Slicing Convolutional Neural Network for Crowd Video Understanding

Jing Shao[1] Chen Change Loy[2] Kai Kang[1] Xiaogang Wang[1]
[1]Department of Electronic Engineering, The Chinese University of Hong Kong
[2]Department of Information Engineering, The Chinese University of Hong Kong
jshao@ee.cuhk.edu.hk, ccloy@ie.cuhk.edu.hk, kkang@ee.cuhk.edu.hk, xgwang@ee.cuhk.edu.hk

Abstract

Learning and capturing both appearance and dynamic representations are pivotal for crowd video understanding. Convolutional Neural Networks (CNNs) have shown its remarkable potential in learning appearance representations from images. However, the learning of dynamic representation, and how it can be effectively combined with appearance features for video analysis, remains an open problem. In this study, we propose a novel spatio-temporal CNN, named Slicing CNN (S-CNN), based on the decomposition of 3D feature maps into 2D spatio- and 2D temporal-slices representations. The decomposition brings unique advantages: (1) the model is capable of capturing dynamics of different semantic units such as groups and objects, (2) it learns separated appearance and dynamic representations while keeping proper interactions between them, and (3) it exploits the selectiveness of spatial filters to discard irrelevant background clutter for crowd understanding. We demonstrate the effectiveness of the proposed S-CNN model on the WWW crowd video dataset for attribute recognition and observe significant performance improvements to the state-of-the-art methods (62.55% from 51.84% [21]).

1. Introduction

Understanding crowd behaviours and dynamic properties is a crucial task that has drawn remarkable attentions in video surveillance research [2, 4, 8, 10, 11, 13, 17–20, 28, 29, 32]. Despite the many efforts, capturing appearance and dynamic information from a crowd remains non-trivial. Ideally, as in most activity analysis studies, objects (*i.e.* groups or individuals) of interests should be segmented from the background, they should be further detected into different categories, and tracking should be performed to capture the movements of objects separately. One can then jointly consider the extracted dynamics for global understanding. Unfortunately, this typical pipeline is deemed too challenging for crowd videos.

Can deep learning offers us a tool to address the aforementioned challenges? Contemporary CNNs are capable of learning strong generic appearance representations from static image sets such as ImageNet. Nevertheless, they lack of the critical capability for learning dynamic representation. In existing approaches, a video is treated as a 3D volume and 2D CNN is simply extended to 3D CNN [5], mixing the appearance and dynamic feature representations in the learned 3D filters. Instead, appearance and dynamic features should be extracted separately, since they are encoded in different ways in videos and convey different information. Alternative solutions include sampling frames along the temporal direction and fusing their 2D CNN feature maps at different levels [7], or feeding motion maps obtained by existing tracking or optical flow methods [21, 27]. While computationally more feasible than 3D CNN, these methods lose critical dynamic information at the input layer.

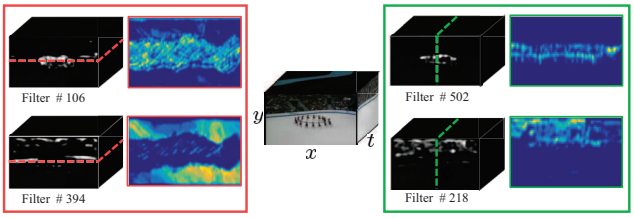

Figure 1. From a crowd video volume *ice ballet performance* in (b), several representative semantic feature cuboids and their temporal slices (xt and yt) are shown in (a) and (c). The temporal slices in the first row of (a) and (c) represent the dynamic patterns of *dancers*. Slices for the background visual patterns are visualized in the second row, where the pattern in (a) corresponds to *background scene* and that in (c) is *audience*.

In this study, we wish to show that with innovative model design, appearance and dynamic information can be effectively extracted at a deeper layer of CNN that conveys richer semantical notion (i.e. groups and individuals). In our new model design, appearance and dynamics have separate representations yet they interact seamlessly at semantic level. We name our model as *Slicing CNN* (S-CNN). It consists of three CNN branches each of which adopts different 2D spatio- or temporal-filters. Specifically, the first S-CNN branch applies 2D spatio-slice filters on video volume (xy-

plane) to extract 3D feature cuboids. The other two CNN branches take the 3D feature cuboids as input and apply 2D temporal-slice filters at xt-plane and yt-plane of the 3D feature cuboids, respectively. An illustration of the model is shown in Fig. 3.

This design brings a few unique advantages to the task of crowd understanding.

(1) *Object-aware* – A 3D feature cuboid generated by a 2D spatial filter records the movement of a particular semantic unit (e.g. groups or individual objects). An example is shown in Fig. 1, the feature map from a selected filter of a CNN hidden layer only shows high responses on the ice ballet dancers, while that from another filter shows high responses on the audience. Segregating such semantic classes in a complex scene is conventionally deemed challenging if not impossible for crowd video understanding.

(2) *Selectiveness* – The semantic selectiveness exhibited by the 2D spatial filters additionally guides us to discriminatively prune irrelevant filters such as those corresponding to the background clutter.

(3) *Temporal dynamics at semantic-level* – By applying temporal-slice filters to 3D feature cuboids generated by spatial filters at semantic-level, we can extract motion features of different semantic units, *e.g.* speed and acceleration in x- and y-directions.

We conduct empirical evaluations on the proposed deep structure and thoroughly examine and analyze the learned spatio- and temporal-representations. We apply the proposed model to the task of crowd attribute recognition on the WWW Crowd dataset [18] and achieve significant improvements against state-of-the-art methods that either apply a 3D-CNN [5] or Two-stream CNN [21].

2. Related Work

Compared to applying CNN to the static image analysis, there are relatively few works on the video analysis [3,5,7,18,21,26,27,30]. A 3D-CNN extends appearance feature learning in a 2D CNN to its 3D counterpart to simultaneously learn appearance and motion features on the input 3D video volume [3, 5].

