Counting Grid Aggregation for Event Retrieval and Recognition

Zhanning Gao*
Xi’an Jiaotong University
gaozn1990@stu.xjtu.edu.cn

Gang Hua
Microsoft Research
ganghua@microsoft.com

Dongqing Zhang
Microsoft Research
dongqing@microsoft.com

Jianru Xue
Xi’an Jiaotong University
jrxue@mail.xjtu.edu.cn

Nanning Zheng
Xi’an Jiaotong University
nnzheng@mail.xjtu.edu.cn

Abstract

Event retrieval and recognition in a large corpus of videos necessitates a holistic fixed-size visual representation at the video clip level that is comprehensive, compact, and yet discriminative. It shall comprehensively aggregate information across relevant video frames, while suppress redundant information, leading to a compact representation that can effectively differentiate among different visual events. In search for such a representation, we propose to build a spatially consistent counting grid model to aggregate together deep features extracted from different video frames. The spatial consistency of the counting grid model is achieved by introducing a prior model estimated from a large corpus of video data. The counting grid model produces an intermediate tensor representation for each video, which automatically identifies and removes the feature redundancy across the different frames. The tensor representation is subsequently reduced to a fixed-size vector representation by averaging over the counting grid. When compared to existing methods on both event retrieval and event classification benchmarks, we achieve significantly better accuracy with much more compact representation.

1. Introduction

The emergence of smartphones and wearable devices and growing practice of video sharing in social media have led to enormous explosion of user-generated videos on the Internet. Such video data record various events related to both people’s private and social life. To facilitate end users to conveniently search or recognize a video event from a large corpus of videos, it is essential to build a comprehensive, compact, and yet discriminative representation at the video-clip level.

*This work was done when Z. Gao was an intern at MSR supervised by G. Hua.
considering information across all shots.

For instance, as shown in Figure 1, the frames of the anchors dominate in the event video related to “Dominique Strauss-Kahn arrested”. Since those frames share similar content, simply averaging frame-level descriptors may lead to over count those descriptors and reduce the discriminative-ness of the video representations. Therefore, it is crucial to balance the influence of frame features at the aggregation step.

We shall point out that there are also previous work which represent a video as a set of frame-level descriptors over a set of key-frames \[14, 13\], which can partially address the over-counting issue by selecting one key-frame from each shot \[13, 14\]. However, for a slowly and smoothly changed shot, it is difficult to select the key frames to represent the whole shot. Besides, the resulting representation is a set of features, whose size varies with the number of key-frames extracted, which makes both storage and matching to be difficult.

In order to ease the issue of over-counting while maintaining a fixed-size video-level representation, we propose a novel aggregation method for the frame-level features to balance the contribution of each frame-level feature in the video representation. Instead of considering the video as a set of independent frames \[28\], we employ a principled counting grid (CG) model \[20\] to capture the interdependence in frame features. The original CG model is proposed in \[20\] for image and scene classification tasks.

After building a CG model with a set of video frames, each individual video frame in the set can be regarded as a “window” inside the scene represented by the counting grid \[20\]. And similar video frames are aggregated to the same or similar locations on the counting grid. Based on these properties of a CG model, instead of directly averaging the frame-level descriptors, as the main contribution of this paper, we propose to aggregate the learned counting grid to form the final video representation. Our frame feature is extracted by a deep CNN model pre-trained on large image dataset. After normalizing the CNN features, we regard these normalized features as counting vectors over visual primitives, and hence fit the CG model.

One shortcomings of the original CG model is that the spatial arrangement of the CG map varies with different initialization in the estimation. Although statistically the different CG maps with different spatial arrangements are equivalent, it is inconvenient to be used for building video representations, as the video representations should be uniquely determined. To resolve this issue, we propose to learn a prior CG model from a large corpus of video data, and then build the CG map for each video by adapting from this prior model. This way, we achieve consistency in the spatial arrangement of the CG maps across different videos.

