Scene-Independent Crowd Understanding and Crowd Behavior Analysis

SHAO, Jing

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Electronic Engineering

The Chinese University of Hong Kong
June 2016
To
my husband Sheng Lu
&
our beloved parents

None of this would be possible without your love and support
Acknowledgements

First and foremost, I wish to thank my supervisor Professor Xiaogang Wang for his ceaseless encouragement, support, and generous mentorship. He is very experienced and has high talents in discovering new ideas. He guided me to think and research in a creative way and provided numerous perfect thinkings to renovate my original view of research. My deep gratitude also goes to Professor Chen Change Loy who offered many important suggestions and generated significant influence on each of my publications. He is my role model in many aspects. His attitude towards perfect always promotes me to move on and work harder. They made me fall in love with computer vision in the past four years. I am lucky for having the chance to work with them and thankful for their great supervision.

I would like to thank other thesis committee members, Professor Thierry Blu, Professor Hanqiu Sun, and Professor Huchun Lu from Dalian University of Technology. My warm appreciation also goes to Professor King Ngi Ngan and Professor Qing-Hu Meng for being examiners in my PhD screen test and proposal defense.

I am also very grateful to the students and staffs in Image and Video Processing Lab (IVPLab) and Multimedia Lab (MMLab). Especially, I benefit a lot from my interactions with Professor Xiaoou Tang, Professor Jianzhuang Liu, Professor Yu Qiao, Professor Dahua Lin, Professor Wanli Ouyang, Professor Hongsheng Li, Dr. Mo Chen, Dr. Rui Zhao, Wei Li, Shuai Yi, Xingyu Zeng, Zhe Wang, Cong Zhang, Xiao Chu, Kai Kang, Tong Xiao, Wei Yang, Zhuoyi Zhao, Shuang Li, Yantao Shen, Hongyang Li, Shuai Li, Dr. Ping Luo, Dr. Shi Qiu, Dr. Limin Wang, Dr. Xixuan Wu, Bolei Zhou, Chen Qian, Chaochao Lu, Yuanjun Xiong, Zhenyao Zhu, Linjie Yang, Chao Dong, Bing Xu, Shuo Yang, Zhirong Wu, Zhanpeng Zhang, Ruohui Wang, Ziwei Liu, Yonglong Tian and Shizhan Zhu. They all have strong capabilities in research and offered many help in my work and daily life. I feel very privileged to have several close friends in the lab. Especially, Xiao Chu and Shuang Li kindly accompanied me in many occasions and
cared about me personally. Kai Kang worked closely with me and put in a lot of time and efforts on my work to make all my achievement possible. I learned a significant amount of novel insights and research skills from him. Tong Xiao and Yuanjun Xiong helped me in many situations and taught me important details in practical experiences. I also want to thank Xiaojing Wu for many reasons. She is Shuai Yi’s wife, and they are both great researchers and always nice to me in many aspects.

I am indebted to my family in particular my mother Yuemei Zhang and my loving husband Lu Sheng for their enduring love, support and understanding. I also would like to thank my aunts to help me take care of my parents when I was far away from hometown. Without them, I would never have completed my PhD study. Finally, I would like to dedicate this work to my father Xiwu Shao, though he could not witness the completion of this thesis. His military spirit and strong mind always encouraged me to get over any difficulties throughout my PhD program.
致謝

首先，請允許我借此一隅向我的導師王曉剛教授致以深深的謝意。感謝他一來對我的鼓勵、支持和淳淳教導。王教授學識淵博，洞察敏銳，給我很多發性建議並指導我從不同的角度思考問題，讓我受益良多。此外，我也要特別感謝Chen Change Loy教授對我的每篇論文每個項目都傾注了很多心血並提供衆多很有價值的建議。Loy教授在各方面都一直是我的偶像，他認真的態度促使我更努力地工作以達到完美。在讀博這四年裡，兩位教授的悉心教導和指引讓我逐漸對機器視覺著迷。我很幸運並感恩能有機會獲得這些優秀教授的指導。

同時，我要感謝我博士論文答辯評審團的教授們：Thierry Blu教授，孫漢秋教授，以及來自大連理工大學的盧協川教授。另外，我也要感謝King Ngi Ngan教授和孟慶虎教授作為我博士論文開題的評審委員。

我還要感謝圖像視頻處理實驗室和多媒體實驗室的同學和老師們。他們是湯曉鵬教授，劉健莊教授，趙宇教授，林達華教授，歐陽萬里教授，李鴻升教授，陳默博士，趙瑞博士，李蔚，伊帥，曾星宇，王哲，張聰，初曉，康樺，肖桐，楊威，趙卓毅，李爽，沈岩濤，李弘揚，李帥，羅平，秋實，夏利民，吳希宣，周博磊，錢晨，陸超，熊元駿，朱臻，楊林傑，蓋超，徐冰，楊碩，武智融，張展鵬，王若輝，劉子瑋，田永龍和朱施展。他們都有很高的學術造詣並且在我的學術道路和日常生活中都給予了很多幫助。我很幸運在讀博士期間遇到了很多知心朋友。初曉和李爽經常陪我，照顧我。康樺跟我一起合作了很長時間，我的成就離不開他的幫助，我從他身上汲取了很多新穎的想法和研究技術。肖桐和熊元駿在教會我很多實驗技巧和操作經驗。此外，我還要感謝吳曉晶，作為伊帥的妻子，他們都是非常優秀的科研人員，在各方面都關心和照顧我良多。

最後，讓我把最真摯的謝意獻給我的家人，尤其是我的母親張曰梅和丈夫盛律，感謝他們一貫以來的理解和支持。而我的三姨和六姨，在我遠離家鄉求學在外的日子裡，一直替我照顧我年邁身體欠佳的父母，他們是我堅實的後盾。這篇博士畢業論文，我要把它獻給我的父親，邵希武，雖然他已經不能親眼看到，但是他作為一名軍人的堅毅精神一直激勵著我克服困難，勇往直前。
Abstract

Abstract of thesis entitled:

**Scene-Independent Crowd Understanding and Crowd Behavior Analysis**

Submitted by SHAO, Jing

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in August 2016

With the steady growth of population as well as the diversity of human activities, crowded scene understanding has been a widely concerned topic in computer vision. It has been of great interest in a large number of applications, such as developing crowd management strategies to avoid crowd-related disasters and analyzing crowd behaviors to help police to catch suspects. From the technical perspective, during the last decade, the field of crowd analysis in computer vision had a remarkable evolution. Much of this progress was sparked by the new and robust features and models for profiling crowd intrinsic properties as well as the creation of crowd datasets. The crowded scenes vary in densities, perspectives, and scales. It brings enormous challenges in capturing common properties from different crowded scenes. Furthermore, most of the existing studies on crowd understanding are scene-specific, that is, the crowd model is learned from a specific scene and thus poor in generalization to describe other scenes. This thesis proposes solutions for exploring the common properties across different crowded scenes by discovering scene-independent crowd attributes from group-level and macroscopic-level respectively.

The thesis first demonstrates that fundamental group-level properties, extensively studied in socio-psychological and biological research, can also be systematically quantified by visual descriptors. Groups are the primary entities that make up a crowd. The tendency of forming a coherent group with other pedestrians becomes more prominent in dense crowds, where pedestrians have to align with others to form collective
behaviors instead of moving freely. Precise group detection in crowd is challenging due to complex interaction among pedestrians. The proposed approach introduces a novel Collective Transition \((CT)\) prior to capture the underlying dynamics of a group and based on which formulates a robust group detector. Besides group detection, the proposed approach further characterizes and quantifies group properties from vision point of view. Crucially, these property descriptors are scene-independent, and can be effectively applied to public-scene with a variety of crowd densities and distributions.

Second, the property profiling is extended from group-level to macro-level. In the recent years, “attributes” are demonstrated to be particularly effective on characterizing objects, faces, and actions, as an alternative or complement to the categorical representations. Indeed, attributes can express more information in a crowd video as they can describe a video by answering “Who is in the crowd?”, “Where is the crowd?”, and “Why is crowd here?”, but not merely define a categorical scene label or event label to it. Furthermore, there is a large number of possible interactions among these attributes. Some attributes are likely to co-occur with each other, whilst some seem exclusive. From the modeling perspective, the attribute learning is conducted by a multi-task learning deep model to jointly learn appearance and motion features and effectively combine them. Instead of directly inputting multiple frames to a deep model to learn motion features as most existing works did for video analysis, the proposed approach specifically designs the crowd motion channels as the input of the deep model. This attribute-centric crowd analysis allows a better job in the traditional crowded scene understanding and provides potential abilities in cross-scene event detection, crowd video retrieval, and crowd video classification.

Third, the thesis studies the spatio-temporal convolutional neural networks. CNNs have shown a remarkable potential in learning appearance representations from images. However, the learning of dynamic representations from videos remains an open problem. To address this challenge, this thesis designed two novel spatio-temporal convolutional neural network architectures. The first approach introduces an intuitive but effective temporal-aware crowd motion channels, named as Slicing Volumetric \((S-Vol)\), by uniformly slicing the video volume from different dimensions. Another more general approach in then proposed by decomposing 3D feature maps (instead of video volume) into 2D spatio- and 2D temporal-slice representations, named as Slicing CNN.
(S-CNN). 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 meanwhile keep proper interactions between them, and (3) it exploits the selectiveness of spatial filters to discard irrelevant background clutters for crowd understanding.

The effectiveness of the proposed approaches is validated using videos captured from public crowded scenes. This thesis successively constructs two datasets, the CUHK Crowd Dataset and the WWW (Who do What at some Where) Crowd Attribute Dataset, with 474 and 10,000 crowd videos respectively. They cover the common crowded places, subjects, actions, and events.
摘要

隨著人口數量的持續增長以及人類活動種類和範圍的擴大，人群聚集場景分析在計算機視覺領域中已經備受關注，從而衍生出一批基於人群場景的應用分析，例如人群流向控制策略可以幫助避免由於人流擁擠踩踏而發生的重大事故，以及人群行為分析可以協助警方抓捕犯罪嫌疑人。在技術層面上，近年來，人群分析的研究取得了很大的進展，這要歸功於發現了重組新穎的特徵和模型以及建立了大規模的人群數據庫。然而，不同的人群場景擁有不同的人群密度、拍攝角度和人群數量。這給從不同場景中提取共同性質特徵帶來了很大挑戰。另外，現有的大部份工作都基於某個特定場景進行分析，即從某個場景中訓練出來的模型延展性比較差，不能用來分析其他場景。本論文對此提出解決方案，分別從小規模群組和大規模人群進行分析，提出可以跨場景分析的普遍性人群屬性。

首先，本論文研究發現一些在社會心理學和生物技術科學領域中常見的群組性質，實際上也可以從計算機視覺角度進行發掘和定量分析。小規模群組是組成大規模人群群體的主要組成部份。群組是指根據個體所處的空間位置及運動的差異，將群體劃分成更小的群組。當群體場景中人群密度越高時，個體行人越有可能一致性地組成群組。這是因為限於周圍的擁擠情況，行人們不能自由移動而是身不由己地隨人潮而動，進而產生群組集體性的行為模式。另外，人群場景中不同個體行人之間複雜的交互使得精確的群組檢測變得很困難。本論文引入了一種新型的先驗概念，名為集體遷移(CT)先驗。它能夠幫助提取群組的動態特性，進而檢測群組。另外，它還可以用來描述和定量計算群組特性。更重要的是，這些群組特性的描述子都是與場景無關的，也就是能夠有效地應用在不同密度和分佈的人群場景。

其次，本論文進一步把上述基於群組的性質擴展到大規模群體中。近些年研究證明“屬性”可以替代或者補充“類別”這種常用於描述物體、人臉和行為的表示形式。同樣的，在人群視頻中，相對於只給一個人群場景或者群組行為的類別標籤，屬性確實能夠表達更多的信息，例如“構成人群的成員是什？”，“群體發生場景是什？”，和“人群在做什？”。而且，這些屬性之間也有一些潛在關係，例如一些屬性會同時出現，而另外一些屬性之間有互斥作用。基於這些觀察和理論，本論文提出了一種多
任務深度學習模型，同時學習了外觀特徵和運動特徵。不同於以往視頻處理方法中把多幀作為深度學習模型的輸入，本論文設計了以人群運動映射圖作為模型輸入的方法。這種以屬性為核心進行人群分析的方法可以有效幫助理解和處理跨場景的人群事件檢測、人群視頻檢索，以及人群視頻分類等。

最後，本論文分析學習了時間-空間深度卷積神經網絡結構(CNN)。近些年，卷積神經網絡已經在圖像外觀識別上表現出非凡的潛能。然而，在視頻中提取學習動態信息卻一直沒有得到很好的解決。本論文基於此，設計了兩種新穎的時間-空間深度卷積神經網絡結構。第一個方法引入了一種直觀且有效的時域動態圖作為網絡輸入，命名為Slicing Volumetric (S-Vol)。這種輸入是把3維視頻從不同方向均勻切割出2維的切片。為了結構的普遍性，本論文又提出了第二個結構，命名為Slicing CNN (S-CNN)，不是切割原視頻，而是直接分解3維時空特征圖成2維的空間表示和時間表示。這種結構的優點包括：（1）能夠提取不同語義的分元（例如群組和物體）的動態信息，（2）既達到了分開學習外觀特徵和動態特徵的目的，同時又保留了外觀和動態信息的內在相關性，（3）輔助分析不同空間濾波器的可選擇性，進而去掉一些描述混亂背景的無關的濾波器以期望更好地對人群場景的分析和理解。

以上提出的方法和實驗都是在大量真實人群視頻場景中進行驗證的。本論文建立了兩個人群視頻數據庫：CUHK人群數據庫和WWW人群屬性數據庫，分別包含474個視頻和10000個視頻。這些視頻涵蓋了一般常見的人群場景、人群成員角色、人群行為和群體事件。
I hereby declare that this thesis is composed by myself and all the contents has not been submitted to this or any other universities for a degree. The materials of some chapters have been published in the following conferences or journals:

- **Chapter 3:**

- **Chapter 4 and 5:**

- **Chapter 6:**

- **Chapter 7:**
## Contents

Dedication i
Acknowledgments ii
Chinese Acknowledgments iv
Abstract v
Chinese Abstract viii
Declaration x
Contents xi
List of Figures xv
List of Tables xxiii

### 1 Introduction 1

1.1 Group-Centric Crowd Understanding 3
1.2 Attribute-Based Crowd Understanding 5
1.3 Deeply-Learned Crowd Features 7
1.4 Contributions 8
1.5 Thesis Road Map 11

### 2 Literature Review 12

2.1 Crowd Behavior Analysis 12
2.1.1 Micro-Level Representation 13
2.1.2 Macro-Level Representation 14
2.2 Group Analysis in Crowd 14
2.2.1 Group-Level Representation 15
2.2.2 Group Detection 15
2.2.3 Group Properties 16
2.3 Deep Convolutional Neural Networks (CNNs) in Video Analysis 16
5 Two-branch Deep Neural Network 55
  5.1 Deep Network Structure and Model Setting ............................. 55
  5.2 Motion Channels ......................................................... 56
  5.3 Experimental Results .................................................... 58
    5.3.1 Settings ............................................................ 58
    5.3.2 Evaluation on Deeply Learned Static Features (DLSF) ............ 58
    5.3.3 Evaluation on Deeply Learned Motion Features (DLMF) .......... 59
    5.3.4 Evaluation on Combined Deep Model .............................. 59
    5.3.5 Multi-task learning ............................................... 62
  5.4 Summary ................................................................. 63

6 Deeply Discovering Volumetric Slices 65
  6.1 Spatial Temporal Crowd Features ....................................... 65
    6.1.1 Crowd Motion Channels ............................................ 66
    6.1.2 Analysis of the Motion Channels .................................. 67
  6.2 Spatial Temporal Deep Architecture .................................. 70
    6.2.1 Network Configuration ............................................. 71
    6.2.2 Slice Fusion ....................................................... 71
  6.3 Experimental Results and Discussions ................................ 72
    6.3.1 Settings ............................................................ 72
    6.3.2 Evaluation on Different Input Channels .......................... 73
    6.3.3 Comparison with State-of-the-Art Methods ........................ 75
  6.4 Summary ................................................................. 79