It has been reported effective on the task of human action recognition. However, to capture long-term dependency, larger filter sizes and more layers need to be employed and the model complexity increases dramatically. To reduce model complexity, Karpathy *et al*. [7] studied different schemes of sampling frames and fused their features at multiple stages. These approaches did not separate appearance and dynamic representations. Nevertheless, traditional activity studies always segment objects of interests first and perform tracking on multiple targets that capture movements of different objects separately [9, 23, 24]. It shows that space and time are not equivalent components and thus should be learned in different ways. Ignoring

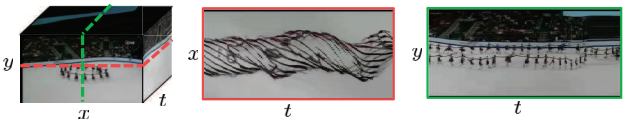

Figure 2. The slices over a raw video volume may inevitably mix the dynamics of different objects. For the raw video volume on the left, the xt-slice in the middle represents the dynamics of both the *dancers* and *background scene* (*i.e. ice rink*), while the yt-slice capture the dynamics of *audience*, *dancers*, as well as *ice rink*.

such prior knowledge and learning feature representation blindly would not be effective. Alternatively, two-branch CNN models [18, 21, 27] have been proposed to extract appearance and dynamic cues separately with independent 2D CNNs and combine them in the top layers. The input of the motion branch CNN is either 2D motion maps (such as optical flow fields [21] and dynamic group motion channels [18]). Different from 3D convolutions, a two-branch CNN is at the other extreme, where the extractions of appearance and dynamic representations have no interactions. These variants are of low cost in memory and calculation, but they inevitably sacrifice the descriptive ability for the inherent temporal patterns.

Albeit video-oriented CNNs have achieved impressive performances on video related tasks, alternative video representations other than spatial-oriented inputs are still under-explored. Besides representing a video volume as a stack of spatial xy-slices cut along the dimension t, previous works have shown that another two representations of xt-slices in dimension y and yt-slices in dimension x can boost feature learning of both appearance and dynamics on a variety of video-tasks [1,12,14–16,31]. However, they extract the motion feature slices directly from video volumes, but ignore the possibility that multiple objects or instances presented in one slice may occupy distinct motion patterns. Therefore, their dynamic feature representation may mix the motion patterns from different objects and thus fail to describe a particular type of motion patterns. An example is shown in Fig. 2. Moreover, the internal properties and connections among different slices were not well learned but just handled independently.

The proposed Slicing CNN model overcomes the limitations listed above. With innovative model design, appearance and dynamic informations can be effectively learned from semantic levels, separately and interactively. In addition, the proposed model is capable of extracting appearance and dynamic informations from long-range videos (*i.e.* 100 frames) without sampling or compression.

3. Slicing CNN Model

In this paper, we propose a new end-to-end model named as *Slicing CNN* (S-CNN) consisting of three branches. We first learn appearance features by a 2D CNN model on each

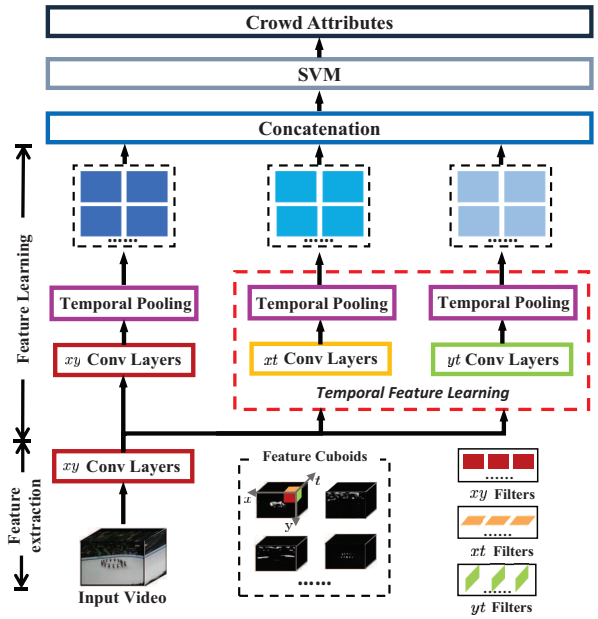

Figure 3. The architecture of the three-branch S-CNN model (*i.e.* S-CNN). The three branches share the same feature extraction procedure in the lower layers while adopt different 2D spatio- and temporal- filters (*i.e.* xy-, xt-, yt-) in feature learning. A classifier (*e.g.* SVM) is applied to the concatenated features obtained from the three branches for crowd attribute recognition.

frame of the input video volume, and obtain a collection of semantic feature cuboids. Each feature cuboid captures a distinct visual pattern, or an object instance/category. Based on the extracted feature cuboids, we introduce three different 2D spatio- and temporal-filters (*i.e.* xy-, xt-, and yt-) to learn the appearance and dynamic features from different dimensions, each of which is followed by a 1D temporal pooling layer. Recognition of crowd attribute is achieved by applying a classifier on the concatenated feature vector extracted from the feature maps of xy-, xt-, and yt-branch. The complete S-CNN model is shown in Fig. 3, and the detailed architecture of the single branch (*i.e.* S-CNN-xy, S-CNN-xt, and S-CNN-yt) is shown in Fig. 6. Their implementation details can be found in Section 4.