The resulting video representation of our method can be as compact as 256 dimension vector. We achieve state-of-the-art retrieval and recognition accuracy on two large-scale video datasets, while maintaining top efficiency both in terms of memory usage and retrieval/recognition speed.

2. The counting grid model

We first briefly introduce the counting grid (CG) model, which was originally proposed by Prina and Jojic \[20\] for analyzing images collections. It assumes that each image is represented by a histogram \(\{c_z\}\), e.g., using a bag of features representation, where \(c_z\) denotes the counts of features \(z\).

Formally, the basic counting grid \(\pi_{1,z}\) is a set of normal-
Figure 2. Illustration of the location distribution $q(l^t = k)$ of each frame. (a) The location distribution of three frames sampled from the video. (b) The frequency distribution obtained by accumulating all the location distribution together. The counting grid is trained with pool$_{3}$ descriptor.

ized counts of features indexed by $z$ on the 2D discrete grid $i = (i_x, i_y) \in E = [1 \ldots E_x] \times [1 \ldots E_y]$, where $i$ is an index that addresses a generic location on the grid and $\sum_z \pi_{i,z} = 1$. A given bag of features $\{c_z\}$ is generated from the counting grid by selecting a location $k$, calculating the distribution $h_k = W^{-1} \sum_{i \in W_k} \pi_i$ form the window $W_k$ in the location $k$ of the counting grid and then generating the bag $\{c_z\}$ with distribution $h_k$.

In other words, the position $k$ of the window $W_k$ in the grid is a latent variable. The probability of generating the bag of features $c = \{c_z\}$ in location $k$ is

$$p(c | l = k) = \prod_z (h_{k,z})^{c_z} = \mu \prod_z \left( \sum_{i \in W_k} \pi_{i,z} \right)^{c_z},$$

where $\mu$ is the normalization constant. Thus, the joint distribution over a training set of bags of features $\{c^t\}$, indexed by $t$, and their corresponding latent window locations $\{l^t\}$ in the grid can be derived as

$$P(\{c^t\}, \{l^t\}) \propto \prod_t \prod_z \left( \sum_{i \in W_k} \pi_{i,z} \right)^{c_z}.$$  

(2)

The counting grid $\pi$ is estimated by maximizing the log likelihood of the joint distribution with an EM algorithm, i.e.,

$$E \text{ step : } q(l^t = k) \propto \exp \left( \sum_z c^t_z \log h_{k,z} \right),$$

$$M \text{ step : } \pi_{i,z} \propto \pi_{i,z}^{\text{old}} \sum_t \left( \sum_{k | i \in W_k} q(l^t = k) \right),$$

(3)

where $q(l^t = k)$ denotes the posterior probabilities $p(l^t = k | c^t)$ and $\pi_{i,z}^{\text{old}}$ is the counting grid at the previous iteration.

After learning the counting grid with a set of training images or video frames, we obtain two outputs from the EM algorithm. The first is the counting grid $\pi$ which is able to generate all the training features. The second output is the location distribution $q(l^t = k)$ of each training image/video frame on the counting grid.

In the next section, by analyzing the location distribution, we will show that similar images (or frames from the video in our case) are usually placed in the same or nearby locations. This is expected since the distribution $h_k$ from different locations are highly tied and similar features would be more likely to be generated by the same or similar distribution $h_k$, which is calculated from the same window or overlapped windows on the counting grid.

3. Counting grid aggregation

3.1. Learning spatially consistent counting grid with a prior

Different from previous work on the counting grid models [20, 18, 12, 19], we aim at using counting grid to represent the event videos. For algorithms which build video representations, it is required that the same input video should have an unique representation output.

However, the counting grid learned form an input video (considering the video as a set of images/frames) may be different for the same video in another trial because of random initialization. To address this problem, we propose to use a pre-trained counting grid to initialize and guide the learning process.

In other words, the learning process will be kicked off from the same initial counting grid for all videos, constrained by a Dirichlet prior with parameters $\beta \pi^{ref}$, where $\pi^{ref}$ denotes the pre-trained counting grid, and $\beta$ is a
weighting factor to adjust the influence of the prior in the learning process.