7 Slicing Convolutional Neural Network 80
  7.1 Semantic Feature Maps Profiling ...................................... 81
    7.1.1 Semantic Selectiveness of Feature Maps .......................... 82
    7.1.2 Feature Map Pruning ............................................... 83
    7.1.3 Semantic Temporal Slices .......................................... 84
  7.2 S-CNN Deep Architecture ................................................ 86
    7.2.1 Single Branch of S-CNN Model .................................... 86
    7.2.2 Combined S-CNN Model ............................................. 88
  7.3 Experiments ............................................................. 88
    7.3.1 Settings ............................................................. 88
    7.3.2 Ablation Study of S-CNN .......................................... 89
    7.3.3 Comparison with State-of-the-Art Methods ........................ 92
  7.4 Summary ................................................................. 95

III Conclusions 96
8 Conclusions 97

Bibliography 99
List of Figures

1.1 Crowded scenes with (a) low-density and (b) high-density. .......... 2
1.2 Representative crowd video shots from (a) surveillance camera, (b) movies, and (c) individual collection. ......................... 3
1.3 Biological and human crowd systems share some same properties. (a) People in the shopping mall search for different commodities, similar to the bacteria colonies searching for food in groups. (b) Performers in band games act as a union, just like the bird flocks in migration. (c) Both animals and human behave similar to compete for resources or have incompatible goals. ................................................. 4
1.4 Crowd behavior can be better understood through inherent intra- and inter-group properties, such as collectiveness, stability, uniformity, and conflict. This thesis shows the possibility of quantifying such properties with scene-independent visual descriptors. Best viewed in color. .... 5
1.5 Representative examples from studies in attribute-based representations of faces [1; 2], actions [3; 4], and scenes [5; 6] (from left to right). .... 6
1.6 Category-based vs. Attribute-based representation. It is difficult to provide a category label to the crowd video in (a). Different viewpoints result in different categories. Alternatively, by defining multiple attributes, the video can be described comprehensively and expressly. Similar attribute-based representations can be found in (b). Red, green, and blue depict where, who, and why, respectively. .................. 7
2.1 Different micro-level crowd behavior analysis: (a) social force model [7], (b) multi-target tracking with social behavior model [8], and (c) hierarchical activity representation [9]. ................................. 13
2.2 Different macro-level crowd behavior analysis: (a) represents the motion pattern with a fixed-size spatio-temporal volume sets [10], (b) decomposes the entire scene into semantic regions for anomaly detection [11], and (c) finds co-occurring activities by computing the codewords of optical flow [12]. ................................. 14
3.1 An overview of the framework. Crowd groups are first detected with the CT prior. Based on the CT prior, a set of visual descriptors are devised to quantify four fundamental intra- and inter-group properties, namely collectiveness, stability, uniformity, and conflict, for each group. The proposed visual descriptors are applied for crowd group analysis, crowd dynamics monitoring, crowd video classification and crowd video retrieval. ................................................................. 21

3.2 (a) Coherent filtering [13] fails to distinguish two subtle groups. It can be addressed through discovering (b) a representative anchor tracklet (in red) and subsequently (c) a set of seeding tracklets to infer a group-specific CT prior. (d) With refinement based on the CT prior, two groups are separated. ................................................................. 23

3.3 Quantify the change of the internal topological structure of one group by the Levenshtein edit distance. A larger distance indicates a lower stability. 26

3.4 At time $t_1$, the ranked $K$-NN set of the current member (black disc) is $N^{t1} = \{2, 5, 1, 4, 3, 6\}$. There are two extreme possible conditions at time $t_2$, i.e. $N^{t2}(1) = \{7, 8, 10, 9, 11, 12\}$ or $N^{t2}(2) = \{6, 3, 4, 1, 5, 2\}$. Though the latter one is more stable than the previous one, according to Levenshtein string metric distance, they cannot be distinguished. .................. 27

3.5 (a) and (b) show a uniform and a non-uniform group, respectively. The original coherent group detection, the sub-groups obtained through further clustering, and the optimal number of cuts inferred by modularity function are shown from left to right. ................................. 28

3.6 (a) Two conditions of conflict location in groups. Hot color represents a high degree of conflict. (b) Conflict distribution of groups. The map is normalized with the groups’ center, moving direction, and the largest distance from groups’ contour to its center. ................................. 29

3.7 Comparative results of group detection with four methods. Groups are distinguished with colors. Red color indicates outliers. Arrows are moving directions. Best viewed in color. ................................. 30

3.8 Left: quantitative comparison of group detection methods. Right: relative improvement of the proposed approach (CT) compared with DTM, HC, and CF. ................................................................. 30

3.9 Parameter evaluation. (a) and (b) show group detection evaluation results with different $\eta$ and $\delta$ respectively. Red, green, and blue indicate the three criteria NMI, Purity, and RI. ................................. 32

3.10 Examples of group detection results with different parameter settings (respect to parameter $\delta = 3, 5, 10$) are shown in the first three rows. The last row shows the groundtruth. Groups are distinguished with colors, and the red crosses indicates outliers (best viewed on screen). ................................. 34
3.11 Running time evaluation. (a) shows the running time comparison between CT and CT w/o seeding tracklets, and (b) shows running time of CT with different tracklet numbers (blue) and group numbers (green). 34

3.12 Property correlations of collectiveness ($\Phi^{coll}$) with (a) stability ($\phi^{stab}$) and (b) conflict ($\Phi^{conf}$). These vector-valued descriptors are compressed by PCA as the projection onto their first principal subspace. A fractional polynomial is fitted to the data of stability. Examples shown in the bottom are corresponding to the yellow-marked dots in the top figure. The target groups are marked in pink. Best viewed in color. 35

3.13 Illustration of different group states. 36

3.14 (a) Confusion matrix of classifying group states by combining all the group descriptors (darker color represents higher accuracy). (b) Average accuracy of using each descriptor and combining them. (c) Performance drops by using only one descriptor on classifying each of the group states. A lower bar indicates that the descriptor is more effective on classifying a particular group state. (d) Legend for (b) and (c). 1 ~ 5 are single descriptors, and 6 is combination of the five descriptors. 37

3.15 (a) Representative frame with group detection results. Colors indicate different groups. (b) Distributions of different types of groups in a shopping mall and a escalator scene. Colors indicate group states automatically recognized with the proposed descriptors. Best viewed in color. 38

3.16 Examples from three semantic clusters. Red boxes mark the patches grouped to the corresponding semantic clusters. Below each sample are three patches captured from the beginning, middle, and end of time. 39

3.17 Monitoring crowd dynamics via group state transition as shown in the left. For each scene, two frames indicate representative states are shown in the right. (a) Pink-marked groups are monitored by recognizing their states over time. (b) Pedestrians inside red boxes are monitored. Their states change as they join different groups along the time line. 40

3.18 Confusion matrices of crowd video classification (darker color represents higher accuracy). Left: using holistic features in [10]. The average accuracy is 44%. Right: using the proposed descriptors. The average accuracy is 70%. 41

3.19 Crowd video classification results. Left: the representative frame of input video. Middle: classification results by holistic features [10]. Right: classification results by the proposed descriptors. The red bar represents correct prediction, and the green bar represents incorrect prediction. 42

3.20 Comparison quantitative results of crowd video retrieval. (a) Average precision over all video classes on top $k$ retrieval results. (b) Average precision of each video class on top 100 retrieval results. 43
3.21 Exemplar results of crowd video retrieval. The first column shows the query videos, and the remains of each row show the top 5 retrieval results.

4.1 Raw tag wordle (partial tag set). Bigger font size of a word suggests higher frequency of a tag appears in dataset.

4.2 A quick glance of WWW Crowd Dataset with its attributes. Red represents the location (Where), green represents the subject (Who), and blue refers to event/action (Why). The area of each word is proportional to the frequency of that attribute in the WWW dataset.

4.3 Several video examples in the WWW dataset. Both two videos in the first row have ambiguous attributes. While the other two videos in the second row have multiple attributes in where, who, and why.

4.4 Visualize users’ response time. The blue circles in (a) plot the response time of all annotators on labeling tasks, and the red line marks the average response time of each annotator. (b) shows the histograms of response time of different cues.

4.5 (a) The number of wrongly labeled samples with the background cue (indicated by blue bars) and how many of them can be corrected after adding the tracklet cue. (b) Accuracy comparison between the tracklet cue and tracklet + background. All the results are obtained from the user study described in Sec. 4.3.

5.1 Deep model. The appearance and motion channels are input in two separate branches with the same deep architecture. Both branches consist of multiple layers of convolution (blue), max pooling (green), normalization (orange), and one fully-connected (red). The two branches then fuse together to one fully-connected layers (red).

5.2 Motion channels. The first row gives an example to briefly illustrate three motion channels construction procedure. For each channel, two examples are shown in the second and third rows. Individuals in crowd moving randomly indicates low collectiveness, while the coherent motion of crowd reveals high collectiveness. Individuals have low stability if their topological structure changes a lot, whereas high stability if topological structure changes a little. Conflict occurs when individuals move towards different directions.

5.3 AUC of each attribute obtained with the two-branch model (i.e. DLSF+DLMF). Blue, red, and green indicate attributes of where, who and why respectively.
5.4 Good and bad attribute prediction examples are shown in (a) and (b). For each image, its top four attributes with the highest prediction scores with the proposed two-branch model (i.e., DLSF+DLMF) are shown. The heights represent prediction scores. Blue indicates correctly predicted attributes and orange indicates wrong prediction (false alarms). In (c) all the user annotated ground truth attributes are shown for each image example. If a prediction score on an attribute is lower than 0.5, it is represented by red, which means miss detection. Otherwise it is blue.

5.5 Six attributes predicted by DLSF, DLMF, and DLSF + DLMF. The first row shows negative examples with blue bar, and the second row shows positive examples with red bar. The prediction scores with DLSF, DLMF and DLSF + DLMF are represented by green, yellow, and orange.

5.6 Attributes with similar and disparate performance from user study and combined deep model. User study is the left score and the proposed combined deep model is the right score.

6.1 Illustration of temporal-aware crowd motion channels. The (a) xy-slice, (b) xt-slice and (c) yt-slice are extracted from a video volume (d) and their corresponding collectiveness maps (hot color indicates high collectiveness). A collectiveness map by average pooling is presented in (e). Much more detailed and complete temporal information is preserved inside the xt- and yt-slices in comparison with the xy-slices as the sampled video frames or feature maps. The average collectiveness map suppresses the rich contents revealed in xt- and yt-slices and just keeps the average collectiveness responses along the xy plane.

6.2 Examples of temporal-aware motion channels extracted from a crowd video volume (a). The (b) xy-slices, (c) xt-slices, and (d) yt-slices are marked by green, red, and blue, respectively. The horizontal direction in each image of (c) and (d) indicates the t-axis.

6.3 Temporal-aware motion channels from a crowd video volume (a) and its optical flow volume (b). These slices are extracted at the same positions from both volumes. Although they are constructed from different features, they share similar spatial and temporal patterns.

6.4 Temporal-aware motion channels extracted from the collectiveness volume. (a) Two crowd videos with different levels of collectiveness. (b)–(d) The xy-, xt- and yt-slices. The video shown in the upper row is about “crosswalk” and the one in the lower row contains crowd “marching”. The different motion patterns between two videos can be easily depicted by examining the temporal evolutions of their collectiveness from xt- and yt-slices. Hot color represents high collectiveness.
6.5 Deep model structure and fusion strategies. (a) S-Vol architecture. The convolutional layers, global sum-pooling layer, and fully-connected layers are represented by blue, green, and red, respectively. (b) Fusion strategies. Early fusion, late fusion and slow fusion are listed from left to right.

6.6 Comparison of the feature responses of optic-xt-slices and optic-yt-slices before the first fully connected layer and learned by late fusion with shared- and separate-weight strategies. (a) the input video snapshot and its corresponding optical flow displacement field. (b) ten selected optic-xt-slices (upper) and optic-yt-slices (lower). (c) feature responses learned by shared-weight strategy. (d) feature responses learned by separate-weight strategy. The feature responses shown in upper column come from optic-xt-slices, and those in lower column are from optic-yt-slices. The feature response of a single slice is of dimension 32 and the responses are normalized into [0, 1].

6.7 Attribute prediction examples by optical flow-based temporal-aware motion channels. For each video, the ground-truth attributes are depicted next to it. The length of bar from the left origin to its end indicates the prediction probability of an attribute either by xy-, xt- or yt-slices (marked by red, blue and purple) as well as the preceded method DLMF in Chapter 5] (marked by green). The rightmost two examples marked by a black box are the failure cases.

6.8 Attribute prediction examples by optical flow-based temporal-aware motion channels and by the appearance channels separately. Each example has the positive attribute results by optic-xyt shown on the left and by the appearance (RGB) shown on the right. The groundtruth attributes are listed under each examples. The heights represent prediction scores. Orange indicates correctly predicted attributes, and blue indicates wrong prediction (false alarm).

7.1 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.
7.2 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*.  

7.3 Example of temporal slices from 3D video volume. The top row shows the video volume with a horizontal ($x$) and vertical ($y$) line marked on it. 5 sampled frames are presented besides it. (c0) and (d0) are $xt$- and $yt$-slice with the entire timeslot. (c1)-(c5) and (d1)-(d3) are visual patterns observed in $xt$- and $yt$-slice. (c1) and (d2) are buses moving from left top towards right bottom. (c2) are two pedestrians moving along horizontal zebra crossing. (c3) are pedestrians moving along vertical zebra crossing in two timestamps. (c4) and (d3) are vehicles moving horizontally. (c5) is a green bus come across with another red truck from another direction. (d1) is a stationary bus waiting on the vertical line.  

7.4 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.  

7.5 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.  

7.6 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 $1 \times 1$ 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.
7.7 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. ............................................. 87

7.8 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. .......................................................... 92

7.9 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. ......................................................... 93

7.10 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. .................................................. 94
List of Tables

3.1 List of group descriptors. .............................................. 25
3.2 Quantitative comparison of CT and CT w/o seeding tracklets. .... 33
3.3 List of crowd video classes. ........................................... 41

4.1 Comparison of WWW and other existing datasets. WWW offers the largest number of videos, scenes, and frames. ..................... 47
4.2 User accuracy with four types of cues. ................................ 52

5.1 Compare deeply learned features with baselines. The last column shows the number of attributes (out of the total number of 94) on which the proposed deep features have higher AUC than baselines. ................. 58
5.2 Compare average AUC with single-task learning and multi-task learning. The last column is the number of attributes where multi-task learning outperforms single-task learning. ............................... 62

6.1 Crowd attribute recognition results (AUC score) on the WWW Crowd Dataset with different temporal-aware input channels and fusion strategies. The best results for each kind of input are in bold. The results for the stack-combined models are shown on the right. ................. 72
6.2 Compare with baselines. Best are in bold. ............................ 76

7.1 Results of S-CNN-\(x_{ys}\) learned from raw- and semantic-level with different temporal ranges (\textbf{bolds} are the best). ......................... 90
7.2 Results of S-CNNs learned from semantic-level with short- and long-range temporal slices. Results in \textbf{bold} are the best. ................. 90
7.3 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. ................................................................. 91
Chapter 1

Introduction

Crowd is a common phenomenon in our daily life. In the past decades, many researchers from different fields made lots of efforts on studying crowd. Civil Engineers [14] use the crowd information to design public spaces such as large concerts and shopping malls, to avoid crowd-related disasters. Computer graphic researchers [15] use crowd simulation to guide designing variety of animations.

In computer vision, recognizing crowd behaviors and activities in crowd videos have drawn a remarkable amount of attention due to its wide applications and also its scientific interest. Many works have done including crowd behavior analysis [16–18], crowd tracking [19–21], crowd segmentation [22–24] and crowd counting [25–27]. The first IEEE workshop on Modeling, Simulation and Visual Analysis of Large Crowds was held in 2011 \(^1\), to encourage the viewpoint and technique exchanges among researchers from different fields.

In general, compared with works focus on behavior analysis in sparse environments, understanding and recognizing behaviors/patterns in crowded scenes own several underlying challenges:

1. **Crowdedness.** Pedestrians in dense environments behave different from those in sparse environments by presenting new characteristics other than individual personalities. It is difficult to detect and track pedestrians due to frequent occlusions and scene clutters.

2. **Behavior Complexity.** There are various types of behavior patterns simultaneously happen in a crowded scene. Ideally, as in most activity analysis studies, \(^1\)http://vision.eecs.ucf.edu/ICCVWorkshop/home.html
subjects of interests should be segmented from the background, they should be fur-
ther 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. Besides, the behaviors vary a lot in different
crowded scenes with various densities. Studies [28] have proved that when the
crowd density reaches a certain value, people will automatically form some certain
lanes, without any external planning or organization. As shown in Figure 1.1(b),
as crowd density grows higher, especially in extremely crowded scenes, pedestrians
start to lose their self-organization ability but tend to be influenced more from the
interactions with other surrounding individuals and then form a limited number of
groups with common goals.