3.1. Semantic Selectiveness of Feature Maps

Recent studies have shown that the spatial filters in 2D CNNs on image-related tasks posses strong selectiveness on patterns corresponding to object categories and object identities [25]. Specifically, the feature map obtained by a spatial filter at one intermediate layer of a deep model records the spatial distribution of visual pattern of a specific object. From the example shown in Fig. 4, convolutional layers of the VGG model [22] pre-trained on ImageNet depict visual patterns in different scales and levels, in which the `conv4_3` layer extracts the semantic patterns in object level. For instance, the filter #26 in this layer precisely captures ice ballet dancers in all frames. Further examining the selectiveness of the feature maps, Fig. 5(a–c) demonstrates that different filters at `conv4_3` layer are possibly linked to different visual patterns. For example, filter #5 indicates the pedestrians on the crosswalk and filter #211 means extremely dense crowd; both of them extract patterns related to crowd. While filter #297 and #212 correspond to background contents like trees and windows of building.

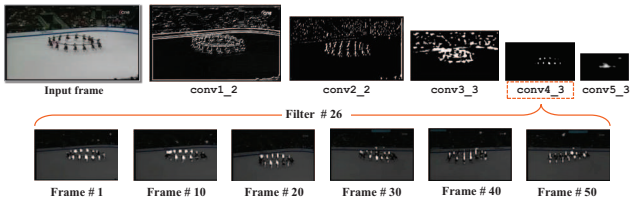

Figure 4. Feature responses of selective filters from different convolutional layers of the VGG model, in which `conv4_3` layer owns the best description power for semantic visual patterns in object level. This semantic feature maps precisely capture the *dancers* in ice ballet at all frames presented.

Motivated by the aforementioned observations, we could actually exploit such feature cuboids to separately monitor the movements of different object categories, both spatially and temporally, while reducing the interference caused by the background clutter and irrelevant objects.

3.2. Feature Map Pruning

The selectiveness of feature cuboids allows us to design models on a particular set of feature cuboids so as to capture crowd-related dynamic patterns and reject motions from irrelevant background contents. As shown in Fig. 5, some feature maps rarely respond to the subjects in crowd but mainly to background regions. How to efficiently learn dynamic feature representations from temporal slices obtained from these feature cuboids? Are all the feature cuboids meaningful to learn dynamic patterns? We answer these questions by pruning spatial filters that generate "irrelevant" feature maps and investigate its impact to the attribute recognition performance.

The "relevance" of a feature map is estimated by investigating their spatial distributions over a fixed validation set of images whose foreground crowds are annotated. The annotation is a binary mask estimated by a crowd segmentation method [6], denoted as $\mathbf{S}_i$ for a query image $i \in \mathcal{I}$, which is then resized to match the resolution of the extracted feature maps. We adopt two scores (*i.e.* affinity score and conspicuous score) to measure the "relevance".

Affinity score. The affinity score α_i^n measures the overlap ratio of the crowd foreground instances between the mask $\mathbf{S}_i$ and the n^{th} binarized feature map $\mathbf{F}_i^n \in \mathcal{F}_i$,

$$\alpha_i^n = \|\mathbf{1}_{[\mathbf{F}_i^n > 0]} \bullet \mathbf{S}_i\|_1 / \|\mathbf{S}_i\|_1, \tag{1}$$

where $\mathbf{1}_{[\cdot]}$ is an indicator function that returns 1 when its

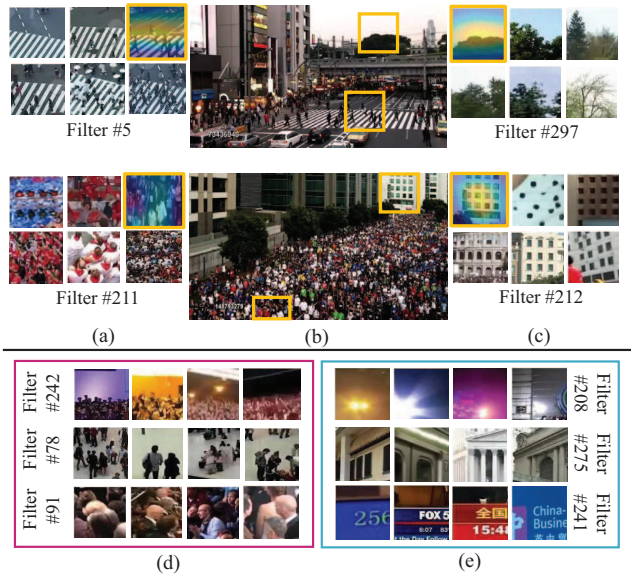

Figure 5. Semantic selectiveness of visual patterns by the spatial filters in `conv4_3` layer of the VGG model. The orange patches in (a) and (c) mark the receptive fields of the strongest responses with a certain filter on the given crowd images in (b). The top five receptive fields from images in WWW crowd dataset that have the strongest responses of the corresponding filters are listed aside. (d) and (e) present patches that have strongest responses for the reserved spatial filters and pruned spatial filters.

input argument is true. $\bullet$ denotes the element-wise multiplication.

Conspicuous score. The conspicuous score κ_i^n calculates the feature's energy inside the crowd foreground annotated in the mask $\mathbf{S}_i$ against its overall energy,

$$\kappa_i^n = \|\mathbf{F}_i^n \bullet \mathbf{S}_i\|_1 / \|\mathbf{F}_i^n\|_1. \tag{2}$$

We then construct a histogram $\mathbf{H}$ with respect to the filters in a certain layer. For filter $\#n$, if the feature map $\mathbf{F}_i^n$ satisfies either $\alpha_i^n > \tau_\alpha$ or $\kappa_i^n > \tau_\kappa$, given two thresholds τ_α and τ_κ, we have the value of its histogram bin as

$$\mathbf{H}(n) = \mathbf{H}(n) + \mathbf{1}_{[\alpha_i^n > \tau_\alpha \cup \kappa_i^n > \tau_\kappa]}, \forall i \in \mathcal{I}. \tag{3}$$

By sorting $\mathbf{H}(n)$ in a descending order, we retain the first r spatial filters but prune the left filters. The reserved filters are denoted as $\mathcal{N}_r$.