By considering the prior, the joint distribution over \( \{ c^t \} \) and \( \{ l^t \} \) can be modified as

\[
P(\{ c^t \}, \{ l^t \}) \propto \left( \prod_t \sum_k \sum_z \pi_{i,z} \right)^{c^t} \prod_k \prod_z (\pi_{k,z})^{\beta \pi_{rcf}^{l^t = k}. (4)}
\]

Based on this joint distribution, the M step should be modified as

\[
M \text{ step : } \pi_{1,z} \propto \beta \pi_{1,z}^{rcf} + \pi_{1,z}^{old} \sum_t \left( c^t \sum_{k|\bar{W}_k} q(l^t = k) \right) h_{k,z}. (5)
\]

In our implementation, the pre-trained counting grid \( \pi^{rcf} \) is learned from a large set of video frames sampled from all the videos in the database. This is different from the pseudo counts \( \eta \) in [20], which was set to a constant value without training to prevent overtraining and numerical problems.

### 3.2. CNN features on counting grid

Recently, image descriptors based on the activations within deep convolutional neural networks (CNN) have emerged as state-of-the-art generic descriptors for visual recognition [24, 7]. For example, Xu et al. [29] directly use the CNN features as frame-level descriptors and aggregate them with VLAD to produce video-level representation, which significantly improves the state-of-the-art event detection results on the TRECVID MED 13 and 14 datasets.

Our experiments also confirm the superior performance of CNN features on the event retrieval task. To further improve the accuracy for event retrieval, we propose to employ the CG model to enhance the CNN-feature based video representation. Since the CG model is designed for histogram features, such as bag of features, we explain why the CNN features, in particular the activations of the first fully-connected layer and the last convolutional layer after rectified linear unit of a pre-trained AlexNet [15], may also be adopted for the CG model. Below we refer to those two CNN features as \( \text{pool}_5 \) and \( \text{fc}_6 \) descriptors.

Since both the \( \text{pool}_5 \) and \( \text{fc}_6 \) features are outputs from a rectified linear unit layer, each dimension of these two CNN features is strictly larger than or equal to zero. If we treat each dimension of the CNN features as a response to a visual primitive, then the response in each dimension can be treated as the counts of the corresponding visual primitives, which is actually activated by the corresponding convolutional filters that define the visual primitives. Therefore, with appropriate normalization, we can also regard the CNN feature as a bag of visual primitives representation.

To further demonstrate how the normalized CNN features are interacting with the CG model, in Figure 2, we visualize the location distribution \( q(l^t = k) \) of each frame in a CG model trained from a video. The counting grid is trained with \( \text{pool}_5 \) descriptors, where we simply average the feature maps from the \( \text{pool}_5 \) layer as the \( \text{pool}_5 \) descriptor to represent the frame.

We highlight the location on the counting grid of each frame with maximum probability of \( q(l^t = k) \) (Figure 2(a)). The horizontal axis denotes all the frames along the timeline and the vertical axis denotes the location of the counting grid. We extend the 2D location map to 1D following the raster-scan order for convenience in visualization. The same color means the frames are assigned to the same or similar location on the counting grid.

We also sample three frames from the video and plot the distribution of each frame as shown in Figure 2(a). Their distributions look like several discrete impulse functions combined together. As a result, each frame is assigned to one of the locations with high probability (close to 1). After accumulating all the location distributions together, we can obtain the frequency distribution (as shown in Figure 2(b)) of the frames assigned on each location.

Interestingly, this frequency distribution can be actually thought as a bag of “location” vector considering the location distribution of each frame. Since similar frames usually come from the same shot or recurrent shots and share the same or similar location distribution as shown in Figure 2 they are aggregated at the same or similar locations. In the meanwhile, according to Eq. [5] all the normalized CNN features of the frames aggregated on the same location is smoothly merged during the training step.