3. **Scene Diversity.** In the past decades, most researchers studied crowds mainly in
the surveillance systems for the security concern. However, most of these studies
are scene-specific, that is, the crowd model is learned from a specific scene and
thus poor in generalization to describe other scenes. As the rapid growth of videos
in the Internet, the crowd videos are not limited to surveillance cameras. There
are lots of crowded scenes in the movies, TV shows and individual collections, as
shown in Figure 1.2(b) and (c). Compared to surveillance videos, these videos
are much more diverse with different viewpoints, camera motions, and various

**Figure 1.1:** Crowded scenes with (a) low-density and (b) high-density.
1.1 Group-Centric Crowd Understanding

Group dynamics have been extensively studied in socio-psychological [29] and biological [30] research as the primary processes that influence crowd behaviors. In these studies, group dynamics are characterized by both intra- and inter-group properties. *Intra-group properties*, e.g. collectiveness, stability, and uniformity, denote internal coordination among members in the same group. Whilst *inter-group properties*, e.g. conflict, reflect the external interaction between members in different groups. Such properties widely exist in animal/insect crowd systems (e.g. bacterial colonies and bird flocks) as shown in Figure 1.3, and are also frequently researched in socio-psychological studies [31]. For instance, bacterial colonies were found to exhibit collective behavior to achieve a common goal, i.e. spreading of diseases [30]. From the sociological view-point, conflict occurs for competition of resources or goal incompatibility [29].
Figure 1.3: Biological and human crowd systems share some same properties. (a) People in the shopping mall search for different commodities, similar to the bacteria colonies searching for food in groups. (b) Performers in band games act as a union, just like the bird flocks in migration. (c) Both animals and human behave similar to compete for resources or have incompatible goals.

In the context of visual surveillance, groups primarily make up human crowds. There is no single agreed definition of what constitutes a group. Following a rich body of literature [28; 32–34], in this work, a group is defined as a cluster of members, e.g. friends or family members, who tend to move together for a sustained period of time, due to unconscious self-organization induced by common destination and social interaction. Interestingly, the tendency of forming a coherent group with other pedestrians becomes more prominent in dense crowds, where pedestrians have to align with others to form collective behaviors instead of moving freely [28].

When pedestrians form groups, they exhibit some interesting properties in their dynamics, which share commonalities with socio-psychological and biological studies, as shown in Fig. 1.4. For instance, a collective behavior is observed when pedestrians in a group maneuver towards a common destination. During a crowd disaster, turbulent dynamics in the crowd can be characterized by the stability property. A crowd tends to have non-uniform distribution when its members have different social relationships and walk in less restrictive area. Two pedestrian groups with different goals, e.g. when crossing roads from different directions, exhibit conflict behavior. Clearly, understanding such properties provides critical mid-representation to crowd motion analysis [7;
1.2. Attribute-Based Crowd Understanding

In the recent years, studies in attribute-based representations of objects [36–38], faces [1; 2], actions [3; 4; 39], and scenes [5; 6; 40; 41] (as shown in Figure 1.5) have drawn a large attention as an alternative or complement to categorical representations as they characterize the target subject by several attributes rather than discriminative assignment into a single specific category, which is too restrictive to describe the nature of the target subject. Typically, the existing works defined a vocabulary of attributes relevant to the corresponding domain knowledge, such as a set of textures or parts of animals [37], facial characteristics (e.g, types of hairs, eyes, lips, and levels of ages) [1; 2; 42], and clothing properties (e.g, various types of shoes, necktie, bags, and earrings) [38; 43]. The attribute-based representation has been demonstrated to have the capability to aid both classification and search/retrieval. First, it is useful to describe both familiar and unfamiliar subjects. For example, given an image of leopard, it is hard to name it by most people but can be simply described as the animal with “spotted” and “furry” [36; 44]. Second, it enables zero-shot learning, and also can be served as a mid-level features for high-level tasks like object classification [1; 37; 45].

However, there has been little investigation into this direction in crowd analysis.

Figure 1.4: Crowd behavior can be better understood through inherent intra- and inter-group properties, such as collectiveness, stability, uniformity, and conflict. This thesis shows the possibility of quantifying such properties with scene-independent visual descriptors. Best viewed in color.

12; 22; 35], and could facilitate other high-level semantic analysis such as crowd scene understanding, crowd video classification, and crowd event retrieval.
Even though scientific studies [46; 47] have shown that different crowd systems share similar principles that can be characterized by some common properties or attributes. A crowd video example is shown in Figure 1.6(a). The traditional way is to provide it with one category label. However, from the performers’ viewpoint, it can be labeled as an “orchestra performance”, while from the audience’s viewpoint, it can be regarded as “watching performance”. The similar confusion also exists in the exemplar videos shown in Figure 1.6(b). For example, if an orchestra performance happens upon the military, it can also be categorized as military marching. This problem can be fixed by adopting attribute-based representation. Indeed, attributes can express more information in a crowd video as they can describe a video by answering “Who is in the crowd?”, “Where is the crowd?”, and “Why is crowd here?”, but not merely define a categorical scene label or event label to it. For instance, an attribute-based representation might describe the crowd video shown in Figure 1.6(a) as the “conductor” “conduct” “orchestra performance” in a “concert” with “audience” “watching performance”. Other videos shown in Figure 1.6(b) can also be described in the same way.

Figure 1.5: Representative examples from studies in attribute-based representations of faces [1; 2], actions [3; 4], and scenes [5; 6] (from left to right).
1.3 Deeply-Learned Crowd Features

Recognizing crowd behaviors and activities in videos is a challenging task that has drawn a remarkable amount of attention in the field of crowd understanding. It is clear that “behavior” or “activity” is characterized by both its appearance and motion features, thus a unified feature learning approach that parses both the spatial and temporal contents from a crowd video is necessary to enable thorough analysis and understanding. Among the appearance and motion clues, the latter is of the utmost importance for crowd understanding. First, informative motion features are vital to complement the appearance features which are often ambiguous in videos with dense crowds and background clutters. Second, many crowd behaviors cannot be identified according to their appearance but have to be described based on their dynamics over time. Although flourishing studies have been conducted to explore the appearance representations of single-shot images [48–50], effective description of temporal information

Figure 1.6: Category-based vs. Attribute-based representation. It is difficult to provide a category label to the crowd video in (a). Different viewpoints result in different categories. Alternatively, by defining multiple attributes, the video can be described comprehensively and expressly. Similar attribute-based representations can be found in (b). Red, green, and blue depict where, who, and why, respectively.
in crowd videos is still an open problem.

Witnessing the recent advances of the deep convolutional neural networks (CNNs), it is interesting to explore whether CNNs can achieve the same success in video understanding by learning the underlying motion features from videos. In existing approaches, a video is treated as a 3D volume and 2D CNN is simply extended to 3D CNN [51], 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 [52], or feeding motion maps obtained by existing tracking or optical flow methods [53; 54]. While computationally more feasible than 3D CNN, these methods lose critical dynamic information at the input layer. Moreover, it is observed that the features learned by these learning schemes tend to bias on spatial clues than temporal ones [52]. Furthermore, none of these studies has attempted to deeply learn the features from the crowd videos.

1.4 Contributions

This thesis contributes to scene-independent crowd understanding and behavior analysis by tackling some of the challenges stated above. The contributions of the thesis are as follows:

1. A new set of scene-independent group properties are proposed for crowded scene understanding. The goal is to characterize and quantify these group properties by descriptors from vision point of view, and to study their potentials on crowd behavior analysis and crowd scene understanding. These descriptors convey richer group-level information in comparison to the conventional group size and velocity information [55]. Importantly, they are scene invariant and robust to public scenes with variety of crowdedness. The proposed descriptors are demonstrated to be effective in identifying the intrinsic group states (gases, fluids, and solid) following the common analogy employed in crowd modeling literature [56–58], and also show their superiority in crowd video classification over existing activity descriptors [10] and can be used to retrieve crowd videos across different scenes. Experiments are
conducted on hundreds of video clips collected from over 200 crowded scenes. The dataset (named as *CUHK Crowd Dataset*) and the ground truth are made publicly available to facilitate future research in group-level crowd analysis\textsuperscript{2} [59; 60].

2. Attributes are particularly effective on characterizing generic properties across scenes. The *CUHK Crowd Dataset* focuses on group properties, and both the scene diversities and the video numbers are small. To this end, a new large-scale crowd video dataset is designed to understand crowded scenes from attribute-level, named as the *Who do What at someWhere* (WWW) Crowd Dataset\textsuperscript{3}. It contains 10,000 videos from 8,257 crowded scenes. To date the WWW Crowd Dataset is the largest crowd dataset. The videos in the WWW crowd dataset are all from real-world, collected from various sources, and captured by diverse kinds of cameras. Importantly, there are 94 meaningful attributes defined on the dataset as high-level crowd scene representations. These attributes are navigated by tag information of the crowd videos from Internet. They cover the common crowded places, subjects, actions, and events. A user study is further designed to measure how accurately humans can recognize crowd attributes, and with which type of data that users can achieve the highest accuracy [61].

3. From the modeling perspective, deeply-learned crowd features outperform traditional hand-crafted features on crowd attribute recognition. Compared with the existing methods that directly input a single frame and multiple frames to the deep neural network, the proposed two-branch model adopts a novel motion feature channels as input. The motion channels are inspired by generic properties of crowd systems, which have been well studied in biology and physics. With multi-task learning, the correlations among attributes are well captured when learning deep features. Furthermore, it allows a better job in the traditional crowded scene understanding and provides potential abilities in cross-scene event detection, crowd video retrieval, and crowd video classification [61].

4. To tailor with the properties of CNNs and unleash the expressive power for temporal component of crowd videos, an intuitive but effective temporal-aware crowd

\textsuperscript{2}http://www.ee.cuhk.edu.hk/~jshao/CUHKcrowd_files/cuhk_crowd_dataset.htm
\textsuperscript{3}http://www.ee.cuhk.edu.hk/~jshao/WWWCrowdDataset.html
motion channels are proposed by uniformly slicing the video volume ($S$-$Vol$) along a set of probe lines from the dimensions $X$, $Y$, and $T$ respectively. This kind of representation was first introduced by E.H. Adelson et al. [62], and has been studied in different fields, such as video segmentation [63], human tracking [64], and gait analysis [65]. These previous works demonstrate the effectiveness of temporal $xt$- and $yt$-slices that explicitly profile the temporal evolutions of video contents or features across the probe lines determined by the slicing positions. Nevertheless, to date there was no attempt neither to adopt them to analyze crowds nor explore the potential and capability of them with deep neural networks. Importantly, such volumetric slices enable the learning of complex and long-term motion patterns. In addition, by multiple deep structures with various data fusion and weights sharing schemes upon different temporal slices, the proposed method is examined and proved to be efficient to learn the spatial and temporal clues for the application on crowd understanding [66].

5. The $S$-$Vol$ method is extended to allow slices extracting from not only the raw video volume but also any arbitrary feature cuboids, named as Slicing CNN ($S$-$CNN$). In this way, the appearance and dynamic information can be effectively extracted at a deeper layer of CNN that conveys richer semantical notion (i.e. groups and individuals). Besides, the appearance and dynamics have separate representations yet they interact seamlessly at semantic level. It consists of three CNN branches each of which adopts different 2D spatio- or temporal-filters. This design brings a few unique advantages to the task of crowd understanding [67]: (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). (2) Selectiveness – The semantic selectiveness exhibited by the 2D spatial filters additionally guides the discriminatively pruning of the 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, it is able to extract motion features of different semantic units, e.g. speed and acceleration in $x$- and $y$-directions.
1.5 Thesis Road Map

The thesis is organized into eight chapters. **Chapter 2** reviews related work on micro-, macro-, and group-level behavior analysis as well as video analysis with deep convolutional neural networks. The other six chapters are divided into two parts: one is Scene-Independent Group Profiling (i.e. Chapter 3) and the other one is General Crowd Attribute Learning (i.e. Chapters 4–8). **Chapter 3** first designs a group detection algorithm. Then the group properties and their visual descriptors are proposed. They are demonstrated to have potentials in multiple applications including crowd dynamic monitoring, crowd video classification, and crowd video retrieval. **Chapter 4** provides the details of establishing a large-scale crowd video dataset. Compared with the existing crowd video datasets, the proposed dataset is superior in both scale and diversity. Besides, the chapter describes the way to design and annotate 94 crowd-related attributes for describing each video in the dataset. In **Chapter 5**, a two-branch CNN model is proposed to achieve the multi-task learning. The input channels of the motion branch model are designed by extending group properties to generic scene properties. **Chapters 6 and 7** present long-term crowd feature learning by slicing the raw video volume and the feature cuboid respectively. With innovative model design, appearance and dynamic informations can be effectively learned from different levels, separately and interactively. All the approaches of Chapters 5–7 are evaluated on the crowd attribute dataset proposed in the Chapter 4. The conclusion is provided in **Chapter 8** which summarizes the contributions of these works and suggests several areas for future directions.
Chapter 2

Literature Review

During the last decade, the field of crowd analysis had a remarkable evolution from crowded scene understanding, including crowd behavior analysis [7; 13; 16–18; 68–80], crowd tracking [19–21], crowd counting/crowd density estimation [26; 27; 81–85], and crowd segmentation [22–24]. Much of this progress was sparked by the creation of novel crowd representations as well as the new and robust features and models. Since this thesis highlights on the crowd behavior analysis, the literatures of crowd tracking, crowd counting/crowd density estimation, and crowd segmentation are not thoroughly presented here. A survey written by Zhan et al. [86] provided a detailed review on the aforementioned topics. Whilst a recent review by KoK et al. [87] summarized the crowd-related studies inspired by biology and physics phenomenon. For crowded behavior analysis, a complete survey can be found in Li et al. [88]. Beyond the computer vision view, Le Bon [89] provided a comprehensive coverage on crowd analysis from the psychological view. Some specific techniques such as single-target/multi-target tracking and common-used low-level features like optical flow are not described thoroughly in this review. Hu et al. [90] wrote a survey to provide the details of these techniques.

Specifically, this review is structured into three sections: (1) crowd behavior analysis from micro-level and macro-level (Section 2.1), (2) detailed group-level analysis including group detection and group properties (Section 2.2), and (3) deeply learned features for crowd video understanding (Section 2.3).

2.1 Crowd Behavior Analysis

Most existing imagery-based crowd behavior analysis methods tend to treat a crowd either as a collection of individuals [8; 9; 68] or as an aggregated whole [7; 12; 22;
Figure 2.1: Different micro-level crowd behavior analysis: (a) social force model [7], (b) multi-target tracking with social behavior model [8], and (c) hierarchical activity representation [9].

The selection of crowd representations is governed by several factors such as scene density, camera viewpoint, and video frame rate.

2.1.1 Micro-Level Representation

From the microscopic level, some approaches [16; 91; 92] extracted moving pixels as visual features and jointly modeled single-agent behaviors with hierarchical Bayesian models. The well-known agent-based approach proposed by Helbing [93] has been used in crowd behavior analysis [7; 94] too (Figure 2.1(a)). It measured the interaction among individuals. For example, Braun et al. [94] extended the social force model by characterizing agents with different personalities and attributes, such as personal ID, dependence level, and desired speed. Some methods [8; 9] focused on multi-target tracking and measuring crowd behavior via individual correlations. Pellegrini [8] modeled social behavior between individuals to facilitate multi-target tracking (Figure 2.1(b)). Choi et al. [9] treated the crowd as one group with a collection of individuals and estimated collective activities between individuals. They tracked multiple targets and measured their activities by introducing a hierarchy of activity types (Figure 2.1(c)).

Beyond particle physics, Kaminka et al. [95] adopted Social Comparison Theory (SCT), a popular social psychology theory, to model crowd behaviors. The theory suggests that pedestrians would evaluate their current states through comparing themselves to others. Using this theory the study [95] generated improved pedestrian movements and accounts for group formation in pedestrians that are inter-related.
CHAP. 2. LITERATURE REVIEW

(a) (b) (c)

Figure 2.2: Different macro-level crowd behavior analysis: (a) represents the motion pattern with a fixed-size spatio-temporal volume sets [10], (b) decomposes the entire scene into semantic regions for anomaly detection [11], and (c) finds co-occurring activities by computing the codewords of optical flow [12].