3.3. Semantic Temporal Slices

Existing studies typically learn dynamic features from raw video volumes [7] or hand-crafted motion maps [18, 21, 27]. However, much information is lost at the input layer since they compress the entire temporal range by subsampling frames or averaging spatial feature maps along the time dimension. Indeed, dynamic feature representations can also be described from 2D temporal slices that cut across 3D volume from another two orthogonal planes, as xt- or yt-slices shown in Fig. 2. They explicitly depict the temporal evolutions of objects, for example, the *dancers* in the xt-slice and *audience* in the yt-slice.

It is a general case that a xt- or yt-slice captured from a raw video volume contains motion patterns of multiple objects of different categories, which cannot be well separated since the features that identify these categories always refer to appearance but not motion. For instance, the yt-slice in Fig. 2 contains motion patterns from *audience*, *dancers* and ice rink. It is not a trivial task to divide their motion patterns apart without identifying these objects at first.

Motivated by this observation, we propose *Semantic Temporal Slice* (STS) extracted from semantic feature cuboids, which are obtained from the xy convolutional layers, as shown in Fig. 3. As discussed in the previous subsections, such kind of slices can distinguish and purify the dynamic representation for a certain semantic pattern without the interference from other objects, instances or visual patterns inside one temporal slice. Furthermore, given multiple STSs extracted from different horizontal and vertical probe lines and fed into S-CNN, their information can be combined to learn long-range dynamic features.

4. S-CNN Deep Architecture

In this section, we provide the architecture details of each branch (*i.e.* S-CNN-xy, S-CNN-xt, S-CNN-yt) and their combination (*i.e.* S-CNN).

4.1. Single Branch of S-CNN Model

Our S-CNN starts with designing a CNN for extracting convolutional feature cuboids from the input video volume. In principle, any kind of CNN architecture can be used for feature extraction. In our implementation, we choose the VGG architecture [22] because of its excellent performance in image-related tasks. As shown in Fig. 6, for an input raw video volume, we first follow the original setting of the lower layers of VGG-16 from `conv1_1` to `conv4_3`[1] to extract spatial semantic feature maps. The size of the feature cuboid $\mathcal{F}_i^s$ of time i is $c \times h_s \times w_s$, where c is the number of feature maps determined by the number of neurons, h_s and w_s denote the size of each feature map in the xy-plane. The number of feature cuboid is determined by the input video length τ.

S-CNN-xy branch. The S-CNN-xy branch learns spatio-temporal features from the xy-plane by xy-convolutional filters. Based on the spatial feature cuboids $\{\mathcal{F}_i^s\}_{i=1}^{\tau}$, we continue convolving feature maps with xy-filters from `conv5_1` to `conv5_3`, following VGG-16's structure to get the xy-temporal feature cuboids with a size of $\tau \times c \times$

[1]This structure is used for all experiments except S-CNN-RTS (raw temporal slices from video volume), whose lower layers are not for feature extraction, but also fine-tuned for feature learning.

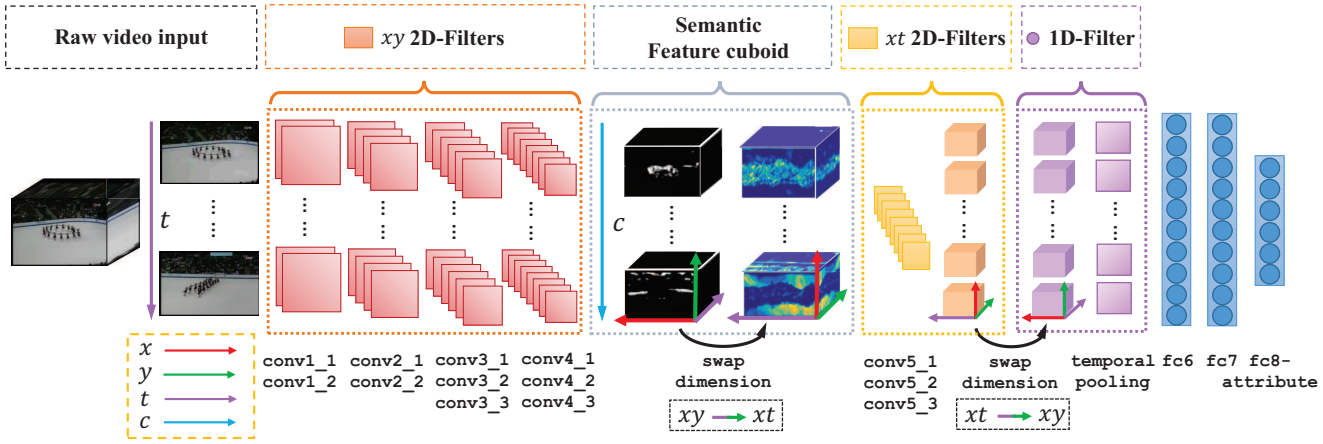

Figure 6. Single branch structure (*i.e.* S-CNN-xt). The whole structure is the same as the VGG-16 except the swap layers and the temporal pooling layer. Arrows in different colors denote different dimensions. (i) The first four bunches of convolutional layers in red are 2D convolutions on xy-slices, while the last bunch in orange are 2D convolutions on xt-slices. Following each bunch of the convolutional layers is a pooling layer. (ii) After the last convolutional layer (*i.e.* `conv5_3`), a temporal pooling layer in violet is adopted to fuse cues learned from different xt-slices by a 1×1 filter. (iii) The first two fully-connected layers both have 4096 neurons while the last one is 94 as the number of crowd attributes. All three branches (*i.e.* S-CNN-xt/-yt/-xy) use the same structure except with different types of filters.