### 3.3. Aggregating counting grid for video representation

Our counting grid aggregation method has two steps: generating the active map and aggregating the counting grid descriptors as the video representation. We denote \( h_k = \frac{1}{W_k} \sum_{i | \bar{W}_k} \pi_i \) at the location \( k \) as the counting grid descriptor. The active map is employed to filter out the noisy counting grid descriptors which associate with the locations that no frame is assigned.

#### 3.3.1 Generating the active map

The active map is generated by converting the frequency distribution of the frames in a video to a binary map with a certain threshold. Formally, we use \( A = \{ a_k | k \in E \} \), \( a_k \in \{0, 1\} \) to denote the active map. Thus, the active map can be computed as

\[
a_k = \begin{cases} 
1 & \sum_{t=1}^{N} q(l^t = k) > \tau \\
0 & \text{otherwise}
\end{cases}
\]



where $\tau$ is the threshold. The influence of $\tau$ will be further studied and discussed in the experiments section.

### 3.3.2 Constructing the final video representation

After generating the active map, the second step is quite simple, i.e., applying sum-aggregation over the whole activated counting grid descriptors to produce the final video representation. Formally, our counting grid aggregation can be derived as

$$\phi_{CGA}(\pi, A) = \frac{\sum_{k \in E} a_k \left( \sum_{i \in W_k} \pi_i \right)}{\sum_{k \in E} a_k}.$$  

The obtained representation $\phi_{CGA}(\pi, A)$ is subsequently power normalized [22] and PCA-whitened [2, 8] and then $l_2$-normalized.

Figure 3 illustrates the framework of the counting grid aggregation method. After learning the counting grid for the video, similar frames are aggregated to the same or nearby locations. In addition, as shown in Figure 3, the frames aggregated at the same or nearby locations on the counting grid are usually from the same shot. Therefore, the activated counting grid descriptors can be thought as the representations of the shots from the video. As a result, sum-aggregate the activated counting grid descriptors will balance the contribution of each shot to the final video representation. We will show that this aggregation strategy can enhance the video representation for both the retrieval and recognition tasks.

### 4. Experiments

In this section, we first evaluate the counting grid aggregation method on an event retrieval benchmark, i.e., the EVVE dataset [25]. In addition, we also explore the performance of our compact video representation on a recognition task with the FCVID dataset [11], one of the largest datasets for video categorization with accurate manual annotations.

#### 4.1. Datasets and evaluation protocol

In this section, we briefly introduce the datasets and their evaluation protocol.

The EVVE dataset [25] is dedicated to the retrieval of particular events which differs from recognizing event categories as in the TRECVID MED task [1]. It contains 2,995 videos (620 videos are set as query) related to 13 specific event classes such as the “Concert of Madonna in Rome 2012”, and “The wedding of Prince William and Kate Middleton”, etc. Given a single video of an event, it aims at retrieving videos related to the same event from the dataset.

The evaluation is performed in a standard retrieval scenario. The mean AP (mAP) is computed per event. Then the

http://pascal.inrialpes.fr/data/evve/
http://bigvid.fudan.edu.cn/FCVID/
over all performance is evaluated by averaging the mAPs over the 13 events. In addition, a large distractors dataset (100,000 videos) is also provided to evaluate the retrieval performance on large-scale data.

The FCVID dataset contains 91,223 Web videos annotated manually to 239 categories. The categories in FCVID cover a wide range of topics such as social events (e.g., tailgate party), procedural events (e.g., making a cake), objects (e.g., panda), scenes (e.g., beach), etc. We use the same evaluation protocol as presented in [1] to evaluate our proposed video representation. The mean AP (mAP) is reported at last.