2.1.2 Macro-Level Representation

Differing from the microscopic-level crowd analysis, modeling crowd behaviors from the macroscopic level treats the crowd as a whole. Zhou et al. [72] proposed a Mixture model of Dynamic pedestrian-Agents (MDA) for learning crowd behaviors. Mahadevan et al. [18] represented activities by mixture of dynamic textures (DT). Using spatio-temporal features to detect crowd behaviors [10–12; 96; 97] is another popular macroscopic approach. For instance, Kratz et al. [10] represented the motion pattern in a spatio-temporal volume with a Gaussian distribution (Figure 2.2(a)). Zelnik-Manor et al. [97] applied space-time intensity gradients as salient features to measure the disparity between activities. And Li et al. [11] decomposed a scene into semantic regions which capture coherent activities, differs from those observed in other regions (Figure 2.2(b)). Instead of directly computing spatio-temporal gradients, Zaharescu et al. [96] computed spatio-temporal oriented energy with second derivative of 3D Gaussian filters. Behaviors were modeled by the histogram of relative energy at different orientations and scales. It is worth pointing out that many crowd modeling approaches are scene-specific [12; 16; 98], i.e. activity models learned from a specific-scene cannot be applied to other scenes (Figure 2.2(c)).

2.2 Group Analysis in Crowd

Both microscopic- and macroscopic-level analyses have their own weaknesses. The object-centered micro-level approaches require explicit detection and segmentation of
individuals from crowd. These techniques are infeasible in crowded scenes where inter-object occlusion is severe. The activity representation employed by holistic methods, e.g. optical flow codewords [12; 70], dynamic texture [23], and grid of particles [7; 22], are useful for learning scene-level spatio-temporal pattern, but not directly applicable for learning group-level properties, which requires finer group segregation. In the following subsections, the comprehensive analysis on group in crowd is presented.

2.2.1 Group-Level Representation

A number of approaches [99] have been proposed for recognizing group activities such as meeting and fighting. These studies tend to analyze small social groups, with focus on scenario-specific predicates learning [55], contextual and interaction modeling [9; 34; 100; 101], and social signaling analysis [102]. Specifically, Chang et al. [55] proposed a probabilistic algorithm to softly assign individuals into groups. Lan et al. [34] introduced a discriminative model to jointly recognize group behaviors, individual behaviors and interactions among pedestrians. Zha et al. [103] used multi-camera contexts to analyze group activities. The standard Hidden Markov Model (HMM) has also been developed to interpret group activities [104; 105]. Zhang et al. [105] learned different levels of actions exhibited from individuals to group of people based on layered HMMs framework.

2.2.2 Group Detection

Moussaid et al. [28] found that group members tend to walk side-by-side at low crowd density and the formation is bent to a V-shape pattern as the density increases. Some works grouped pedestrians by analyzing their relative distances and moving patterns. Haritaolu et al. [106] regarded group detection as a graph partition problem, whilst Ge et al. [107] discovered small groups by bottom-up hierarchical clustering (HC) of trajectories based on pairwise objects’ velocity and distance. Another trajectory-based approach was proposed by Zhou et al. [13] and it used a so-called Coherent Filtering (CF) algorithm to segment coherent motion in crowd. Li et al. [108] used trajectory information of multiple objects to learn models for segmenting two group patterns: offensive and defensive patterns. Whilst Solera et al. [109] detected social groups by means of a Correlation Clustering method based on trajectories.
As shown in the experiments, the above methods are either too sensitive to tracking errors or unscalable to extremely crowded scenes. Importantly, neither of them learn group properties further nor analyze crowd behaviors at the group-level.

### 2.2.3 Group Properties

Unfolding group properties and their descriptors have largely been ignored in computer vision, although they are well-known in other disciplines. In biology, Zhang et al. [30] studied collective motion in bacterial colonies to prevent disease spreading while McPhail [110–112] treated crowds and collective behavior as synonyms in sociology. Makris et al. [113] conducted quantitative study on the collective spatial and temporal processes formed by ocean shoals. In the field of sociology and psychology, sociologists measured inter-group relationship based on social conflict, discrimination, and prejudice [114], while psychologists explored how intra-group interactions influence the completion of group tasks [115]. Russell et al. [116] stated that a social group should poses “membership stability”. Carley [117] attempted to explain why some groups endure longer and more stable than others. Dahrendorf [118] stated that the social conflict was one of the central themes in social research. Wheelan [29] pointed out that conflict may be caused by competition for resources, goal incompatibility, and contentious influence tactics. Yi [76; 78] proposed multiple attributes to facilitate stationary group invariance measurement, while ours measures the dynamics of the group movements.

### 2.3 Deep Convolutional Neural Networks (CNNs) in Video Analysis

The approaches discussed above are all traditional hand-crafted features and models. Recently, the deep convolutional neural networks (CNN) have gained a tremendous improvement on many tasks in computer vision. A comprehensive summary of deep neural network can be found in the book [119] written by Goodfellow et al.

In recent years, the improvements on GPU hardwares enable CNNs with millions of parameters, leading to significant improvements especially in single-shot image understanding, such as image classification [120–123], object detection [124], and semantic segmentation [24; 125; 126]. Notably, the current state-of-the-arts on ImageNet ILSVRC [127; 128] and PASCAL VOC [129] benchmarks are all variants of the CNNs.
In addition, the CNN models trained on ImageNet ILSVRC [127], served as pre-trained models, had shown an outstanding performance on many other image-related tasks (e.g. image recognition, semantic segmentation, object detection, and image captioning) with or without fine-tuning [130–134]. Even adopting SVM as the classifier on the features learned by these models have shown to yield good performance [135].

Compared to apply CNN to the static image analysis, there are relatively few works on the video analysis [51–54; 136–138]. 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 [51; 136; 139]. 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. [52] studied different schemes of sampling frames and fused their features at multiple stages. Due to the limitation of current available CNN models, only a small amount of sampled frames are feasible. 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 [140–142]. It shows that space and time are not equivalent components and thus should be learned in different ways. Ignoring such prior knowledge and learning feature representation blindly would not be effective. Alternatively, two-branch CNN models [53; 54] have been proposed to extract appearance and dynamic cues separately with independent 2D CNNs and combine them in the top layers. They designed several input alternatives for temporal stream (i.e. optical flow and trajectory). Both of the two kinds of inputs are frame-based and cannot maintain adequate temporal information. 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.

Few works have attempted to adopt deep neural network in crowd analysis. Zhang et al. [81] handled both crowd density estimation and crowd counting with a CNN model. To handle the cross-scene crowd counting, they proposed a data-driven method to fine-tune the trained CNN model for the target unseen scene. Another similar work done
by Hu et al. [143] adopted a CNN for extracting features and count individuals in the crowded scenes with mid-level or high-level densities. Kang et al. [24] proposed a fully-convolutional neural network (FCNN) on crowd segmentation. The FCNN structure accepts inputs of arbitrary sizes and extracts local features, which is suitable for capturing scene-independent crowd properties. A general semantic segmentation designed by Long et al. can be found in [130].
Part I

Scene-Independent Group Profiling
Chapter 3

Profiling Group Properties

Understanding group-level dynamics and properties is scientifically important and practically useful in a wide range of applications, especially for crowd understanding. This chapter shows that fundamental group-level properties, such as intra-group stability and inter-group conflict, can be systematically quantified by visual descriptors. This is made possible through learning a novel Collective Transition prior, which leads to a robust approach for group segregation in public spaces. From this prior, a rich set of group property visual descriptor is devised. Extensive experiments on hundreds of public scene video clips demonstrate that such property descriptors are complementary to each other, scene-independent, and they convey critical information on physical states of a crowd.

This chapter’s structure is summarized in Fig. 3.1. The proposed group detection approach is stated in Section 3.1. Section 3.2 illustrates four scene-independent group properties and their corresponding visual descriptors. All the experiments are implemented on a new crowd dataset described in Section 3.3. The results are reported and discussed in Sections 3.4, 3.5, and 3.6. Finally, conclusions are drawn in Section 3.7.

3.1 Group Detection

In this chapter, a group is regarded as a set of members with a common goal and collective behaviors. Given a short video clip of $\tau$ frames, a set of groups $\{G_i\}_{i=1}^m$ are detected. There are totally $m$ groups, and each group $G_i$ encompasses a set of tracklets $\{z_k\}_{k=1}^n$ detected by a tracker$^2$. The Kanade-Lucas-Tomasi (KLT) [144] feature point

---

$^1$Tracking results are usually described as trajectory, and short-range trajectory fragments are defined as tracklets. The tracklets often occur in crowd videos because of frequent occlusion and complex clutters.

$^2$It should be $\{z_{i,k}\}_{k=1}^n$, but is shorten as $\{z_k\}_{k=1}^n$ for brevity.
3.1.1 Collective Transition Prior

Precise group detection in crowd is challenging due to complex interaction among pedestrians. It is assumed that pedestrian movements in a scene are intimately governed by a finite number of Collective Transition (CT) priors. These priors are discovered simultaneously with the group detection process. Group detection can be then made more robust by considering the temporal smoothness and consistency enforced by the priors. Furthermore, Sec. 3.2 demonstrates that certain group properties can be readily derived from the discovered CT priors.

Each pedestrian group has a specific CT prior, which can be discovered from a video clip. More precisely, for \( n \) tracklets, \( \{z_k\}_{k=1}^n \), \( m \) Markov chains are assumed to

---

**Figure 3.1:** An overview of the framework. Crowd groups are first detected with the CT prior. Based on the CT prior, a set of visual descriptors are devised to quantify four fundamental intra- and inter-group properties, namely collectiveness, stability, uniformity, and conflict, for each group. The proposed visual descriptors are applied for crowd group analysis, crowd dynamics monitoring, crowd video classification and crowd video retrieval.
exist, where \( m < n \) and \( m \) is inferred automatically. Each Markov chain is a time-series model with the form of

\[
z_k^t = Az_{k}^{t-1} + v^t, \tag{3.1}
\]

where the continuous observation \( z_k^t \) evolves by a transition matrix \( A \in \mathbb{R}^{3\times3} \). Gaussian noise \( v^t \sim N(0, Q) \) is assumed between transition. Let \( z_k^t = [x^t, y^t, 1]^T \) represent the position of a pedestrian in homogeneous coordinates\(^3\) and the initial observation \( z_1^t \) follows a Gaussian distribution \( N(\mu, \Sigma) \). \( \Theta = \{A, Q, \mu, \Sigma\} \) is denoted as the parameters of the chain. \( A \) represents the CT prior, which reveals the collective motions of all the members in a group, while \( \{\mu, \Sigma\} \) ensure that group members are spatially proximate at the initial frame. The following subsections discuss how to learn this prior and perform robust group detection simultaneously.

### 3.1.2 Group Detection by Collective Transition

The key idea is to search for pedestrian groupings that fit well to the discovered priors within the video clip. The method is robust as it permits fragmented tracklets that fail to sustain over the whole clip. In particular, the missing data of \( z_k \) can be inferred with an Expectation-Maximization (EM) algorithm. It is thus suitable for group detection in dense crowds. In addition, it relies on local spatio-temporal relationships and velocity correlations without assumption on the global shape of the pedestrian group. Therefore it can be applied to scenes with different scales and perspectives.

The key steps of learning the CT priors for group discovery are summarized in Algorithm 1.

**Step-1: Generate coherent filtering clusters**: A set of initial tracklet clusters \( \{C_j\}_{j=1}^n \) are generated by Coherent Filtering [13], which is a clustering technique for the detection of coherent motion from time-series data and based on a so-called Coherent Neighbor Invariance that characterizes the local spatio-temporal relationships of individuals. These clusters do not align with the group perception defined in this thesis perfectly but can serve as the basis for finding the final tracklet groups \( \{G_i\}_{i=1}^m \). Examples are shown in Fig. 3.2a.

**Step-2: Identify anchor tracklets**: The iterative scheme begins by randomly picking a cluster \( C_i \) and finding its anchor tracklet \( z_i^* \) with long duration and low variance

\(^3A\) represents projective transforms, which include translation, contraction, expansion, dilation, rotation, shear, and their combinations.

\(^4\)The cluster used here refers to the initial group result.
Algorithm 1: Group detection by collective transition.

**Input:** Tracklets $\{z\}_{k=1}^m$ in a video clip.

**Output:** $m$ tracklet groups, $\{G\}_{i=1}^m$.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$i = 1$, generate coherent filtering clusters ${C}_{j=1}^r$.</td>
</tr>
<tr>
<td>if</td>
<td>${C} \neq \emptyset$ then</td>
</tr>
<tr>
<td>2</td>
<td>Identify an anchor tracklet, $z_i^*$;</td>
</tr>
<tr>
<td>3</td>
<td>Discover seeding tracklets set $S_i$ from $C_i$;</td>
</tr>
<tr>
<td>4</td>
<td>Learning the collective transition prior $A_i$ with $S_i$;</td>
</tr>
<tr>
<td>5</td>
<td>Perform group refinement to discover $G_i$;</td>
</tr>
<tr>
<td></td>
<td>${G} = {G} \cup G_i$, ${C} = {C} \setminus G_i$, $i = i + 1$;</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.2: (a) Coherent filtering [13] fails to distinguish two subtle groups. It can be addressed through discovering (b) a representative anchor tracklet (in red) and subsequently (c) a set of seeding tracklets to infer a group-specific CT prior. (d) With refinement based on the CT prior, two groups are separated. (Fig. 3.2b).

- **Long duration:** A well-tracked pedestrian owns at least one long-duration trajectory. Thus he has more opportunities to meet with people on his route, and generate a group around him.

- **Low variance:** In most cases, unusual velocity rarely occurs within a group, but over an individual, e.g. running and sudden stop. Therefore, the individual with velocity of high variance cannot be a lasting member in one group.

**Step-3: Discover seeding tracklets:** As shown in Fig. 3.2c, a set of *seeding tracklets*, $S_i$, are selected with the following criteria: (1) they are also from $C_i$; and (2) have high velocity correlation with $z_i^*$,

$$\frac{\langle v_{z \in C_i}, v_{z^*} \rangle}{\|v_{z \in C_i}\| \cdot \|v_{z^*}\|} > \eta,$$

(3.2)

where $\eta$ is a threshold.

**Step-4: Learning CT prior:** The seeding tracklets discovered in Step-3, together
with the anchor tracklet, form a set $S_i$, which is used to learn a representative CT prior with EM. The CT prior is used to refine the group itself in Step-5.

The details of the EM optimization can be found as follows. Given the Markov chain defined in Equation (3.1), the transition probability is

$$p(z_t^k|z_{t-1}^k) = N(z_t^k|Az_{t-1}^k, Q). \quad (3.3)$$

Assuming that the noise introduced by the observation is Gaussian distributed, the observation probability is represented as $p(x_t^k|z_t^k) = N(x_t^k|z_t^k, R)$, where $R$ is a given diagonal matrix. Let the initial observation follows a Gaussian distribution $z_1^k \sim N(z_1^k|\mu, \Sigma)$, the joint likelihood can be formulated with respect to the parameters $\Theta = (A, Q, \mu, \Sigma)$ for all the observations $X_{\tau}^k = \{x_1^k, \ldots, x_{\tau}^k\}$ and the hidden tracklet $Z_{\tau}^k = \{z_1^k, \ldots, z_{\tau}^k\}$ as

$$p(X_\tau^k, Z_\tau^k; \Theta) = p(z_1^k) \prod_{t=2}^{\tau} p(z_t^k|z_{t-1}^k) \prod_{t=1}^{\tau} p(x_t^k|z_t^k). \quad (3.4)$$

The parameters $\Theta$ can be effectively estimated by EM algorithm aiming at maximizing the marginal likelihood $L(\Theta; X_{\tau}^k) = p(X_{\tau}^k; \Theta) = \sum_{Z_{\tau}^k} p(X_{\tau}^k, Z_{\tau}^k; \Theta)$ as

$$\text{E-Step: } Q(\Theta|\Theta^{(n)}) = \mathbb{E}_{Z_{\tau}^k|X_{\tau}^k, \Theta^{(n)}} [\log L(\Theta; X_{\tau}^k, Z_{\tau}^k)] \quad (3.5)$$

$$\text{M-Step: } \Theta^{(n+1)} = \arg \max_{\Theta} Q(\Theta|\Theta^{(n)}). \quad (3.6)$$

The optimisation typically converges in a few iterations.

**Step-5: Group refinement**: Each tracklet $z$ in the initial cluster $C_i$ is fitted by $A_i$ of the $i^{th}$ Markov chain. The fitting error $\epsilon$ of a tracklet is defined as

$$\epsilon = \frac{1}{\tau - 1} \sum_{t=1}^{\tau-1} \|Az^t - z^{t+1}\|^2. \quad (3.7)$$

Any tracklet with $\epsilon < \delta$ is retained to construct $G_i$. Unqualified tracklets will need to repeat the iterative process to be considered for a different group. A refined group is shown in Fig. 3.2d.