$h_t \times w_t$. In other words, there are c xy spatio-temporal feature cuboids $\mathcal{F}^{xy}$, each of which is $\tau \times h_t \times w_t$. A 1×1 filter is then adopted on each $\mathcal{F}_i^{xy}$ to fuse the temporal information from different frames. The spatio-temporal feature maps $\mathcal{F}^{xy(t)}$ are fed into three fully-connected layers to classify the crowd-related attributes.

S-CNN-xt / -yt branch. For the purpose of learning features from $\{\mathcal{F}_i^s\}_{i=1}^{\tau}$ by xt- or yt- branch, we first swap dimensions of the original xy-plane to the corresponding xt- or yt-plane. Take xt-branch as an example, as shown in Fig. 6, the semantic feature cuboids turn to be $h_s \times c \times \tau \times w_s$ after swapping dimensions. We then substitute the xy-convolutional filters used in xy-branch with xt-filters for `conv5_1` to `conv5_3` layers. Before temporal pooling at the last stage, again we need to swap dimensions from xt-plane to xy-plane. The following structures are the same as those in xy-branch. The yt-branch is similar to xt-branch but with a different types of convolutional filters.

4.2. Combined S-CNN Model

After training each branch separately, we fuse the features learned from different spatial and temporal dimensions together by concatenating the spatio-temporal feature maps (*i.e.* $\mathcal{F}^{xy(t)}$, $\mathcal{F}^{xt(y)}$, and $\mathcal{F}^{(x)ty}$) from three branches with ℓ_1 normalization. Linear SVM is adopted as the classifier for the sake of its efficiency and effectiveness on high-dimensional feature representations. We train a SVM independently for each attribute, thus there are 94 models in total. To train each SVM, we consider videos containing the target attribute as the positive samples and leave all the rest as the negative samples. The complete S-CNN model is visualized in Fig. 3.

5. Experiments

5.1. Experimental Setting

Dataset. To demonstrate the effectiveness of the proposed S-CNN deep model, we investigate it on the task of crowd attribute recognition with the WWW Crowd Dataset [18], which is a comprehensive crowd dataset collecting videos from movies, surveillance and web. It covers 10,000 videos with 94 crowd attributes including places (Where), subjects (Who), and activities (Why). Following the original setup in [18], we train the models on 7220 videos and use a set of 936 videos as validation, while test the results over the rest 1844 videos. These sets have no overlap on scenes to guarantee the attributes are learned scene-independently.

Evaluation Metrics. We adopt both Area Under ROC Curve (AUC) and Average Precision (AP) as the evaluation metrics[2]. AUC is a popular metric for classification and its lower-bound is fixed to 0.5. It fails to carefully measure the performance if the ratio between the positive and negative samples is extremely unbalanced, which is just the case we confront. AP is effective to evaluate the multi-attribute detection performance, which is lower bounded by the ratio of positive samples over all the samples. Its lower bound can be written as $\text{mAP}_{\text{lb}} = \frac{1}{N_{\text{attr}}} \sum_{k=1}^{N_{\text{attr}}} |\mathcal{T}_k|/|\mathcal{T}|$, where N_{attr} is the number of attributes, $\mathcal{T}$ is the test set, $\mathcal{T}_k$ is the set of samples with the attribute indexed by k. In our experiments, the theoretical lower bound is 0.067.

Model Pre-training. As a common practice in most deep learning frameworks for visual tasks, we initialize the proposed S-CNN models with the parameters pre-trained on ImageNet. This is necessary since VGG requires diverse

[2] [18] only uses AUC for evaluation.

	mean AUC			mean AP		
τ	20	50	100	20	50	100
RTS-xy	91.06	92.28	-	49.56	52.05	-
STS-xy	91.76	92.39	**92.52**	54.97	55.31	**55.67**

Table 1. Results of S-CNN-xys learned from raw- and semantic-level with different temporal ranges (**bold**s are the best).

methods	τ	mean AUC			mean AP		
		xy	xt	yt	xy	xt	yt
STS	20	91.76	91.11	90.52	54.97	52.38	50.08
STS	100	**92.52**	**93.33**	**92.62**	**55.67**	**59.25**	**57.57**

Table 2. Results of S-CNNs learned from semantic-level with short- and long-range temporal slices. Results in **bold** are the best.

data to comprehensively tune its parameters. Although WWW crowd dataset has million of images, the diversity of scenes is low (*i.e.* around 8000). Specifically, we employ the VGG-16 model with 13 convolutional (`conv`) layers and 3 fully-connected (`fc`) layers. All `conv` layers in S-CNN models are initialized with the pre-trained model while three `fc` layers are randomly initialized by Gaussian distributions. We keep the first two `fc` layers with 4096 neurons followed by Rectified Linear Units (`ReLUs`) and `Dropout` while the last `fc` layer with 94 dimensions (attributes) followed by a cross-entropy loss function. If no specific clarifications are stated, we apply this strategy to initialize all experimental models.

5.2. Ablation Study of S-CNN

5.2.1 Level of Semantics and Temporal Range

The unique advantage of S-CNN is that it is capable of learning temporal patterns from semantic layer (higher layer of deep network). In addition, S-CNN can naturally accommodate larger number of input frames due to its effective network design, thus capable of capturing long-range dynamic features.

To understand the benefits of learning long-range dynamic features from semantic level, we compare the recognition performance of the proposed S-CNN models based on semantic temporal slices (STS) extracted from layer `conv4_3` and raw temporal slices (RTS) extracted directly from the video volume. The video length τ has three ranges: 20, 50, and 100 frames, denoted as S(/R)TS$_{[\tau]}$. Due to hardware limitation of current implementation, we cannot afford RTS$_{[100]}$ with full spatial information.