4.2. Implementation details

Frame-level descriptor. Given an input video, we sample 5 frames per second (5 fps) to extract the CNN features. We employ the pre-trained CNN model AlexNet [15] and adopt the output from the pool5 and fc6 layers (after ReLU) as the frame-level descriptors. The pool5 feature map is of size $6 \times 6 \times 256$, we average the feature map to produce the 256 dimensions vectors as the pool5 descriptor. For the fc6 layer, we directly employ the output vector after ReLU (4096 dimensions) as the fc6 descriptor. In addition, we average all the pool5 or fc6 descriptors over the video, i.e., sum-aggregation, as the baseline to evaluate our counting grid aggregation method.

Post-processing. For the baseline CNN feature based video representation, we apply the same post-processing strategy as in [2], i.e., the representation vector of a video is first $l_2$-normalized, and then PCA-whitening [8] performed and $l_2$-normalized again. For our video representations, power normalization shows better results than $l_2$-normalization in our experiments. Therefore, our video representation vector is first power normalized, then PCA-whitening performed, and $l_2$-normalized.

Re-ranking methods for event retrieval. For the event retrieval task on the EVVE dataset, we employ two variants of query expansion methods presented by Douze et al. [5] for global video representation, i.e., Average query expansion (AQE) and Difference of neighborhood (DoN) to further boost the event retrieval performance. In our experiments, we set $N_1 = 10$ for AQE and $N_1 = 10$, $N_2 = 2000$ for DoN.

Video event recognition. We evaluate the performance of our compact video representation on recognition task with linear SVM implemented by the LibLinear library [6].
Table 1. Influence of PCA-whitening operation for sum-aggregation and counting grid aggregation. “no PCA-w” denotes the video representation without PCA-whitening.

| Method       | Sum-fc6 | Sum-pool5 | Sum-(pool5+fc6) | CGA-fc6 | CGA-pool5 | CGA-(pool5+fc6) |
|--------------|---------|-----------|-----------------|---------|-----------|-----------------|
| no PCA-w     | 36.2    | 33.8      | 35.8            | 37.6    | 36.5      | 38.9            |
| PCA-w        | 38.4    | 38.3      | 40.6            | 41.4    | 42.6      | 45.5            |

We utilize cross-validation to chose the regularization coefficient $c$ in linear SVM.

4.3. Impact of parameters

Threshold of the active map. Figure 4(a) shows the retrieval performance with different threshold of the active map. We can observe that slightly large $\tau$ achieves better performance because it can filter out some very short shots (with only several frames) which are usually not that meaningful. We set $\tau = 4$ in the subsequent experiments.

Counting grid size. To evaluate the influence of counting grid size, we first fix the window size (W = 8) of the counting grid. Then we chose 7 different counting grid size to perform the counting grid aggregation. The performance of each size is presented in Figure 4(b). We can observe that no further improvement can be obtained when $E > 16$. Therefore, the size of counting grid is fixed to 16 for computation efficiency in our experiments.

Power normalization. The impact of power normalization exponent $\alpha$ is shown in Figure 5(a). We can see the different effects of power normalization for fc6 and pool5 descriptors. The counting grid aggregation based on pool5 descriptor achieves better results with lower exponent $\alpha$ yet fc6 descriptor based representation prefers higher exponent $\alpha$. Therefore, we set $\alpha = 0.8$ and 0.2 for CGA-fc6 and CGA-pool5 in the subsequent experiments, respectively.

PCA-whitening. Table 1 shows the performance gain from PCA-whitening. Consistent improvement is obtained for both sum-aggregation and counting grid aggregation. For instance, the retrieval performance (mAP) of sum-aggregation based on pool5 descriptor (Sum-pool5) is improved from 33.3 to 38.3 after PCA-whitening. And for counting grid aggregation, e.g., CGA-pool5, the result is improved from 36.5 to 42.6. To obtain a compact video representation, we also apply dimension reduction for fc6 based representation jointly with the PCA-whitening operation [8]. Figure 5 shows the influence of dimension reduction. We reduce the dimension of fc6 based representation (CGA-fc6 and Sum-fc6) to 2048 dimension for a trade-off between compactness and performance.