### 3.2 Group Descriptors

This thesis formulates a set of descriptors to quantify group properties (Table 3.1). The first three quantify the spatio-temporal evolvement of intra-group structure, whilst the fourth characterizes inter-group interaction. Sections 3.5 and 3.6 show that they complement each other to perform well on scene-independent group state analysis and crowd video classification.
§ 3.2.1. Collectiveness

<table>
<thead>
<tr>
<th>Property</th>
<th>Descriptor</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collectiveness</td>
<td>$\phi^{\text{coll}}(G)$</td>
<td>3.8</td>
</tr>
<tr>
<td>Stability</td>
<td>$\phi^{\text{stab}}(G)$</td>
<td>3.14</td>
</tr>
<tr>
<td>Uniformity</td>
<td>$\phi^{\text{unif}}(G)$</td>
<td>3.17</td>
</tr>
<tr>
<td>Conflict</td>
<td>$\phi^{\text{conf}}(G)$</td>
<td>3.18</td>
</tr>
</tbody>
</table>

Table 3.1: List of group descriptors.

An analogy between a point and a member is established to facilitate explanation. A detected group has $n$ members in a frame, which form a $K$-NN graph, $G(V,E)$, whose vertices $V$ represent the members, and member pairs are connected by edges, $E$. The edges are weighted by an affinity matrix $W$, with elements $w_{ij} = \exp(-d_{ij}^2/\sigma^2)$, where $d_{ij}$ is the spatial distance between two members. The set of nearest neighbors of a member $z$ is denoted as $\mathcal{N}_1^z, \ldots, \mathcal{N}_\tau^z$ at every frame of a given clip. The descriptors in details are discussed next.

3.2.1 Collectiveness

The collectiveness property indicates the degree of individuals acting as a union in collective motion. It is a fundamental and universal measurement for various crowd systems [74; 145]. A collectiveness measurement for the whole video was proposed in [74] using manifold learning. In contrast, the collectiveness in this thesis is quantified at the group level with the proposed collective transition prior $A$, since it captures the coherent motion of all group members. In particular, the collectiveness of group $G$ is computed as

$$\phi^{\text{coll}}(G) = \frac{1}{|G|} \sum_{z \in G} \epsilon(z, A),$$  \hspace{1cm} (3.8)

where $|\cdot|$ denotes the cardinality of the input set, and $\epsilon(z, A)$ is defined in Eqn. (3.7).

A low value in $\phi^{\text{coll}}(G)$ suggests that the members of a group move coherently towards a common destination. The descriptor is useful for distinguishing low-collectiveness groups, e.g. in a train station or wet market, from high-collectiveness groups, e.g. observed during a marathon or on an escalator track.

3.2.2 Stability

The stability property characterizes whether a group can keep internal topological structure over time. It is analogous to molecules stability in a chemical system. In particular, stable members tend to (1) maintain a similar set of nearest neighbors; (2) keep a consistent topological distance with its neighbors throughout a clip; and (3) a member is less likely to leave its current nearest neighbor set. Following this idea, three stability descriptors are formulated as follows.
Figure 3.3: Quantify the change of the internal topological structure of one group by the Levenshtein edit distance. A larger distance indicates a lower stability.

(1) The first stability descriptor is computed by counting and averaging the number of the invariant neighbors of each member in the $K$-NN graph over time

$$
\phi_{\text{stab}}^a(G) = \frac{1}{|G|} \sum_{z \in G} \left( K - |\mathcal{N}_{\tau}^z \setminus \mathcal{N}_{\tau}^z| \right),
$$

(3.9)

where $|\mathcal{N}_{\tau}^z \setminus \mathcal{N}_{\tau}^z| = |\{z : z \in \mathcal{N}_{\tau}^z \text{ and } z \notin \mathcal{N}_{\tau}^z\}|$.

(2) The second stability descriptor is formulated to examine if the members keep consistent topological distance with their nearest neighbors, as shown in Fig. 3.3. This is achieved by first ranking the nearest neighbors of a member ($z$) in accordance to their pairwise affinity, and subsequently applying the Levenshtein string metric distance ($d_{\tau}^z$) [146] to compare the rankings at every two consecutive frames. $d_{\tau}^z = 0$ if two rankings are the same, and $d_{\tau}^z = K$ if the ranking indices of all the members have changed. Through collecting $d_{\tau}^z$ over $\tau$ frames, its histogram is constructed with $K$ bins, $h(z)$, for each member $z$. The second stability descriptor is then obtained as an averaged histogram

$$
\Phi_{\text{stab}}^b(G) = \frac{1}{|G|} \sum_{z \in G} h(z).
$$

(3.10)

It reveals information about the change of topological distances between members in a group.

(3) The third stability descriptor measures how likely a member would depart from its existing nearest neighbor set. Assuming a random walk behavior on all the group members, i.e. the members are allowed to transit freely within the group and join other members to form new neighborhood, the stability of a member can be measured as the difference between its initial and final transition probabilities. The transition
3.2.2. Stability

Figure 3.4: At time $t_1$, the ranked $K$-NN set of the current member (black disc) is $N^{t_1} = \{2, 5, 1, 4, 3, 6\}$. There are two extreme possible conditions at time $t_2$, i.e. $N^{t_2(1)} = \{7, 8, 10, 9, 11, 12\}$ or $N^{t_2(2)} = \{6, 3, 4, 1, 5, 2\}$. Though the latter one is more stable than the previous one, according to Levenshtein string metric distance, they cannot be distinguished.

The probability matrix $P \in \mathbb{R}^{n \times n}$ is initialized as

$$P = D^{-1}W,$$

where $D$ is a diagonal matrix whose elements are $D_{ii} = \sum_j w_{ij}$. The probability distribution of the $i$th member ‘walks’ to and ‘joins’ other members is defined by

$$q_i = e_i^T \left[(I - \alpha P)^{-1} - I\right],$$

where $q \in \mathbb{R}^{1 \times n}$, $I$ is the identity matrix, and $e_i = (e_1, \ldots, e_n)^T$ is an indicator vector with $e_i = 1$ and $e_i = 0$. The parameter $\alpha$ has a range of $0 < \alpha < 1/\rho(P)$, where $\rho(P)$ denotes the spectral radius of $P$. In the experiment, $\alpha$ is set to $0.9/K$. The stability of $i$th member is computed by measuring the Kullback-Leibler (KL) divergence [147] of $q_i$ between the first and final frames. A lower KL-divergence score, $s_{kl}$ suggests higher stability. The third stability descriptor is computed by averaging the scores across all members

$$\phi_{stab}^c(G) = \frac{1}{|G|} \sum_{z \in G} s_{kl}(z).$$

The final stability descriptor is

$$\Phi_{stab}(G) = [\phi_{stab}^a(G), \phi_{stab}^b(G), \phi_{stab}^c(G)].$$

The shortcoming can be complemented by three stability descriptors. For example,
3.2.3 Uniformity

Uniformity is an important property for characterizing homogeneity of a group in terms of spatial distribution. This property is in contrast to the two previous properties that measure temporal aspects. A group is uniform if their members stay close with each other and are evenly distributed in space. A non-uniform group has a tendency to be further divided into subgroups. A comparative example of uniform and non-uniform groups is shown in Fig. 3.5a and 3.5b.

The uniformity is quantified by inferring the optimal number \( c^* \) of graph cuts on the K-NN graph. A higher \( c^* \) suggests a higher degree of non-uniformity. A hierarchy of clusters \( \mathcal{H} \) is generated with agglomerative clustering [148] and the modularity function \( Q \) [149] is used to find \( c^* \). Specifically, given a cluster number \( c \), a graph partition \( \{V_1, \ldots, V_c\} \) is obtained from \( \mathcal{H} \). Computing \( Q_c \) for \( c \in \{1, \ldots, C\} \) and its maximum value suggests the optimal number of cuts:

\[
\begin{align*}
c^* &= \arg \max_{c \in \{1, \ldots, C\}} Q_c \quad (3.15) \\
given \quad Q_c &= \sum_{i=1}^c \left[ \frac{A(V_i, V_i)}{A(V, V)} - \left( \frac{A(V_i, V)}{A(V, V)} \right)^2 \right] , \quad (3.16)
\end{align*}
\]

where \( A(V', V'') = \sum_{i \in V', j \in V''} w(i, j) \). Examples are shown in the last column of

Figure 3.5: (a) and (b) show a uniform and a non-uniform group, respectively. The original coherent group detection, the sub-groups obtained through further clustering, and the optimal number of cuts inferred by modularity function are shown from left to right.

as shown in Fig.3.4, only with the second descriptor, it cannot measure stability well from \( t_1 \) to \( t_2^{(1)} \) and from \( t_1 \) to \( t_2^{(2)} \) since both of them have the same Levenshtein distance as 6. And this can be complemented by another two stability descriptors.
3.2.4 Conflict

The conflict property characterizes interaction/friction between groups when they approach each other. The spatial distribution and level of conflict experienced by a group can be visualized on a 2D normalized map as shown in Fig. 3.6. Such a map is informative for crowd understanding as it contains rich information about different natures of inter-group interactions observed in different scenes. On this map, the group contour is obtained as the outer boundary of the internal members, whereas a conflict point is defined as a member with external group members in its $K$-NN set, $N$. Note that the $K$-NN sets defined here differ from those employed earlier, as the current sets are allowed to include members from external groups.

To represent the conflict map compactly with invariance to scales, a Conflict Shape Context (CSC) descriptor is formulated, inspired by shape context [150]. The first step is to capture the spatial distribution for each conflict point by computing a histogram of the relative coordinates of group contour points. This is achieved by introducing a polar coordinate system [150] centered on each conflict point, and computing the frequency of contour points in the bins. 8 equally spaced angle bins and 5 equally spaced radius bins are used. The second step is to perform K-means clustering over

**Figure 3.6:** (a) Two conditions of conflict location in groups. Hot color represents a high degree of conflict. (b) Conflict distribution of groups. The map is normalized with the groups’ center, moving direction, and the largest distance from groups’ contour to its center.

Fig. 3.5. They show that a non-uniform group has a relatively higher number of cuts. Since the uniformity of a group may change as group evolves, the uniformity is thus measured by the mean $\mu_{c^*}$ and variance $\sigma_{c^*}$ of the optimal number of cuts over time:

$$\Phi_{\text{unif}}(G_i) = \{\mu_{c^*}, \sigma_{c^*}\}. \quad (3.17)$$

3.2.4 Conflict
Figure 3.7: Comparative results of group detection with four methods. Groups are distinguished with colors. Red color indicates outliers. Arrows are moving directions. Best viewed in color.

<table>
<thead>
<tr>
<th>Methods</th>
<th>NMI</th>
<th>Purity</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTM [23]</td>
<td>0.30</td>
<td>0.71</td>
<td>0.68</td>
</tr>
<tr>
<td>HC [107]</td>
<td>0.27</td>
<td>0.73</td>
<td>0.62</td>
</tr>
<tr>
<td>CF [13]</td>
<td>0.42</td>
<td>0.78</td>
<td>0.73</td>
</tr>
<tr>
<td>CT</td>
<td>0.48</td>
<td>0.83</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Figure 3.8: Left: quantitative comparison of group detection methods. Right: relative improvement of the proposed approach (CT) compared with DTM, HC, and CF.

training clips to build a vocabulary on the histograms, and produce Bag of Words (BoW) representation. Using locally constrained linear coding [151], the $i^{th}$ conflict point has a distribution $u_i$ over the vocabulary. The level of conflict of this conflict point is computed based on the CT prior introduced in Sec. 3.1.1

$$
\epsilon_{i}^{conf} = \frac{1}{|N_i|} \sum_{z \in N_i} \epsilon(z, A),
$$

The $\epsilon(z, A)$ is defined in Eqn. 3.7, and $A$ is the CT prior of the group where the conflict point is residing. Intuitively, if the nearest neighbors of a conflict point are mostly external members that do not fit well to $A$, a high value in $\epsilon^{conf}$ is obtained. The final conflict property of a group is computed by max pooling $\{u_i\}$ weighted by $\{\epsilon_{i}^{conf}\}$ as in [151].
3.3 Group-based Crowd Dataset

Evaluations are conducted on a new CUHK Crowd Dataset. It includes crowd videos with various densities and perspective scales, collected from many different environments, e.g. streets, shopping malls, airports, and parks. It consists of 474 video clips from 215 scenes, among which 419 clips were collected from Pond5\(^5\) and Getty Image\(^6\), and 55 clips were captured by us. It is larger than any existing crowd datasets [20; 22; 74] (they are actually covered in the proposed dataset) in terms of scene diversity and clips number. Although the video clips have various lengths, only the first 30 frames from each clip are adopted for implementing the proposed approach\(^7\). In all experiments, it is guaranteed that there is no overlap on scenes in training and test sets to demonstrate the scene-independence of the proposed descriptors. The full video clips are available in the dataset. The resolutions of videos are different, varying from 480 × 360 to 1920 × 1080. The original resolutions are kept for all videos. The frame rates are also different, varying from 20 to 30 frames per second. And all experiments are conducted on the extracted frames with respect to their corresponding frame rates.

The ground truth of group detection, group state analysis, crowd video classification, and crowd video retrieval are manually annotated and checked by multiple annotators. All the experiments are implemented on a MATLAB platform. The results reported in the following sections were measured on a 3.3 GHz Intel Core i7 processor with 16 GB RAM.

3.4 Experimental Results on Group Detection

3.4.1 Comparison Results

Tracklets from 300 video clips are manually annotated into groups for evaluation based on the criterion that members in the same group have a common goal and form collective movement. Tracklets not belonging to any group are annotated as outliers. The proposed group detection method of using Collective Transition priors (CT) is compared with three state-of-the-art approaches: mixture of dynamic texture (DTM) [23], hierarchical clustering (HC) [107], and coherent filtering (CF) [13]. Examples of the ground truth and the detection results in comparison are shown in Fig. 3.7.

DTM well separates background and simple group motions, but it performs poorly on complex and mixed group motions. Besides, it requires manual specification of group number (this thesis provides ground truth as input) and for each clip it takes hundred-fold longer time than the proposed method. HC hierarchically clusters tracklets with

\(^5\)http://www.pond5.com/

\(^6\)http://www.gettyimages.com/

\(^7\)The Section 3.6.2 also compares the video classification results by 30 frames with that by 75 frames.
velocity and spatial constraints and does not consider group dynamic prior. It thus leads to more errors than CT. CF detects coherent motion with a neighborhood measurement without modeling dynamics shared by the whole group. It is thus sensitive to tracking failures. This can be observed in the first row of Fig. 3.7, where CF splits a group moving in the same direction into subgroups. Moreover, CF first detects groups with coherent motion between consecutive frames, and then associates the groups through the whole clip. Its errors are therefore accumulated. In the second and third rows of Fig. 3.7, CF associates two groups moving in different directions into one due to errors made in single frames.

For quantitative evaluation, this thesis considers group detection as a clustering problem, and adopt three widely used measurements in clustering evaluation, i.e., Normalized Mutual Information (NMI) [152], Purity [153], and Rand Index (RI) [154]. The comparison is shown in Fig. 3.8. The bar chart on the right shows the relative improvement of the proposed method (CT) compared with DTM, HC, and CF.

3.4.2 Ablation Study

This subsection first measures the influence of parameters on group detection, and then analyzes the significance of different steps. The same evaluation criteria (i.e. NMI, Purity, and RI) are used here.

Parameter analysis.