Low-level v.s. Semantic-level Temporal Slices. In comparison with the results by RTS$_{[\tau]}$, STS$_{[\tau]}$ ($\tau = 20, 50$) is superior especially in mAP scores, as shown in Table 1. The results of $xt/yt-$ semantic slices in Table 2 also reveal that the feature learning stage discovers motion patterns for semantic visual patterns, and they act well as the proxies to convey the motion patterns.

Short-range v.s. Long-range Feature Learning. As shown in Table 2, STS$_{[100]}$ performs the best and beats the other variants under both evaluation metrics. It demonstrats that the learned long-range features can actually increase the recognition power to find the crowd attributes that distinctively respond to long-range dynamics but are less likely to be identified by appearance alone, such as "performance" and "skate". See examples in Fig. 7(c).

$\lvert\mathcal{N}_r\rvert$	mean AUC				mean AP			
	xy	xt	yt	xyt	xy	xt	yt	xyt
100	91.02	91.70	91.16	92.31	46.83	50.32	48.18	53.14
256	92.61	92.49	92.22	93.49	54.68	54.13	53.26	60.13
256_rnd	91.40	92,32	90.21	92.69	51.87	53.12	46.38	57.03
512	92.52	93.33	92.62	94.04	55.67	59.25	57.57	62.55

Table 3. Results of STS$_{[100]}$ learned from different number of semantic neurons. Results by single-branch models (xy, xt and yt) and the complete model (xyt) are presented.

5.2.2 Pruning of Features

Feature pruning is discussed in Section 3.2. Here we show that by pruning features that are less relevant to the characteristics of crowd, it is promising to observe that the pruned irrelevant features cuboids do not make an significant drop on the performance of crowd attribute recognition. In particular, we prune 412 and 256 feature cuboids respectively out of the total set (*i.e.* 512) at the layer `conv4_3` with respect to the score defined in Section 3.2, and re-train the proposed deep models under the same setting as that of STS$_{[100]}$[3]. Their mAUC and mAP are reported in comparison with the results by the default STS$_{[100]}$ in Table 3.

Compared with the default model STS$_{[100]}$ with $|\mathcal{N}_r| = 512$, the models with $|\mathcal{N}_r| = 256$ (1) approach to the recognition results by STS$_{[100]}$, (2) outperform STS$_{[100]}$ on 13 attributes, 7 of which belong to "why" (*e.g.* "board", "kneel", and "disaster"), and (3) save about 3% on memory and 34% on time. With 100 feature cuboids remained, the proposed S-CNN can still perform well, and superior to the state-of-the-art methods (*i.e.* DLSF+DLMF [18] and 3D-CNN [5]), even with a single branch. For example, the xt-branch has 50.32% mAP which improves 9.1% and 11.2% from DLSF+DLMF and 3D-CNN respectively, and approaches to 51.84% by the Two-stream [21]. To further demonstrate the proposed pruning strategy, we randomly pruned half of the filters ($|\mathcal{N}_r| = 256$_rnd) for the comparison. As observed from the Table 3, the proposed pruning method performs much better than random pruning, suggesting the effectiveness of the proposed pruning strategy.

The results demonstrate: 1) the relevant spatial features are always companied with top ranks in $\mathbf{H}(n)$, proving the effectiveness of the proposed criteria. 2) spatial and dynamic representations can be represented by sparse yet effective feature cuboids. A small fraction of semantic feature cuboids are enough to fulfil crowd attribute recognition.

[3]Without other notations, STS$_{[100]}$ denotes the 100 frames-based S-CNN without feature cuboids pruning.

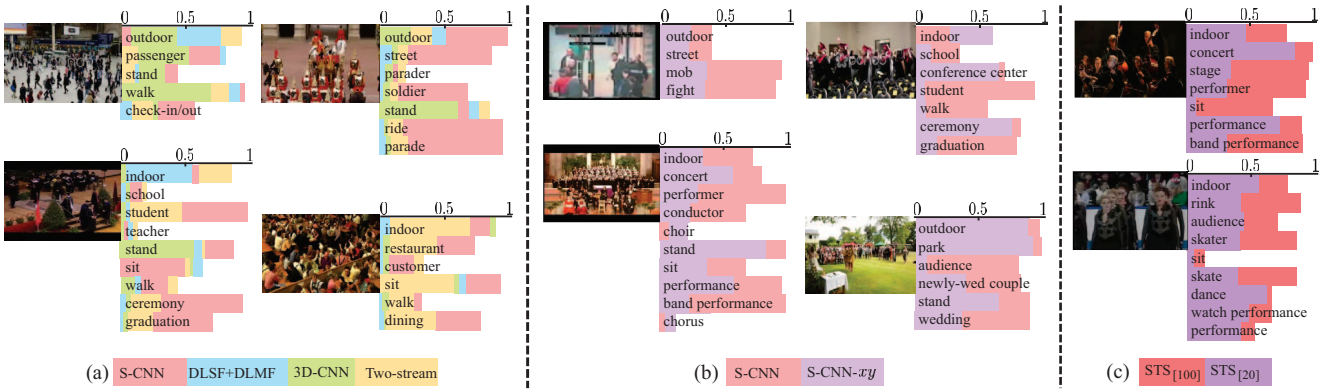

Figure 7. Qualitative recognition results on ground truth attributes annotated for the given examples. (a) Comparison between S-CNN and state-of-the-art methods. (b) Results by feeding temporal branch (S-CNN) and without feeding (S-CNN-xy). (c) S-CNN-STS learns on 100 and 20 frames. Different colors represent different methods. Bars are plot by the predict probabilities. Best viewed in color.