4.4. Evaluation results of the event retrieval task

We evaluate the performance of the counting grid aggregation on the specific event retrieval task. The sum-aggregation, i.e., producing the video representation by averaging all the frame-level descriptors, is adopted as baseline to evaluate our aggregation method.

4.4.1 Compared with sum-aggregation

Table 2 shows the retrieval performance with different aggregation methods. We shall note that sum-aggregation with CNN feature (Sum-pool5 and Sum-fc6) already achieves state-of-the-art result compared with previous work [25, 5]. Although VLAD can also improve the retrieval accuracy, it leads to much higher dimension of the video representation. Our aggregation method can improve the retrieval performance without increasing the dimension of the video representation for both fc6 and pool5 frame descriptors.

Table 3 presents the retrieval performance for each event class of EVVE. We can see that fc6 and pool5 based representation are complementary. For instance, the pool5 based representation (Sum-pool5 and CGA-pool5) achieves bet-
Table 3. Retrieval performance (mAP) per event.

| Event ID | Sum-fc6pca | Sum-pool5 | Sum-(pool5+fc6pca) | CGA-fc6pca | CGA-pool5 | CGA-(pool5+fc6pca) |
|----------|------------|-----------|-------------------|------------|----------|-------------------|
| ♯1       | 64.3       | 71.4      | 71.4              | 67.5       | 78.2     | 79.2              |
| ♯2       | 54.7       | 48.5      | 54.8              | 56.2       | 49.6     | 57.5              |
| ♯3       | 10.3       | 10.1      | 11.1              | 15.5       | 17.6     | 19.5              |
| ♯4       | 54.5       | 53.4      | 54.9              | 51.2       | 55.7     | 55.0              |
| ♯5       | 29.1       | 26.7      | 29.0              | 29.3       | 27.3     | 30.0              |
| ♯6       | 26.4       | 22.9      | 26.0              | 25.7       | 26.0     | 28.3              |
| ♯7       | 23.3       | 19.3      | 22.0              | 22.0       | 21.7     | 23.5              |
| ♯8       | 14.9       | 15.4      | 16.7              | 20.9       | 17.0     | 23.3              |
| ♯9       | 13.7       | 8.7       | 12.1              | 46.2       | 53.5     | 56.6              |
| ♯10      | 40.6       | 42.8      | 45.9              | 32.7       | 29.6     | 33.7              |
| ♯11      | 30.8       | 31.2      | 32.2              | 83.9       | 82.6     | 86.2              |
| ♯12      | 84.5       | 83.3      | 86.0              | 74.5       | 80.1     | 81.8              |
| ♯13      | 74.1       | 63.7      | 71.5              | 41.6       | 42.6     | 45.5              |

Table 4. Performance on large-scale dataset (EVVE+100K distractors) and comparison with other methods.

| Method                  | EVVE | EVVE+100K |
|-------------------------|------|-----------|
| MMV [25]                | 512  | 33.4      |
| CTE [25]                | 35.2 | 20.2      |
| MMV+CTE [25]            | 37.6 | 25.4      |
| Hyper-pooling (SSC) [5] | 16384| 36.3      |
| Sum-(pool5+fc6pca)      | 2304 | 41.0      |
| CGA-pool5               | 256  | 42.6      |
| CGA-fc6pca              | 2048 | 41.6      |
| CGA-(pool5+fc6pca)      | 2304 | 45.5      |

4.4.2 Performance on large-scale video dataset

As shown in Table 4, consistent improvement of counting grid aggregation is observed on the large-scale dataset (EVVE+100K distractors) compared with other method. After merging with 100K distractors, the mAP of CGA-(pool5+fc6pca) achieves 33.8 which is better than the baseline (mAP = 28.7) and Hyper-pooling [5] (mAP = 26.5). Different from [25] which combines the two video represen-
tations (MMV and CTE) by adding the normalized scores of each method at the ranking step, our CGA-(pool5+fc6pca) is only a single global representation for a video.