There are two parameters $\eta$ and $\delta$ for group detection: $\eta$ defines the threshold of velocity correlation and $\delta$ determines the range of fitting errors of each tracklet during
group refinement. In this part, we show the sensitivity of these two parameters to the
group detection result. This analysis also provides how to choose a proper parameter
set. As shown in Fig. 3.9a, when keep the value of $\delta$ unchanged and let $\eta$ increase, the
three evaluation criteria show that the parameter $\eta$ does not have a remarkable impact
on the detection results when $\eta$ is smaller than 0.9. This is because the first condition in
seeding tracklets selection has already discarded some tracklets seldom correlating with
the anchor tracklet. Even though a small $\eta$ might make a large set of seeding tracklets,
the refinement step can help further filter out non-coherent tracklets. As $\eta$ increases
beyond 0.9, the curves significantly decrease. $\eta = 1$ leads to an extreme case that
almost only one tracklet (i.e. anchor tracklet) is selected as the seeding tracklet. In this
case, the criteria achieve the worst since too few seeding tracklets cannot infer accurate
priors. Meanwhile, the running time decreases when $\eta$ increases. This is because the
set of seeding tracklets becomes smaller as $\eta$ increases, and with a smaller and more
accurate set of seeding tracklets, it costs less time to get EM algorithm converged. For
instance, the average running time with $\eta = -1$ is 12.36 seconds, while with $\eta = 0.99$
is 7.65 seconds. In the same way, by fixing $\eta$, the evaluation results with respect to
different $\delta$ are shown in Fig. 3.9b. As shown in the first (NMI) and third (RI) figures in
Fig. 3.9b, a small $\delta$ might result in poor detection results due to the missing of many
coherent tracklets. Noted that the second figure (Purity) in the Fig. 3.9b varies little
as $\delta$ changes, and it presents a decreasing trend when $\delta < 5$. High purity might be
achieved when the number of clusters is large (e.g. Purity is 1 if each document gets
its own cluster.). And compared to the Purity, NMI can trade off the quality of the
clustering against the number of clusters. Furthermore, all the three curves nearly stop
increasing when $\delta > 5$. When $\delta$ continues increasing, the curves slightly decrease, and
the gap between the poorest score and the best score is within 0.2. It shows that the
proposed group detection algorithm is not too sensitive on $\delta$. The qualitative results
are shown in Fig. 3.10. Results in the first row are the worst because small $\delta$ makes
more clusters, and when $\delta$ is larger than 5, it can achieve a stable detection results.

Step analysis.

As stated in Sec 3.1.2, there are four steps in the group detection algorithm. The
key step is to select seeding tracklets to learn CT priors. Its significance is demonstrated
by comparing the detection results when omitting this step from the proposed method
(named as $CT \, w/o \, seeding$), with the complete one (named as $CT$). The quantitative

<table>
<thead>
<tr>
<th>Methods</th>
<th>NMI</th>
<th>Purity</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
<td>0.48</td>
<td>0.83</td>
<td>0.78</td>
</tr>
<tr>
<td>CT w/o seeding</td>
<td>0.41</td>
<td>0.80</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Table 3.2: Quantitative comparison of CT and CT w/o seeding tracklets.
Figure 3.10: Examples of group detection results with different parameter settings (respect to parameter $\delta = 3, 5, 10$) are shown in the first three rows. The last row shows the groundtruth. Groups are distinguished with colors, and the red crosses indicates outliers (best viewed on screen).

Figure 3.11: Running time evaluation. (a) shows the running time comparison between CT and CT w/o seeding tracklets, and (b) shows running time of CT with different tracklet numbers (blue) and group numbers (green).

results are shown in Table 3.2 and their running times are shown in Fig. 3.11a. Both detection accuracy and running time of CT w/o seeding are worse than those of CT. Without seeding tracklets, CT priors learned from the initial tracklets clusters are not accurate enough since these tracklets are too disordered to perform well in CT prior learning. Therefore, the detection results will be messed up if the prior cannot correctly measure the collective behavior of a group. On the contrary, the set of seeding tracklets can ensure the tracklets used in prior learning are collective and coherent which makes the detection results better. Moreover, the set of seeding tracklets is smaller than the
3.5 Analysis on Group Descriptors

More about the group descriptors can be discovered according to questions like: (1) Are different group descriptors correlated and complement each other? (2) Can these descriptors identify intrinsic group states (gases, fluids, and solid)? (3) Are these descriptors scene-independent? The following three subsections provide the quantitative and qualitative results to answer these questions.

3.5.1 Group Descriptors Correlation

This subsection examines the correlation of different group descriptors and show that they each play an important role and complement each other to describe a scene. Take collectiveness as an example, this section compare it separately with stability and conflict descriptors.
Based on “collectiveness” value (i.e. $\phi^{\text{coll}}$), the 927 labeled groups are ranked in an ascending order. The ranked group ID is regarded as the x-axis in Fig. 3.12a and Fig. 3.12b. The descriptors of “stability” and “conflict” are high-dimensional, and their correlations cannot be directly compared with “collectiveness”. Consequently, by projecting them onto their first principal subspace, their dimensions are reduced to 1.

The correlation between stability and collectiveness is shown in Fig. 3.12a and Fig. 3.12b, respectively. Fig. 3.12a suggests that the increase of $\phi^{\text{coll}}$ usually corresponds to the decrease of $\phi^{\text{stab}}$. As stated in Section 3.2.1, higher $\phi^{\text{coll}}$ value represents lower collectiveness. It shows that collectiveness, as a whole, presents a weak positive correlation with stability – the higher the collectiveness is, the higher the stability of the group will be. However, being examined in more details, groups with a similar level of collectiveness may not own similar stability. Two examples are shown at the bottom of Fig. 3.12a. The first example is a video of marathon in which pedestrians run coherently and stably, while the group (marked in pink color) in the second example also coherently moves towards the same direction, but intervened by the other surrounding groups, its structure is not stable. It shows that the collectiveness and stability both play an important role and complement each other to describe a scene.

Similarly, by reducing the dimension of conflict, the correlation between conflict and collectiveness is shown in Fig. 3.12b. Different from the stability, the correlation between conflict and collectiveness is much lower. Interestingly, there are many data around $-0.3$, since some groups do not have “conflict” interactions with other groups. Nevertheless, their collectiveness are not guaranteed to be similar. The examples show that groups with conflict may also have high collectiveness.

The next section shows that the proposed group descriptors, thanks to their complementary nature, are effective in identifying different intrinsic group states.
3.5.2 Capturing Intrinsic States with Group Descriptors

Research on crowd modelling and analysis [56–58] generally classifies crowd particles into the following states with an analogy of classifying different phrases of matter in equilibrium statistical mechanics, as shown in Fig. 3.13. It is assumed that the underlying physical models are different for different states.

- **Solid**: particles moving in the same direction collectively. Their relative positions remain unchanged, bounded by internal forces.

- **Pure fluid**: particles moving towards the same direction; however, their relative positions change constantly due to the lack of inter-particle forces.

- **Impure fluid**: this state is added by us to better capture the crowd characteristics. It is similar to pure fluid, but with invasion of particles from other groups.

- **Gas**: particles moving in different directions without forming collective behaviors with others.

These states are decided by multiple socio-psychological and physical factors including crowd density, goals, interactions and relationships of group members, and scene structures. There are 927 groups manually labeled as ground truth: 128 gas groups, 291 solid groups, 349 pure fluid groups, and 159 impure fluid groups. Half of the data is randomly selected for training and the remaining for test (the training and test sets do not contain the same scenes). All of the proposed descriptors together with “group size” are combined as features inputed to a non-linear SVM classifier. The confusion matrix averaged over 10 trials is shown in Fig. 3.14a. The average accuracy\(^9\) is 60%.

---

\(^8\)Group size is the number of tracklets in a group normalized by the total number of tracklets in a scene. It is useful for classifying gas groups. Some groups have one pedestrian with multiple feature points.

\(^9\)By first calculating the accuracy within each class and then averaging them, the biggest class will not dominate the average accuracy.
CHAP. 3. PROFILING GROUP PROPERTIES

Figure 3.15: (a) Representative frame with group detection results. Colors indicate different groups. (b) Distributions of different types of groups in a shopping mall and an escalator scene. Colors indicate group states automatically recognized with the proposed descriptors. Best viewed in color.

while the chance of random guess is 25%. The result shows the effectiveness of the group descriptors and their generalization power across scenes. It is understandable that pure and impure fluid groups are the most confusing classes, since some pure fluid groups have interactions with other groups on their boundaries. The major but subtle difference between the two classes is the spatial distributions of conflict points. Tracking errors also increase difficulty in separating these two classes. Fig. 3.14b and Fig. 3.14c show the effectiveness of each group descriptor on classifying different group states. It is observed that stability and conflict are most effective on classifying solid groups. Collectiveness and conflict are most effective on classifying pure fluid groups. Group size is effective for gas groups.

As examples shown in Fig. 3.15, in a large open area, pedestrians behave more like gas and fluid, while move as flying solid on an escalator track or in a queue. In the figure on the left, fluid groups appear frequently on the paths connecting entrances and exits regions, while gas groups locate randomly and they are isolated customers walking around. In the figure on the right, the states of groups transit between solid and fluid at the exits of escalators. These examples show that the proposed group descriptors capture the group states well to reflect various activities in a scene.

3.5.3 Scene-Independence of Group Descriptors

In order to evaluate the scene-independence of the proposed descriptors, this section clusters arbitrary regions across scenes and examine if each cluster could successfully
captures coherent regional flows observed in different scenes.

More precisely, the image space of each crowd video is divided into $N$ cells. The group descriptors are then extracted from each cell. Multiple groups might pass through the same cell. Therefore, the descriptors conveyed by all the groups that traverse this cell are gathered, and the average descriptors depict the dynamic of crowds occurs in this cell.

$K$-means algorithm is used here to cluster the cells of different videos. Examples of three clusters are shown in Fig. 3.16. For each cluster (displayed in row), there are three representative cells marked with red boxes. Similar to [155], these cell clusters have semantic meanings related to the scene layout like (a) bottleneck (b) blocking/crossing (c) lane, indicating that the particular locations inside the marked cells always present some unique crowd dynamics constrained by the facilities or layout in the scenes. Importantly, the fact that each cluster captures coherent regional flow verifies the scene-independence of the proposed descriptors.

3.6 Applications

This thesis explores different applications to demonstrate the potentials of the proposed group descriptors.
Figure 3.17: Monitoring crowd dynamics via group state transition as shown in the left. For each scene, two frames indicate representative states are shown in the right. (a) Pink-marked groups are monitored by recognizing their states over time. (b) Pedestrians inside red boxes are monitored. Their states change as they join different groups along the time line.

3.6.1 Crowd Dynamics Monitoring

The group states defined in Section 3.5.2 can help to monitor crowd dynamics over time. Typically, the transition of group states within one entire group or between groups exert valuable impact on the detection and prediction of crowd events. This subsection first extracts group descriptors for each group inside a crowded scene, and use the scene-independent model learned in Sec. 3.5.2 to recognize the group states over time.

Intra-group state transition. As shown in the first scene in Fig. 3.17a, the state of pink-marked group turns from pure fluid into impure fluid when the group collide with another group from the opposite direction. There is only one group in the second scene in Fig. 3.17a with its state transferred from pure fluid to solid. The runners accelerate at the beginning of a race with different velocities, and then keep a stable running pattern. Both examples suggest that critical points or events in crowd can be
### 3.6.2 Crowd Video Classification

This thesis also demonstrates the robustness and effectiveness of the proposed group descriptors in the application of classifying crowd videos instead of individual groups. There exist research studies [10; 16] on using holistic descriptors to classify crowd video

<table>
<thead>
<tr>
<th>Class name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Highly mixed pedestrian walking</td>
<td></td>
</tr>
<tr>
<td>2. Crowd walking following a mainstream and well organized</td>
<td></td>
</tr>
<tr>
<td>3. Crowd walking following a mainstream but poorly organized</td>
<td></td>
</tr>
<tr>
<td>4. Crowd merge</td>
<td></td>
</tr>
<tr>
<td>5. Crowd split</td>
<td></td>
</tr>
<tr>
<td>6. Crowd crossing in opposite directions</td>
<td></td>
</tr>
<tr>
<td>7. Intervened escalator traffic</td>
<td></td>
</tr>
<tr>
<td>8. Smooth escalator traffic</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3.3:** List of crowd video classes.

**Figure 3.18:** Confusion matrices of crowd video classification (darker color represents higher accuracy). Left: using holistic features in [10]. The average accuracy is 44%. Right: using the proposed descriptors. The average accuracy is 70%.

inferred through group state transition.

**Inter-group state transition.** The pedestrians inside the red box in the first example of Fig. 3.17b are moving out of escalator along the time line. They belong to the pink-marked “solid” group on the escalator. When they move out of the escalator and walk away, they split from the pink-marked group and turn to be a distinct violet-mark “gas” group. In the second example shown in Fig. 3.17b, pedestrians inside the red box originally belong to the orange-marked group and move towards the escalator. When they approach the escalator, they deviate from the orange-marked group and join into the group entering escalator and marked by pink. Their state then turns from the impure fluid (orange-marked group) to the solid (pink-marked group) state. These examples further demonstrate that either profiling the entire group or its members is useful and significant in crowd understanding and control.

### 3.6.2 Crowd Video Classification

This thesis also demonstrates the robustness and effectiveness of the proposed group descriptors in the application of classifying crowd videos instead of individual groups. There exist research studies [10; 16] on using holistic descriptors to classify crowd video
Figure 3.19: Crowd video classification results. Left: the representative frame of input video. Middle: classification results by holistic features [10]. Right: classification results by the proposed descriptors. The red bar represents correct prediction, and the green bar represents incorrect prediction.

clips. For instance, Kratz et al. [10] divided a video clip into spatio-temporal cubiod and extracted motion features from each cube. The experiments show that the specially designed group property descriptors are much more effective than generic features.

All the 474 video clips in the proposed dataset are manually assigned into 8 classes as shown in Table 3.3. The 8 classes are commonly seen in crowd videos and some are of special interest in crowd management and traffic control. For example, crowd merge and crowd crossing may cause traffic congestion and crowd disasters such as stampede. It is also important to keep escalator traffic smooth at the entrance and exit regions to avoid blocking, collisions, and potential dangers. In class 1, pedestrians in a scene walk in multiple directions with highly mixed behaviors. In classes 2 and 3, most pedestrians follow the main stream. In class 2, the relative positions of pedestrians are stable and there are rarely overtake events, while pedestrians in class 3 are not well organized. Most crowd videos can be generally classified into the above three categories. However, a few other classes (4 ∼ 8) which are of particular interest in crowd management are identified, distinguishing from the remaining crowd videos. Therefore, classes 1 ∼ 3 have excluded videos from classes 4 ∼ 8. All the 8 categories are classified together.

Leave-one-out evaluation is used. Each time one scene (which may include multiple
video clips) is selected for test, and the remaining scenes for training. Thus the cross-scene generalization capability is evaluated. If a video has multiple groups, the video descriptor is the average of a descriptor over groups. The non-linear SVM with RBF-kernel is used for classification. The confusion matrices are shown in Fig. 3.18. The average accuracy of the proposed approach is 70%, much higher than that of random guess (12.5%) and the result of using the holistic crowd scene descriptor proposed in [10] (44%).

This subsection also includes some further analysis:

1. **Compare non-linear and linear SVMs.** In comparison to non-linear SVM (70%), classification by linear SVM only yields an accuracy rate of 65%, which shows the effectiveness of adopting non-linear SVM.

2. **The effect of using different numbers of frames in video classification.** Specifically, results obtained by using the default 30 frames are compared with those with 75 frames. The results show that using more frames does not improve video classification accuracy significantly. Although longer frames contain more temporal information that can help motion modeling, the result by 75 frames is 72% and has only subtle improvement over that by 30 frames. It is observed that the group motions in each segmented clip remain similar inside one crowd video. A satisfactory video classification result can therefore be obtained based on fewer frames.

To further evaluate the proposed method qualitatively, Fig. 3.19 shows several examples and the probability of being classified into each class. The video will be assigned to the class with the largest prediction probability. From the results, holistic features [10] often mistakenly classify the crowd video into the second class and cannot recognize other classes well. This observation is consistent to the corresponding confusion matrix depicted in Fig. 3.18, which has a high score in recognizing class 2 and class 8 but poor in other classes.
3.6.3 Crowd Video Retrieval

This subsection reports the performance of the crowd video retrieval by the proposed descriptors. Each time a query video is picked from the CUHK Crowd Dataset, while the remained 473 videos form the set for retrieval. Similar to Sec. 3.6.2, an average descriptor over all the groups inside a video is adopted as the video descriptor. Euclidean distance is used to measure the similarity between the corresponding scalar descriptor components, while $\chi^2$-distance is for histogram-styled descriptor components. The performance is evaluated by the average precision in the top $k$ retrieved samples (AP@$k$). The retrieved annotations adopt the ground truth labels introduced in Sec. 3.6.2.

Fig. 3.20a shows that the proposed descriptors outperform the holistic features [10] by enhancing the average top 100 precision (AP@100) over all video classes from 22% to 32% (a relative improvement of 45%). The performance of the average precision of each video class defined in Sec. 3.6.2 is shown in Fig. 3.20b, compared with the holistic features, the proposed descriptors have superior performance over [10] on the first 6 classes, and have a comparable AP on the 7th class, while poorer on the last class. This is consistent with the classification results shown in Fig 3.18. The holistic features [10] can only well recognize the second and the last class, and especially had bad performance on the recognition of the 4th and 5th classes. Several query videos and their retrieval results are shown in Fig. 3.21. Even though the proposed descriptors do not involve any appearance features, they can well characterize the inherent features of

Figure 3.21: Exemplar results of crowd video retrieval. The first column shows the query videos, and the remains of each row show the top 5 retrieval results.
the same class of videos. The errors often occur between the second class and the last class. The fourth row gives a failure example, the query video belongs to the second class, while three of top 5 retrieval results are from the last class. This is because both of them have the similar pattern of “following a mainstream and well-organized”.