5.2.3 Single Branch Model v.s. Combined Model

The combination of appearance and dynamic features indeed composes representative descriptions that identify crowd dynamics. Not surprisingly, the combined model integrating xy-, xt- and yt-branches outperforms all single-branch models under both evaluation metrics. Under the setting of semantic temporal slices with a temporal range of 100 frames and keeping all feature cuboids, the combined model S-CNN reports remarkable mAUC score $\mathbf{94.04}\%$ and mAP score $\mathbf{62.55}\%$, which improve the optimal results of single-branch models by $\mathbf{3.3}\%$ (reported by xt-branch) in mAP. The improvement over mAUC is only 0.71%, but it might attribute to the deficiency of evaluation power. As shown in Table 3, the S-CNN with $|\mathcal{N}_r| = 100$ and $|\mathcal{N}_r| = 256$ are also superior to the optimal single branch with improvements of 2.82% and 5.45% respectively.

Qualitative comparisons between the spatial branch S-CNN-xy and the combined model S-CNN are in Fig. 7(b), which further demonstrate the significance of the temporal branches as they help to improve the performance for most attributes. In particular, for attributes of motion like "mob" and "fight", "sit", "stand", "walk" and etc, S-CNN presents a remarkable discriminative power for identification.

5.3. Comparison with State-of-the-Art Methods

We evaluate the combined Slicing CNN model (S-CNN) with recent state-of-the-art spatio-temporal deep feature learning models:

1) *DLSF+DLMF* [18]. The DLSF+DLMF model is originally proposed for crowd attribute recognition. It is a two-branch model with a late fusion scheme. We employ their published model with the default setting.

2) *Two-stream* [21]. The Two-stream contains two branches as a spatial net and a temporal net. We follow the setting by inputting 10-frame stacking optical flow maps for temporal net as adopted by both [21] and [18]. Besides, the parameters for temporal nets are also initialized with the VGG-16 model, as that in [27] for action recognition.

3) 3D-*CNN* [5]. A 3D-CNN model requires very large memory to capture long-range dynamics. As [5] applied 3D kernels on hand-crafted feature maps, for fair comparison, we mimic it by extracting features in lower layers of $\text{STS}_{[100]}$, and substitute $3 \times 3 \times 3$ 3D kernels for all 2D kernels after `conv4_3` layer and cut off half kernel numbers[4].

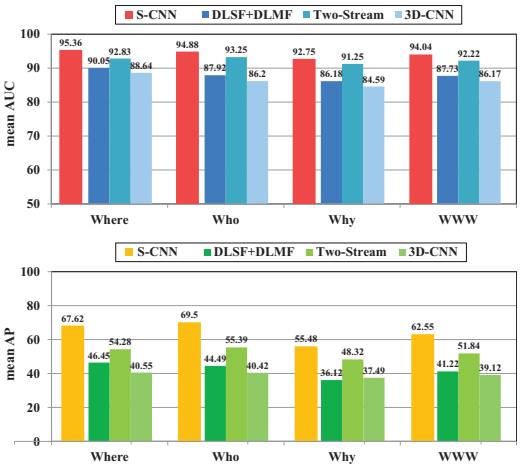

Figure 8. Performance comparisons with the referenced methods. The upper one is evaluated by mean AUC and the lower one is by mean AP. The histograms are formed based on the mean scores for attributes of "Where", "Who" and "Why", respectively. "WWW" represents the evaluations on all attributes.

5.3.1 Quantitative Evaluation

As shown in Fig. 8, histograms with respect to mAUC and mAP scores are generated to measure the performance on each type of crowd attributes, *e.g.* "Where", "Who" and "Why", as well as on the complete set "WWW". Clearly the proposed model outperforms the state-of-the-art meth-

[4]It needs 90G to handle 100 frames by the original number of kernels.

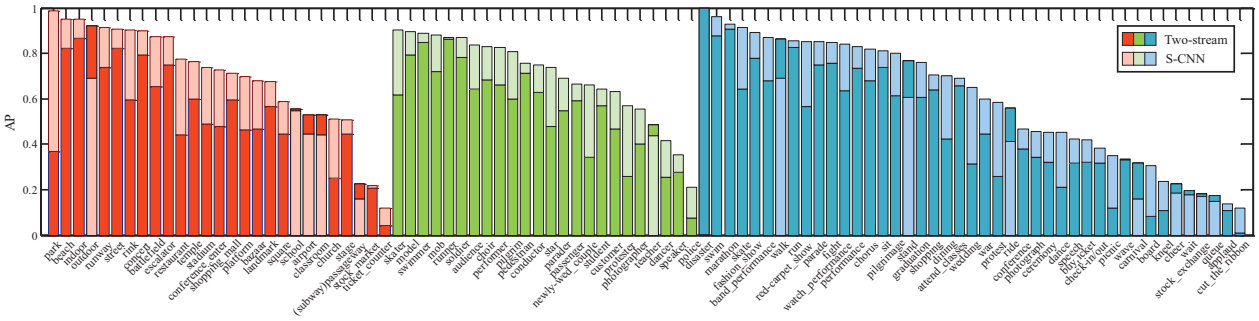

Figure 9. Average precision scores for all attributes by S-CNN and Two-stream. The set of bars marked in red, green, and blue refer to "where", "who", and "why" respectively. The bars are sorted according to the larger APs between these two methods.

ods under both metrics, and shows a large margin (particularly on mAP) over the second best approach in each sub-category. Among the reference methods, the Two-stream presents the best performance in all sub-categories. DLSF+DLMF wins 3D-CNN by the mAUC score on all three attribute types but loses at "Why" by mAP score. The reference methods tend to perform worst on motion-related attributes like "why", because they can neither capture long-term dynamics as Two-stream or 3D-CNN, nor extract dynamic features from specific and hand-craft motion feature maps as DLSF+DLMF. Since the proposed method is able to capture the dynamic feature representations from long-range crowd video and semantically push the features to be crowd-related, its result is thus superior over all the rest methods. Notice that S-CNN also incorporates the appearance features, which increases the performance of attributes at "Where" and "Who" even further. Even with a pruning of 412 feature cuboids from S-CNN model, it can still reach 53.14% mAP which also outperforms 51.84% by Two-stream [21].

