhofer, Pinz, and Wildes 2014), to better present the improvements of our method. In Activitynet dataset, since no public testing labels are available, the performances on Activitynet datasets are all reported on validation set as it did in (Qiu, Yao, and Mei 2017).

Ablation study

We conduct an ablation study to separately evaluate the contribution of each component in our model to the overall performance.

In the ATP-Net, we have two main components, namely pyramid attention matrix and kernel aggregation layer.

Pyramid attention matrix Firstly, we focus on exploiting how pyramid attention matrix can affect the performance of dynamic scene classification. We consider average pooling as mentioned in Eq. 2 and different settings of pyramid attention matrix. To fully examine the performance, we vary the parameter of $N = 1, 3, 7$ and $M = 1, 8, 32$ in the attention matrix where N represents pyramid feature subsets we used and M is the total number of independent attention modules in a single pyramid feature subset. Thus, we will have $3 \times 3 = 9$ different attention matrix with $F_{1,1}, F_{1,8}, F_{1,32}, F_{3,1}, F_{3,8}, F_{3,32}, F_{7,1}, F_{7,8}, F_{7,32}$. The attention matrix experiments are conducted without kernel aggregation layer in order to avoid influences. Instead, we use a simple fully-connected layer with 512 hidden units and Relu activation to reduce feature dimensionality. The attention matrix is evaluated on two dynamic scene datasets. As shown in Table 1, we observe a significant performance gain between the results of using average pooling and pyramid attention matrix with $N = 7$, $M = 32$ on both YUPENN++ and Maryland datasets, which means that our ATP-Net can play an important role in this situation and exploits better representations. The accuracy rates are 2.50% higher on the YUPENN++ dataset and 4.62% higher on Maryland dataset. Since Maryland dataset contains much more moving camera situations and scene cuts, our ATP-Net can better deal with such challenging situations and improve the performance. We also observe that on both datasets, the performance is increasing with the addition of attention units. However, with the increase of pyramid structure N and number of attention modules M, the accuracy rates are becoming saturated on both sides. This indicates that more attention modules are helpful with more parameters, but it is not universal applicable. By changing the data structure to a pyramid way, the two factors N and M will generate an attention matrix which will be more beneficial. It is interesting that if we use only one single attention module, the performance is very limited or even worse, indicating that such single attention modules can only reflect partial aspects of video, especially in video of large frames (100 frames in our case).

Kernel aggregation layer: In the case of multiple pyramid attention matrix, the features can be of very high dimensionality if simply concatenated. The kernel aggregation layer is to reduce dimensionality while keeping the high nonlinear discriminative ability. We consider the effect of the kernel aggregation layer in Table 2 with the output of attention matrix at $N = 7$, $M = 32$. We compare the performance of simple concatenation, fully-connected (FC) layer with 512 hidden units and Relu activation and kernel aggregation layer. The performance is also evaluated on two dynamic scene datasets. As shown in Table 2, the kernel aggregation layer is slightly higher than FC layer with Relu in the YUPENN++ dataset and 1.53% higher than FC layer in Maryland dataset, which indicate that our aggregation can perform better in moving camera situations. The simple concatenation is of very high-dimensionality and poor in performance.

Comparison on dynamic scene classification

In this section, we compare our proposed ATP-Net with other state-of-the-art methods to further illustrate the effectiveness of our model. We use three benchmark datasets for evaluation, that is, the Maryland, YUPENN++ and Activitynet datasets.

YUPENN++: For fair comparison and to benchmark with previous methods (Feichtenhofer, Pinz, and Wildes 2017),

Table 3: Performance comparison of different algorithms on the YUPENN++, Maryland and Activitynet datasets.

Datasets	YUPENN++	Maryland	Activitynet
SFA	56.9	60.0	-
BoSE	77.0	77.7	-
IDT	85.6	-	64.7
C3D	84.0	87.7	65.8
Resnet-50	85.9	85.5	65.3
T-Resnet	89.0	-	-
P3D Resnet	-	94.6	**75.1**
Single Attention	88.0	89.2	66.7
Attention Matrix	92.0	93.9	72.4
ATP-Net	**92.3**	**95.4**	74.6

we use random split with fixed training/testing ratio of 10/90 in the YUPENN++ dataset, which means we use 10% video samples from each classes for training and 90% video samples for testing. This training/testing ratio makes this YUPENN++ dataset very challenging. As shown in Table 3, we select 6 best performing methods in dynamic scene classification field for comparison, including SFA (Theriault, Thome, and Cord 2013), BoSE (Feichtenhofer, Pinz, and Wildes 2014), IDT (Wang and Schmid 2013), C3D (Tran et al. 2015), Resnet(He et al. 2016) (pre-trained on Imagenet), T-Resnet (Feichtenhofer, Pinz, and Wildes 2017). Among these methods, only Resnet is a purely spatial algorithm which simply aggregates frame-level features with average pooling strategy. However, it is interesting to see that Resnet achieves very competitive performance on the YUPENN++ dataset. This is owing to the powerful feature learning of deep neural networks. Also, this result reflects that spatial information plays an essential role in dynamic scene classification task.

The top performing algorithm is the newly proposed ATP-Net. We also list the results of a single-attention model, attention matrix model for comparison. As can be concluded from Table 3, all our proposed methods have obtained very promising performance. It is particularly interesting to compare our methods to Resnet-50 since our ATP-Net is constructed based on this CNN architecture. Even with a very small number of training samples, e.g., 10% of the total dynamic scene dataset, our ATP-Net still generates distinctive representations and improves the performance largely. And the addition of attention as well as pyramid attention matrix can largely enhance the dynamic scene classification and outperform other spatiotemporal methods such as previous best BoSE and T-Resnet.