3.7 Summary

This chapter systematically studies the fundamental and universal group properties, which exist in various crowd systems, from the vision point of view. These properties are motivated by the socio-psychological studies and important in crowded scene understanding. A robust group detection algorithm is proposed through learning the collective transition prior. From a graph-driven view, this chapter designs a rich set of group-property visual descriptors, including geometric structure, topological structure, and collective degree. They are well applied to scene-independent group states analysis, crowd dynamics monitoring, crowd video classification, and crowd video retrieval.

As the first few attempts on profiling group properties and descriptors from computer vision, this research will inspire new applications and extensions in the future work. The proposed group detection and group descriptors are both based on tracklets. Thus some crowded scenes with bad tracking results (e.g. time-lapse crowd videos and crowd videos shot from horizontal view) or without tracklets (e.g. crowds sit still watching performance) are difficult to characterize group properties precisely. The future work can substitute tracklets by other medium conveying motion information. The analysis of group states transition can be extended to detect cross-scene crowd events and monitor pedestrian dynamics. The proposed descriptors not only exist in the group, but can be also extended to entire scene. The most important characteristic of these descriptors is scene-independent. Therefore, the enhanced descriptors can be applied to cross-scene crowd video retrieval.
Part II

General Crowd Attribute Learning
Chapter 4

Constructing WWW Crowd Dataset

Most of the existing public crowd datasets [11; 16; 72; 84; 85] contain only one or two specific scenes. Even the CUHK crowd dataset proposed in the Section 3.3 of Chapter 3 merely provides 474 videos from 215 crowded scenes. On the contrary, the proposed WWW dataset provides 10,000 videos\(^1\) with over 8 million frames from 8,257 diverse scenes, therefore offering a superiorly comprehensive dataset for the area of crowd understanding. The abundant sources of these videos also enrich the diversity and completeness. The comparison between the proposed WWW dataset and other publicly available crowd datasets is shown in Table 4.1. Over all the comparison items listed in the table, the proposed dataset surpasses the rest both in scale and diversity.

<table>
<thead>
<tr>
<th></th>
<th>CUHK (Sec. 3.3)</th>
<th>Collectiveness [74]</th>
<th>Violence [156]</th>
<th>Data-driven [20]</th>
<th>UCF [22]</th>
<th>WWW</th>
</tr>
</thead>
<tbody>
<tr>
<td># video</td>
<td>474</td>
<td>413</td>
<td>246</td>
<td>212</td>
<td>46</td>
<td>10,000</td>
</tr>
<tr>
<td># scene</td>
<td>215</td>
<td>62</td>
<td>246</td>
<td>212</td>
<td>46</td>
<td>8,257</td>
</tr>
<tr>
<td># frame</td>
<td>60,384</td>
<td>40,796</td>
<td>22,074</td>
<td>121,626</td>
<td>18,196</td>
<td>&gt; 8 million</td>
</tr>
<tr>
<td>resolution</td>
<td>multiple</td>
<td>670 × 1000</td>
<td>320 × 240</td>
<td>640 × 360</td>
<td>multiple</td>
<td>640 × 360</td>
</tr>
<tr>
<td>source</td>
<td>Getty Images, Pond5, surveillance</td>
<td>web, surveillance</td>
<td>YouTube</td>
<td>web</td>
<td>BBC Motion Gallery, Getty Images</td>
<td>Getty Images, Pond5, YouTube, surveillance, movies</td>
</tr>
</tbody>
</table>

**Table 4.1:** Comparison of WWW and other existing datasets. WWW offers the largest number of videos, scenes, and frames.

4.1 Crowd Video Construction

4.1.1 Collecting Keywords

In order to obtain a large scaled and comprehensive crowd dataset, a set of keywords is selected which related to common crowd scenarios (e.g. street, stadium, and rink) and

---

\(^1\)The average length of all videos is around 23 seconds, and its std. is around 26 seconds.
crowd events (e.g. marching, chorus, and graduation) for the sake of searching efficiency and effectiveness.

For the purpose of generalization, none of keywords referring to specific places are included, but used general keywords that describe the functionalities of places instead. For instance, “landmark” is preferable rather than names of specific places like “Time Square” and “Grand Central Station”. It is common sense that “landmark” attracts crowds of tourists. Some keywords are derived by one general vocabulary. For example, “building” can be identified as a “shopping mall”, “museum”, “conference center”; and “station” contains “(railway/subway) platform”, “subway passageway”, “ticket counter”. Besides keywords about functional places like “station” and “restaurant”, several specific types of places are also included, such as “escalator” and “stage”. Although these can be seen as objects, they are known to have high correlation with crowd.

4.1.2 Collecting Crowd Videos

WWW crowd dataset is constructed with the goal that crowds in videos should have various appearances, behaviors, viewpoints, background clutters and occlusions. For example, it should contain videos with different perspective effects and camera motions from stationary cameras, moving aerial vehicles, and handled cameras.

To this end, the gathered keywords were used to search for videos from multiple public video search engines including Getty Images\(^2\), Pond5\(^3\), and YouTube\(^4\). To increase the chance of retrieving crowd videos, “crowd” or “group” was added in most of the keywords, except keywords that explicitly describe the crowd (e.g. “chorus” and “marathon”). Besides these three sources, 469 videos from 23 movies were further collected\(^5\). To control the video quality, videos with blurred motion, synthetic crowd, and extremely short length were removed. The videos with extreme long-duration were cut out parts with crowd and split into several video clips. In addition, all the duplicated videos were filtered. The collection also covers major existing crowd video datasets such as [74; 156].

4.2 Crowd Attribute Annotation

Given a video collection of many different crowded scenes, there is a enormous number of possible attributes describing different scenarios, subjects, and events. The capacity

---

2http://www.gettyimages.com/
3http://www.pond5.com/
4http://www.youtube.com/
to infer attributes allows us to describe a crowded scene by answering “Where is the crowd?”, “Who is in the crowd?”, and “Why is crowd here?”. Importantly, when given a new crowded scene, it can still be described with these three types of cues (e.g., newly-wed couple [Who] are in the wedding [Why] at beach [Where]).

Furthermore, there is a large number of possible interactions among these attributes. Some attributes are likely to co-occur with each other, whilst some seem exclusive. For example, the scenario “street” attribute is likely to co-occur with subject “pedestrian” when the subject is “walking”, and also likely to co-occur with subject “mob” when the subject is “fighting”, but not related to subject “swimmer” because the subject cannot “swim” on “street”. In addition, there exist attributes that are grouped hierarchically, e.g. “outdoor/indoor” contains almost all the other attributes of location. Some attributes, like “stadium” and “stage” belong to both “outdoor” and “indoor”.

### 4.2.1 Crowd Attribute Gathering from Web Tags

The attribute taxonomy is constructed by first collecting tags from Pond5 and Getty Images, partially as shown in Fig. 4.1, as a form of wordle. By carefully examining the contents of the retrieved tags with a total number of 7000+, it emerged that people tend to describe videos with five types of tags: place, subject, motion, video shot, and environment condition. (1) **Place**, which contains general locations like “city”, “metropolis”, “street”, “square” and “indoor”, as well as specific locations like “USA”, “Asia” and “Time Square”, also some functional locations like “market”, “bridge” and “subway”. (2) **Subject**, which contains age (e.g. “adult 30s-40s” and “young adult”),

---

6 Other websites, i.e. YouTube, movies, and existing datasets do not have tag information.

7 [http://www.wordle.net/](http://www.wordle.net/)
gender (e.g. “male/female” and “man/woman”), function (e.g. “consumer”, “passenger” and “conductor”), religion (e.g. “Christian” and “multi-ethnic”), as well as some other general tags like “contains people”, “human” and “citizen”. (3) Motion, which contains common action (e.g. “walk”, “run”, and “applause”) and event (e.g. “chorus”, “fashion show”, and “pilgrimage”). (4) Video shot information, including viewpoint (e.g. “elevated view”, “wide shot” and “high angle”), quality (e.g. “hdtv” and “HD format”). (5) Environment condition, including weather (e.g. “rain”, “wet” and “sunny”) and seasons.

It is laborious and non-trivial to define attributes from these raw tags since the majority of them are not relevant to the problem this thesis are interested in, even not related to crowd, such as Video shot information and Environment condition. In addition, tags with the highest frequency (e.g. people, adult, time, and ethnicity) are


4.2.2 Crowd Attribute Labeling

16 annotators are hired to label attributes in the WWW dataset, and another 3 annotators to refine labeling for all videos. All the attributes in the proposed dataset are commonly seen and experienced in daily life, so the annotation does not require special background knowledge. Each annotator was provided with a sample set containing both positive and negative samples of each attribute. They were asked to select possible attributes for each test video in a label tool containing three attribute lists of where, who, and why, respectively. In every round, each annotator was shown a 10-second video clip and was required to label at least one attribute from each attribute.

\[\text{http://www.ee.cuhk.edu.hk/ jshao/WWWCrowdDataset.html}\]
Figure 4.4: Visualize users’ response time. The blue circles in (a) plot the response time of all annotators on labeling tasks, and the red line marks the average response time of each annotator. (b) shows the histograms of response time of different cues.

<table>
<thead>
<tr>
<th>Cues</th>
<th>Single Frame</th>
<th>Background</th>
<th>Background + Tracklet</th>
<th>Tracklet</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.82</td>
<td>0.71</td>
<td>0.74</td>
<td>0.41</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Table 4.2: User accuracy with four types of cues.

list without time constraint.

Fig. 4.3 shows several examples in the WWW Crowd dataset. As shown in the first row of Fig. 4.3, not all the video clips can tell a complete story in the form of “somebody” “do something” “at somewhere”. Therefore, before labeling, “ambiguous” options are added in each attribute list. Totally, the annotators labeled 2855 “ambiguous” among all the marked 980,000 labels, taking 0.1%, 0.2%, and 0.4% in Where, Who and Why, respectively. Two videos shown in the second row of Fig. 4.3 demonstrate that a video might have quite a number of attributes, i.e. multiple subjects doing different tasks at different locations in a single video. For instance, as shown in the second row of Fig. 4.3, a “fashion show” (why) video has the “model” (who) “walking” (why) on the “runway” (where) as well as “audience” (who) “sitting down” (why) to “watch performance” (why) in an “indoor” (where) room. After delivering this observation to the annotators, they are required to label all possible attributes that can be discovered from the video.

4.3 User Study on Crowd Attribute

Appearance and motion cues play different roles in crowd scene understanding. A plenty of positive results in domain of images have demonstrated that the single still images can work well on scene understanding and attribute recognition. How about crowd attribute recognition? Does it need motion information (i.e. tracklets)? And which type of data can be identified most easily and laboriously? This section conducts a user study on the WWW crowd dataset to investigate human performance if only one
§ 4.3. User Study on Crowd Attribute

![Figure 4.5:](a) The number of wrongly labeled samples with the background cue (indicated by blue bars) and how many of them can be corrected after adding the tracklet cue. (b) Accuracy comparison between the tracklet cue and tracklet + background. All the results are obtained from the user study described in Sec. 4.3.

8 users are distributed with four types of data, including single frame image, background, tracklets, and background with tracklets. The compared ground truth is the set of annotations in Section 4.2 from whole videos. To avoid bias, every user is provided with all the four types of data and randomly selected 10 ~ 15 attributes. Each set of data for each attribute was presented in a random order, and contained half positives as well as half negatives. Before starting labeling, each annotator is provided with 5 ~ 10 positive as well as negative samples to help them get familiar with the attributes. Users were informed that their response time would be recorded. This is important because the response time statistics are particularly useful in identifying label rationality and data quality.

9 The average image of all frames of each video.
4.3.1 User Response Time

The average response time of all the users is 1.1094 seconds, as shown in Fig. 4.4(a). This indicates that human can identify an attribute within a short time which proves that the attributes are well defined and consistent to human vision. Besides, it can be observed from Fig. 4.4(b) that the number of samples labeled within 2 seconds from tracklets is much smaller than the other 3 types of data, whereas the number of samples labeled from tracklets longer than 3 seconds are the largest. It shows that labeling with only tracklets is more laborious, and it is not easy for human to recognize crowd attributes simply from motions without seeing images.

4.3.2 User Accuracy

Table 4.2 shows that with single frames users can achieve much higher accuracy than with only tracklets or background. It means that the appearance of moving people and their poses are useful, but they are blurred on the background image. It is found that the background cue and tracklet cue are complementary. Figure 4.5(a) shows how many samples were wrongly labeled only with the background cue and how many of them were corrected after users also seeing the tracklet cue. Very few failure cases in the first 17 attributes are corrected by the tracklet cues, because these attributes belong to “where”. As shown in Figure 4.5(b), the tracklets perform poorly on recognizing attributes belonging to “where”, whereas more effective on the last 23 attributes belonging to “who” and “why”. However, an interesting observation is that “classroom” has the largest correct ratio by adding background to tracklets, since most of videos with “classroom” have no motion that students or teacher always sit still or stand still. These results demonstrate that motion is complementary to appearance information, and is especially important to identify the subjects with special motion in crowd.

4.4 Summary

This chapter builds a large-scale crowd dataset with 10,000 videos from 8,257 scenes, and propose 94 crowd-related attributes. This is a significant contribution to the field of crowd scene understanding. A user study is further designed to analyze how static appearance cues and motion cues behave differently and complementarily on the three types of attributes: “where”, “who” and “why”.

To jointly learn the crowd appearance and motion features, this chapter exploits a multi-task learning deep model, named as Two-branch CNN which contains the Deeply Learned Static Features (DLSF) and Deeply Learned Motion Features (DLMF). Its superiority is demonstrated on the WWW crowd dataset presented in the Section 4 by applying the learned models for recognizing attributes in unseen crowd videos.

5.1 Deep Network Structure and Model Setting

Fig. 5.1 shows the network structure of the proposed deep model. The network contains two branches with the same architecture. Simple notations are used to represent parameters in the networks: (1) Conv($N,K,S$) for convolutional layers with $N$ outputs, kernel size $K$ and stride size $S$, (2) Pool($T,K,S$) for pooling layers with type $T$, kernel size $K$ and stride size $S$, (3) Norm($K$) for local response normalization layers with local size $K$, and (4) FC($N$) for fully-connected layers with $N$ outputs, (5) The activation functions in each layer are represented by ReLU for rectified linear unit and Sig for sigmoid function. Then the two branches have parameters: Conv(96,7,2)-ReLU-Pool(3,2)-Norm(5)-Conv(256,5,2)-ReLU-Pool(3,2)-Norm(5)-Conv(384,3,1)-ReLU-Conv(384,3,1)-ReLU-Conv(256,3,1)-ReLU-Pool(3,2)-FC(4096). The output fully-connected layers of two branches are concatenated to be FC(8192). Finally, FC(8192)-FC(94)-Sig produce 94 attribute probability predictions. The loss function of the network is cross entropy as in Equation (5.1). The network parameters of Appearance branch are initialized using a pre-trained model for ImageNet detection task [123].

$$E = -\frac{1}{N} \sum_{n=1}^{N} t_n \log o_n + (1 - t_n) \log (1 - o_n) \quad (5.1)$$

where the $N = 94$ denotes the number of output neurons, $t_n$ ($n = 1, \ldots, N$) are the target labels and $o_n$ ($n = 1, \ldots, N$) are the output probability predictions.

The traditional input of deep model is a map of single frame (RGB channels) or multiple frames [52]. This chapter proposes three scene-independent motion channels
Figure 5.1: Deep model. The appearance and motion channels are input in two separate branches with the same deep architecture. Both branches consist of multiple layers of convolution (blue), max pooling (green), normalization (orange), and one fully-connected (red). The two branches then fuse together to one fully-connected layers (red).

as the complement of the appearance channels\(^1\). Some well-known motion features like optical flow cannot well characterize motion patterns in crowded scenes, especially across different scenes. Scientific studies have shown that different crowd systems share similar principles that can be characterized by some generic properties. Inspired by Chapter 3 that introduces several scene-independent properties (e.g. collectiveness, stability, and the conflict) for groups in crowd, these properties are observed also exist in the whole scene space and can be quantified from scene-level. After the reformulation, the collectiveness indicates the degree of individuals in the whole scene acting as a union in collective motion, and the stability characterizes whether the whole scene can keep its topological structures, and conflict measures the interaction/friction between each pair of nearest neighbors of interest points. Examples shown in the Fig. 5.2 illustrate each property intuitively.