4.4.3 Combined with query expansion

As shown in Table [5], our video representation generated by counting grid aggregation is compatible with the query expansion strategies (AQE and DoN) proposed in [5]. These methods can further improve the retrieval accuracy at the re-ranking stage. However, a different observation in our experiments is that, for our video representation, no significant improvement is obtained with DoN compared with AQE (the mAP is improved from 45.5 to 52.5 by AQE and 53.5 by DoN), which is far more efficient.

4.5. Evaluation results on video recognition

Besides the event retrieval task, we also evaluate our method with a fully labeled video dataset (FCVID [11]) to explore the potential of our method for the video recognition or other applications. Table 5 presents the classification performance of the video representations generated by sum-aggregation and counting grid aggregation. For the methods from [11], “Audio”, “Motion” and “Static CNN” both use individual feature to train a neural network for classification. “Motion” employs the improved dense trajectory descriptor [27] and “Audio” combines MFCCs [4] and Spectrogram SIFT (sgSIFT) [29]. For “Static CNN”, the CNN feature (fc7 layer) is employed as video representation. “DNN” combines the CNN feature with motion and audio features to improve the performance. “rDNN-F” further improves the performance by exploring the relationships among the features. Different with those methods, we simply use the lin-
Table 5. Retrieval performance combined with query expansion (AQE and DoN).

| Method                  | EVVE  | AQE  | DoN  | EVVE+100K | AQE  | DoN  |
|-------------------------|-------|------|------|-----------|------|------|
| Hyper-pooling (SSC)     | 36.3  | 38.9 | 44.0 | 26.5      | 30.1 | 33.1 |
| Sum-(pool5+fc6pca)      | 41.0  | 47.1 | 48.5 | 28.7      | 35.7 | 36.4 |
| CGA-(pool5)             | 42.6  | 50.1 | 51.4 | 30.1      | 38.7 | 39.8 |
| CGA-(fc6pca)            | 41.6  | 48.3 | 49.4 | 29.0      | 36.7 | 36.7 |
| CGA-(pool5+fc6pca)      | 45.5  | 52.5 | **53.5** | 33.8 | 41.3 | **42.3** |

Table 6. Performance for video recognition. “MA +” denotes that the motion and audio features are combined with the representations proposed in this paper.

| Method                  | mAP  | Method                  | mAP  |
|-------------------------|------|-------------------------|------|
| Audio (Jiang et al. 2015) | 26.1 | CGA-(pool5)             | 59.0 |
| Motion (Jiang et al. 2015) | 62.8 | CGA-(fc6pca)            | 61.0 |
| Static CNN (Jiang et al. 2015) | 63.8 | CGA-(pool5+fc6pca)      | **67.0** |
| DNN (Jiang et al. 2015) | 72.1 | MA + Sum-(pool5)        | 73.8 |
| rDNN-F (Jiang et al. 2015) |      | MA + CGA-(pool5)        | 73.8 |
| Sum-(pool5)             | 46.6 | MA + CGA-(fc6pca)       | 73.6 |
| Sum-(fc6pca)            | 53.8 | MA + Sum-(fc6pca)       |      |
| Sum-(pool5+fc6pca)      | 59.5 | MA + Sum-(pool5+fc6pca) |      |
| MA + Sum-(pool5)        | 68.9 | MA + CGA-(pool5+fc6pca) |      |
| MA + Sum-(fc6pca)       | 70.1 | MA + CGA-(pool5+fc6pca) | **76.5** |

We can observe that counting grid aggregation generates better video representation for recognition task compared with sum-aggregation. The mAP of CGA-(pool5+fc6pca) is **67.0** compared with 59.5 for Sum-(pool5+fc6pca). We also achieve better result based on CNN feature compared with “Static CNN” (mAP = 63.8). In addition, to further boost the recognition performance, we also combine the motion and audio features provided by the FCVID dataset with the late fusion strategy [11], i.e., the prediction result of each feature is combined to obtain the final categorization result. We obtain better result with MA + CGA-(pool5+fc6) (mAP = 76.5) compared with “DNN” (mAP = 72.1) and “rDNN-F” (mAP = 75.4) which also combine the motion and audio information.