Maryland: For the Maryland dataset, we use more strict train/test split with only 50% samples for training and 50% for testing. And compared with the YUPENN++ dataset, the Maryland dataset is even more challenging because of severe camera motions. As can be seen from Table 3, similar to our model, P3D Resnet (Qiu, Yao, and Mei 2017) is based on a Resnet-152 network, and has obtained 94.6% performance. But our ATP-Net benefits from the pyramid attention structure and the newly designed kernel aggregation layer, that even with much less training samples our method still achieves the top performance among these state-of-the-art algorithms. This suggests that our ATP-Net can quite well handle severe moving camera situations.

Activitynet: Different from previous two dynamic scene datasets, the ActivityNet dataset mainly depicts human action or activities. This dataset has large quantities of both training and testing samples and regarded as the benchmark dataset for Activity challenge competition from 2015 to now. The video samples from this dataset is untrimmed which inevitably introduce much more noise and irrelevant background, which makes it very difficult for classification. We extend our model to this dataset, in order to evaluate the performance of ATP-Net under the large untrimmed video dataset. As can be seen from Table 3, even we do not achieve the top performance, we still get high performance. Since P3D Resnet uses Resnet-152 network to construct their model, it presents slightly higher performance to our ATP-Net that uses Resnet-50 network. And compared with the baseline method, Resnet-50, our ATP-Net constructed on Resnet-50 can improve the performance by 9%, indicating that the proposed attentive pyramid is very helpful for untrimmed video classification.

Conclusion

In this paper, we have presented the attentive temporal pyramid network (ATP-Net) for dynamic scene classification. The ATP-Net extracts multi-scale attentive features from a temporal pyramid attention matrix and aggregates these features by a newly designed kernel aggregation layer to achieve highly discriminative and compact representations. The ATP-Net has been extensively evaluated on three benchmark datasets for dynamic scene classification. Experimental results have shown that ATP-Net achieves the new state-of-the-art performance on the benchmarks and substantially outperforms previous methods.

Acknowledgments

This paper was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1200100, the National Science Fund for Distinguished Young Scholars under Grant 61425014 and National Science Foundation of China under Grant 61871016.

References

Ba, J.; Mnih, V.; and Kavukcuoglu, K. 2014. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K.; and Li, F. F. 2009. Imagenet: A large-scale hierarchical image database. *2009 IEEE International Conference on Computer Vision and Pattern Recognition* 248–255.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2625–2634.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. 2014. Bags of spacetime energies for dynamic scene recognition. In *2014, IEEE Conference on Computer Vision and Pattern Recognition*, 2681–2688.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. 2016. Dynamic scene recognition with complementary spatiotemporal features. *IEEE Transactions Pattern Anal. Mach. Intell.* 38(12):2389–2401.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Temporal residual networks for dynamic scene recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4728–4737.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *2016, IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Computer Vision and Pattern Recognition*, 961–970.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017a. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, 3.

Huang, Y.; Cao, X.; Zhang, B.; Zheng, J.; and Kong, X. 2017b. Batch loss regularization in deep learning method for aerial scene classification. In *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2017*, 3E2–1. IEEE.

Huang, Y.; Cao, X.; Wang, Q.; Zhang, B.; Zhen, X.; and Li, X. 2018. Long-short term features for dynamic scene classification. *IEEE Transactions on Circuits and Systems for Video Technology*.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Mnih, V.; Heess, N.; Graves, A.; et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, 2204–2212.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 5534–5542. IEEE.

Shroff, N.; Turaga, P.; and Chellappa, R. 2010. Moving vistas: Exploiting motion for describing scenes. In *2010 IEEE International Conference on Computer Vision and Pattern Recognition*, 1911 – 1918.

Simonyan, K., and Zisserman, A. 2014a. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, 568–576.

Simonyan, K., and Zisserman, A. 2014b. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Theriault, C.; Thome, N.; and Cord, M. 2013. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2603–2610.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, 4489–4497.

Vasudevan, A. B.; Muralidharan, S.; Chintapalli, S. P.; and Raman, S. 2013. Dynamic scene classification using spatial and temporal cues. In *IEEE Conference on Computer Vision Workshops*, 803–810.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 6000–6010.

Wang, H., and Schmid, C. 2013. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*.

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 6450–6458. IEEE.

Yan, C.; Xie, H.; Liu, S.; Yin, J.; Zhang, Y.; and Dai, Q. 2018a. Effective uyghur language text detection in complex background images for traffic prompt identification. *IEEE transactions on intelligent transportation systems* 19(1):220–229.

Yan, C.; Xie, H.; Yang, D.; Yin, J.; Zhang, Y.; and Dai, Q. 2018b. Supervised hash coding with deep neural network for environment perception of intelligent vehicles. *IEEE transactions on intelligent transportation systems* 19(1):284–295.

Yang, J.; Ren, P.; Chen, D.; Wen, F.; Li, H.; and Hua, G. 2017. Neural aggregation network for video face recognition. *arXiv preprint*.

Zhou, B.; Khosla, A.; Lapedriza, À.; Torralba, A.; and Oliva, A. 2016. Places: An image database for deep scene understanding. *CoRR* abs/1610.02055.