We are also interested in the performance of each attribute. Fig. 9 shows the overlapped histograms of average precisions for all attributes by Two-stream and S-CNN. The bars are grouped by their sub-categories and sorted in descending order according to the larger AP between these methods at one attribute. It is easy to find that the envelope superimposing this histogram is always supported by the bars of S-CNN with prominent performance gain against Two-stream, while just in 15 attributes the latter wins. Among the failure attributes, most of them contain ambiguities with each other and have low APs for both methods. It means the recognition power is defective to these attributes by the existing deep learning methods. For example, "cheer" and "wave" may be confused with each other, "queue" and "stand" may happen in similar scenes.

5.3.2 Qualitative Evaluation

We also conduct quantitative evaluations for a list of exemplar crowd videos as shown in Fig. 7(a). The bars are shown as prediction probabilities. Although the probabilities of one attribute do not directly imply its actual recognition results, they uncover the discriminative power of different method as lower probability corresponds to ambiguity or difficulty in correctly predicting one attribute. The proposed S-CNN reliably predicts these attributes with complex or long-range dynamic features, like "graduation" and "ceremony", "parade" and "ride", "check-in/out"and etc. Moreover, some attributes that cannot be well defined by motion can also be revealed by S-CNN, for example "restaurant", "soldier" and "student". The appearance branch of S-CNN indeed captures the inherent appearance patterns belonging to these attributes. Some ambiguous cases do occur, e.g., "outdoor" in top-left and "sit" in bottom-left examples. The top-left instance takes place in a scene of airport/railway station – it is unclear whether the scene is an outdoor or indoor area.The bottom-left instance is a graduation ceremony, in which both "walk" and "sit" co-exist.

6. Conclusion

In this paper, we present a novel Slicing CNN (S-CNN) for crowd video understanding, with only 2D filters. We show that the spatial (xy-) filters capture appearance information, while temporal-slice (xt- and yt-) filters capture dynamic cues like speed and acceleration in x- and y-directions respectively. Their combination shows strong capacity in capturing spatio-temporal patterns, as evidence its results present superior performance in crowd attribute recognition on a large-scale crowd video dataset, against state-of-the-art deep models. We further show that spatial feature cuboids pruning could reduce redundancy leading to a sparser network. It is interesting to explore more strategies on feature cuboids selection in the future work.

Acknowledgment

This work is partially supported by SenseTime Group Limited, the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project Nos. CUHK14206114, CUHK14205615, CUHK419412, CUHK14203015), the Hong Kong Innovation and Technology Support Programme (No. ITS/221/13FP) and National Natural Science Foundation of China (NSFCNO. 61371192).

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *JOSA A*, 2(2):284–299, 1985. 2

[2] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *TPAMI*, 32(2):288–303, 2010. 1

[3] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential Deep Learning for Human Action Recognition. In *International Workshop on Human Behavior Understanding*, 2011. 2

[4] T. Hassner, Y. Itcher, and O. Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In *CVPR*, 2012. 1

[5] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2013. 1, 2, 6, 7

[6] K. Kang and X. Wang. Fully convolutional neural networks for crowd segmentation. *arXiv*, 2014. 3

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li. Large-scale Video Classification with Convolutional Neural Networks. In *CVPR*, 2014. 1, 2, 4

[8] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *CVPR*, 2009. 1

[9] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005. 2

[10] C. C. Loy, K. Chen, S. Gong, and T. Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*, pages 347–382. 2013. 1

[11] C. C. Loy, S. Gong, and T. Xiang. From semi-supervised to transfer counting of crowds. In *ICCV*, 2013. 1

[12] Z. Ma and A. B. Chan. Crossing the line: Crowd counting by integer programming with local features. In *CVPR*, 2013. 2

[13] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, 2009. 1

[14] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang. Motion analysis and segmentation through spatio-temporal slices processing. *TIP*, 12(3):341–355, 2003. 2

[15] S. Niyogi, E. H. Adelson, et al. Analyzing gait with spatiotemporal surfaces. In *Motion of Non-Rigid and Articulated Objects, Proceedings of the 1994 IEEE Workshop on*, pages 64–69, 1994. 2

[16] Y. Ricquebourg and P. Bouthemy. Real-time tracking of moving persons by exploiting spatio-temporal image slices. *TPAMI*, 22(8):797–808, 2000. 2

[17] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert. Data-driven crowd analysis in videos. In *ICCV*, 2011. 1

[18] J. Shao, K. Kang, C. C. Loy, and X. Wang. Deeply Learned Attributes for Crowded Scene Understanding. In *CVPR*, 2015. 1, 2, 4, 5, 6, 7

[19] J. Shao, C. C. Loy, and X. Wang. Scene-independent group profiling in crowd. In *CVPR*, 2014. 1

[20] J. Shao, C. C. Loy, and X. Wang. Learning scene-independent group descriptors for crowd understanding. *preprint, TCSVT*, 2016. 1

[21] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2, 4, 6, 7, 8

[22] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*, 2014. 3, 4

[23] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 2

[24] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2

[25] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. 3

[26] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015. 2

[27] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv*, 2015. 1, 2, 4, 7

[28] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *TPAMI*, 31(3):539–555, 2009. 1

[29] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *CVPR*, 2010. 1

[30] Y. Xiong, K. Zhu, D. Lin, and X. Tang. Recognize complex events from static images by fusing deep channels. In *CVPR*, 2015. 2

[31] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *TPAMI*, 29(6):915–928, 2007. 2

[32] B. Zhou, X. Tang, H. Zhang, and X. Wang. Measuring crowd collectiveness. *TPAMI*, 36(8):1586–1599, 2014. 1