5. Conclusions

In this paper, we propose an unsupervised aggregation method using a principled probabilistic model, i.e., the counting grid model, to compute memory efficient video-clip level representation for video event retrieval and recognition. It comprehensively exploits relevant information from all video frames while suppresses redundant information. We evaluate our proposed video representation on large-scale video event retrieval and recognition benchmark datasets, which outperforms existing methods both in terms of retrieval/recognition accuracy and speed. Our future work will focus on how to conduct end-to-end learning of the counting grid model and fine-tuning the CNN parameters in an unsupervised fashion to further enhance the efficacy of the proposed video representation.

References

[1] R. Aly, R. Arandjelovic, K. Chatfield, M. Douze, B. Fernando, Z. Harchaoui, K. Mcguiness, N. O’Connor, D. Oneta, O. Parkhi, et al. The axes submissions at trecvid 2013, 2013.
[2] A. Babenko and V. Lempitsky. Aggregating deep convolutional features for image retrieval. arXiv preprint arXiv:1510.07493, 2015.
[3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In ECCV, pages 584–599, 2014.
[4] M. Baillie and J. M. Jose. Audio-based event detection for sports video. In Image and Video Retrieval, pages 300–309, 2003.
[5] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyper-pooling and query expansion for event detection. In ICCV, pages 1825–1832, 2013.
[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. JMLR, 9:1871–1874, 2008.
[7] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In ECCV, pages 392–407, 2014.
[8] H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening. In ECCV, pages 774–787. 2012.
[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In CVPR, pages 3304–3311. IEEE, 2010.
[10] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In CVPR, pages 3310–3317, 2014.
[11] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. arXiv preprint arXiv:1502.07209, 2015.
[12] N. Jojic and A. Perina. Multidimensional counting grids: Inferring word order from disordered bags of words. arXiv preprint arXiv:1202.3752, 2012.
[13] A. Joly, C. Frélicot, and O. Buisson. Robust content-based video copy identification in a large reference database. In Image and Video Retrieval, pages 414–424. 2003.
[14] A. Karpenko and P. Aarabi. Tiny videos: A large data set for nonparametric video retrieval and frame classification. TPAMI, 33(3):618–630, 2011.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, pages 1097–1105, 2012.
[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91–110, 2004.
[17] P. Over, J. Fiscus, G. Sanders, D. Joy, M. Michel, G. Awad, A. Smeaton, W. Kraaij, and G. Quénot. Trecvid 2014–an overview of the goals, tasks, data, evaluation mechanisms and metrics. In Proceedings of TRECVID, page 52, 2014.
[18] A. Perina and N. Jojic. Image analysis by counting on a grid. In CVPR, pages 1985–1992, 2011.
[19] A. Perina and N. Jojic. Spring lattice counting grids: Scene recognition using deformable positional constraints. In ECCV, pages 837–851. Springer, 2012.
[20] A. Perina and N. Jojic. Capturing spatial interdependence in image features: the counting grid, an epitomic representation for bags of features. TPAMI, 37(12):2374–2387, 2015.
[21] F. Perronnin and D. Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In CVPR, pages 3743–3752, 2015.
[22] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In ECCV, pages 143–156. Springer, 2010.
[23] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei. Learning temporal embeddings for complex video analysis. In ICCV, pages 4471–4479, 2015.
[24] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In CVPR Workshops, pages 806–813, 2014.
[25] J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In CVPR, pages 2459–2466, 2013.
[26] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. IJCV, 105(3):222–245, 2013.
[27] H. Wang and C. Schmid. Action recognition with improved trajectories. In ICCV, pages 3551–3558, 2013.
[28] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In CVPR, pages 1798–1807, 2015.
[29] B. Zhu, W. Li, Z. Wang, and X. Xue. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In ACM MM, pages 987–990, 2010.