5.2 Motion Channels

Chapter 1 introduces four descriptors (i.e. collectiveness, stability, uniformity, and conflict) from group level to characterize common features across different crowded scenes. Similarly, from scene-level, the collectiveness can indicate the degree of individuals acting as a union in collective motion, the stability characterizes whether individuals in the whole scene can keep relative topological structures over time, and the conflict can measure the interaction/friction between each pair of nearest neighbors\(^2\).

For motion channel generation, all the descriptors are defined upon tracklets detected by the KLT feature point tracker, and each of them is computed on 75 frames of

\(^1\)The channel input in the appearance branch is RGB single frame, simply selected from the first frame of each video.

\(^2\)The uniformity is not adopted here because it captures the homogeneity of a group in terms of spatial distribution, which cannot offer meaningful descriptions from holistic-level
5.2. Motion Channels

Figure 5.2: Motion channels. The first row gives an example to briefly illustrate three motion channels construction procedure. For each channel, two examples are shown in the second and third rows. Individuals in crowd moving randomly indicates low collectiveness, while the coherent motion of crowd reveals high collectiveness. Individuals have low stability if their topological structure changes a lot, whereas high stability if topological structure changes a little. Conflict occurs when individuals move towards different directions.

each video in the WWW dataset. This chapter first defines a $K$-NN ($K = 10$) graph for the whole tracklet point set. Since no groups are detected in advance, the descriptor proposed in [74] is more suitable to extract collectiveness for each tracklet point in the whole scene. The descriptor for stability is designed, following the similar idea in Chapter 3, by by counting and averaging the number of invariant neighbors of each point in the $K$-NN graph. It reveals the fact that the stable crowd needs to maintain a similar set of nearest neighbors. The conflict descriptor defined in Chapter 3 is based on the group-based transition prior, thus is not suitable in the case in this section. Instead, this descriptor is generalized by computing the velocity correlation between each nearby tracklet points within the $K$-NN graph. The per-frame descriptor map for each motion feature is then averaged across the temporal domain to output three motion maps, which act as the input of the deep model. Although a single frame owns tens or hundreds of tracklets, the total tracklet points are still sparse. These sparse points are then interpolated to output a complete and continuous feature map. The brief channel construction procedure is shown in the first row in Fig. 5.2.
Table 5.1: Compare deeply learned features with baselines. The last column shows the number of attributes (out of the total number of 94) on which the proposed deep features have higher AUC than baselines.

<table>
<thead>
<tr>
<th>Our Methods</th>
<th>mean AUC</th>
<th>Baselines</th>
<th>mean AUC</th>
<th># wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLSF</td>
<td>0.87</td>
<td>SFH</td>
<td>0.81</td>
<td>67/94</td>
</tr>
<tr>
<td>DLMF</td>
<td>0.68</td>
<td>MDH</td>
<td>0.58</td>
<td>85/94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DenseTrack [140]</td>
<td>0.63</td>
<td>72/94</td>
</tr>
<tr>
<td>DLSF + DLMF</td>
<td>0.88</td>
<td>SFH+MDH</td>
<td>0.80</td>
<td>78/94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SFH+DenseTrack</td>
<td>0.82</td>
<td>72/94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMP [10]</td>
<td>0.72</td>
<td>89/94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slow Fusion [52]</td>
<td>0.81</td>
<td>74/94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two-stream [53]</td>
<td>0.76</td>
<td>89/94</td>
</tr>
</tbody>
</table>

As shown in the Section 5.3.4, these motion channels can facilitate appearance to improve the performance on attribute recognition.

5.3 Experimental Results

5.3.1 Settings

The WWW dataset introduced in Chapter 4 is randomly split into training, validation, and test sets with a ratio of 7 : 1 : 2. Note that all the three sets are guaranteed to have positives and negatives of the 94 attributes, and they do not have overlap on scenes to guarantee that the attributes are learned scene-independently. In all the experiments, the area under ROC curve (AUC) is employed as the evaluation criteria.

5.3.2 Evaluation on Deeply Learned Static Features (DLSF)

To evaluate the proposed Deeply Learned Static Features (DLSF) from the appearance channels only, a set of state-of-the-art static features that have been widely used in scene classification are selected for comparison. Literature shows that Dense SIFT [157] and GIST [40] have good performance good on describing general image content, while HOG [158] has been widely used in pedestrian detection. They all have the potential of being applied to crowd scene understanding. The color histogram in the HSV color space and the self-similarity (SSIM) [159] descriptor are added to capture global information. In addition, local binary patterns (LBP) [160] is also employed to quantify texture in crowded scenes.

The six types of features are extracted from the first frame of each video and construct the static feature histogram (SFH) following a standard bag-of-words pipeline with K-means clustering and locality-linear coding [151]. Linear SVM is used to train

---

This ratio is adopted to do all the following experiments in Chapters 5, 6 and 7.

The first row in Fig. 5.1 with the last fully-connected layer is substituted by three fully-connected layers.
5.3.3 Evaluation on Deeply Learned Motion Features (DLMF)

The performance of the Deeply Learned Motion Features (DLMF), compared with two baselines, is shown in Table 5.1. One is the histogram of the proposed motion descriptor (MDH) in Sec. 5.2. And another is dense trajectory [140] showed state-of-the-art result in action recognition. Both baseline features are trained with independent classifiers via linear SVM similar to the SFH baseline.

According to the results shown in the third row of Table 5.1, DLMF outperforms the other two baselines by 10% and 5% on mean AUC respectively. Over 77% attributes, DLMF achieves higher AUC than the baselines. On the other hand, DLMF has a nearly 20% drop compared with DLSF. This is consistent with the observation from the user study in Table 4.2 that the motion cue is less effective on recognizing attributes compared with the appearance cue in general.

5.3.4 Evaluation on Combined Deep Model

The two-branch deep model combining DLSF and DLMF is shown in Fig. 5.1. It is compared with five baselines. The first two baselines are the combination of the static feature (SFH) with two motion features (MDH and dense trajectory). The third one is spatio-temporal motion pattern (STMP) extraction method [10], modeling the input video as the assembly of spatio-temporal cuboids. It combine both appearance and motion cues. The fourth baseline is the slow fusion scheme with multi-frames as input in deep model proposed in [52] recently. It is a state-of-the-art deep learning method for video analysis, and it has achieved the best performance in [52] for sports classification.
Figure 5.4: Good and bad attribute prediction examples are shown in (a) and (b). For each image, its top four attributes with the highest prediction scores with the proposed two-branch model (i.e. DLSF+DLMF) are shown. The heights represent prediction scores. Blue indicates correctly predicted attributes and orange indicates wrong prediction (false alarms). In (c) all the user annotated ground truth attributes are shown for each image example. If a prediction score on an attribute is lower than 0.5, is represented by red, which means miss detection. Otherwise it is blue.

It is interesting to investigate whether this deep learning framework can learn crowd features well. And the last baseline is the two-stream convolutional networks for action recognition [53]. It substitutes the proposed motion channels with optical flow maps (i.e. 2 maps for each frame, and 5 frames for each video) and keeps the appearance channels unchanged. According to the last row in Table 5.1, the proposed two-branch deep features DLSF+DLMF outperform all the baselines and STMP is the worst. Slow Fusion [52] does not outperform handcrafted features. This reason might be its way of inputting multiple frames to the deep model in order to capture motion information. It leads to a much larger net structure with many more parameters, and therefore requires larger scale training data. Similarly, the two-stream structure [53] also involves more parameters caused by ten motion channels, and optical flow itself cannot characterize common features across different scenes. Instead, the input of the proposed deep model is three motion channels, which well summarize motion information and reduce the network size. Summarizing all the resulting in Table 5.1, the conclusion is that the motion cue alone cannot get good result on crowd attribute recognition. By adding deeply learned motion features (DLMF) to deep learned static features (DLSF), the mean AUC has been improved by 1%. A detailed investigation shows the AUC of 41 attributes gets improved by adding DLMF. Most of these attributes belong to “Who” and “Why”. The averaged improvement of AUC is 5%.

Quantitative Evaluation. The AUC for each attribute with the proposed two-branch model (i.e. DLSF+DLMF) is shown in Fig. 5.3. Different colors represent “Where”, “Who”, and “Why” from left to right, and the results are sorted in a descending order.
5.3.4. Evaluation on Combined Deep Model

Figure 5.5: Six attributes predicted by DLSF, DLMF, and DLSF + DLMF. The first row shows negative examples with blue bar, and the second row shows positive examples with red bar. The prediction scores with DLSF, DLMF and DLSF + DLMF are represented by green, yellow, and orange.

The attribute “war” achieves the highest AUC score whereas “disaster” is the lowest. The lowest score may result from too few positive samples in the training set. Some attributes such as “battlefield”, “mob”, and “war” have strong correlation, although they belong to “where”, “who” and “why” respectively. They all have high AUC. It shows that the proposed deep model has learned the features that not only describe an attribute, but also measure the relationship within some attribute pairs or groups, and reveal the common pattern of these related attributes.

Qualitative Evaluation. Some good and bad examples on attribute prediction are shown in Fig. 5.4(a) and (b). Noted that the last example in Fig. 5.4(a) shows an attribute “stand” with a high prediction probability of 0.87, while the groundtruth recommends “sit”. It is actually quite challenging to determine the action as “stand” or “sit” in this example even from human perception. The third example shown in Fig. 5.4(b) has two of the top four attributes mistakenly predicted. The fourth is actually “stock market” but wrongly recognized as “conference center”. This is because people in this example move coherently, behaving like “audience” or “watching performance”, and its appearance cue cannot distinguish it as “conference center” or “stock market”. Fig. 5.4(c) shows some examples whose attributes are miss detected. The proposed deep model can well recognize attributes with distinctive motion and appearance patterns, such as the attributes related to the beach scene. But it may be poor for attributes owning complex and diverse appearance and motions, such as attributes related to the shopping mall scene.

Combined Deep Features vs. Separate Deep Features. To further verify that the proposed combined deep features outperform both DLSF and DLMF, 6 attributes with their quantitative (AUC scores) and qualitative results are shown in Fig. 5.5. The first row shows negative examples, and the higher prediction probability indicates
higher error. On the contrary, the higher prediction probability in the second row indicates higher accuracy. Generally, DLSF extracts static appearance features and thus works poorly at several attributes specified with motion patterns, e.g. “fashion show” and “walk”. But only motion features cannot effectively explore the difference between attributes with similar motion patterns. Likewise, the negative example in the fourth column is actually “skate”, but the given frame shows a short cut image that is similar to “mob” or “fight”. Combinational model fusing the appearance and motion channels and complement the missing cues in DLSF or DLMF, therefore reveals superior performances over all the sample attributes.

**Combined Deep Model vs. User Study.** Compared with the user study in the Section 4.3 of Chapter 4, it is interesting to see how human perception (when given different data types) is correlated with the results of the deep models. Fig. 5.6 shows the attributes which human and the proposed combined deep model recognize both well and both poorly. F-measure is used to compare the performance of the combined deep model and user study, since binary-labeled results from user study can only use F-measure to interpret as a weighted average of the precision and recall. Both the proposed combined deep model and the user study show that all the first three attributes in blue box achieve more than 0.7 while the middle three in green box only get around 0.4. The last two in red box provide attributes that are the most and the least ambiguous ones. Although the annotators labeled 35% of “photograph” as ambiguous, the proposed combined deep model can perform superiorly as 63%, whereas the annotator label results can only obtain 33%. As concluded from these results, in most attributes the proposed model correlates well with human perception. It is worth pointing out that for some ambiguous cases the model performs better.

### 5.3.5 Multi-task learning

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Multi-task</th>
<th>Single-task</th>
<th># wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where</td>
<td>0.89</td>
<td>0.84</td>
<td>22/27</td>
</tr>
<tr>
<td>Who</td>
<td>0.86</td>
<td>0.79</td>
<td>18/24</td>
</tr>
<tr>
<td>Why</td>
<td>0.86</td>
<td>0.79</td>
<td>36/43</td>
</tr>
<tr>
<td>Mean</td>
<td>0.87</td>
<td>0.81</td>
<td>76/94</td>
</tr>
</tbody>
</table>

**Table 5.2:** Compare average AUC with single-task learning and multi-task learning. The last column is the number of attributes where multi-task learning outperforms single-task learning.

Deep models are ideal for multi-task learning. The deep model discussed above is called multi-task learning, while the single-task learning is that training three different deep models for the three sets of attributes “where”, “who” and “why” separately.
Figure 5.6: Attributes with similar and disparate performance from user study and combined deep model. User study is the left score and the proposed combined deep model is the right score.

Since there exist correlations between different types of attributes, joint training of the three sets of attributes implicitly emphasizes the common features that shared by the correlated attributes. For instance, the “swimmer” should be at “beach” or “pedestrian” walks on “street”. Table 5.2 reports the average AUC of each set of attributes by single-task and multi-task learning in the first two columns. The last column shows the numbers of attributes where multi-task learning outperforms single-task learning. It is obvious that the multi-task learning improves the overall AUC from 0.81 to 0.87. The accuracies on most attributes get improved.

5.4 Summary

Both appearance features and motion features are learned by the proposed deep models. Instead of inputting multiple frames to deep models as existing works [52] did for video analysis, a set of motion channels is introduced, motivated by generic properties of crowd systems. Crowd features are learned with multi-task learning, such that the correlations among crowd attributes are well captured. The learned crowd features
and crowd attribute predictors have many potential applications in the future work, such as crowd video retrieval and crowd event detection.
Chapter 6

Deeply Discovering Volumetric Slices

Albeit video-oriented CNNs have achieved impressive performances on video related tasks, alternative video representations other than spatial-oriented inputs are still under-explored. In this chapter, a new multi-task learning deep model is proposed to separately learn appearance and motion features and effectively combine them. Instead of directly feeding multiple frames or hand-crafted feature maps to a deep model to learn motion features as most existing works did for video analysis, this chapter specially designs a new type of crowd motion channels as the input of the deep model. The motion channels are inspired by temporal slices that explicitly demonstrate the temporal evolutions of contents in crowd videos. In addition, extensive evaluations are conducted by multiple deep structures with various data fusion and weights sharing schemes to learn temporal features.

This chapter is structured as follows: the proposed crowd motion channels are explained in Section 6.1. Section 6.2 demonstrates the effectiveness of the proposed motion channels by designing a spatio-temporal deep neural network with different data-fusion and weight-sharing strategies, named as Slicing Volume (S-Vol). Results are reported and discussed in Section 6.3. Finally, conclusions are drawn in Section 6.4.

6.1 Spatial Temporal Crowd Features

This section introduces a novel type of input channels for CNNs, named as temporal-aware crowd motion channels to facilitate the learning of crowd-related motion patterns for crowd understanding. These channels explicitly convey meaningful spatio-temporal contents of the input videos without requiring a large scale of data, and tremendously improve the learning performance even using image-centric CNN architectures. These motion channels are capable of profiling the temporal information in arbitrary viewpoints, and revealing different temporal features in company with the spatial layout of the input video. With such motion channels, much richer temporal clues are retained in comparison with referred methods [52; 53], but without the need to significantly increase the scale of input channels. For instance, the average collectiveness map shown

\footnote{The collectiveness stated here is introduced in Section 3.2.1.}
Figure 6.1: Illustration of temporal-aware crowd motion channels. The (a) \(xy\)-slice, (b) \(xt\)-slice and (c) \(yt\)-slice are extracted from a video volume (d) and their corresponding collectiveness maps (hot color indicates high collectiveness). A collectiveness map by average pooling is presented in (e). Much more detailed and complete temporal information is preserved inside the \(xt\)- and \(yt\)-slices in comparison with the \(xy\)-slices as the sampled video frames or feature maps. The average collectiveness map suppresses the rich contents revealed in \(xt\)- and \(yt\)-slices and just keeps the average collectiveness responses along the \(xy\) plane.

in Fig. 6.1(e) fails to indicate the collectiveness evolved in the video “crosswalk”, while Fig. 6.1(b) and Fig. 6.1(c) explicitly show the variation of collectiveness (high – low – high) of the target video along the \(t\)-axis.

6.1.1 Crowd Motion Channels

RGB-T Motion Channels. As shown in Fig. 6.2, a color video can be considered as a 3-D RGB volume if its frames are stacked consecutively along the time dimension. Denote The video volume is denoted as \(V^{\text{rgb}}\) and the number of slices in each dimension is represented as \(N\). The \(xy\)-slices \(s_{xy}^{\text{rgb}}(t)\) are just the sampled video frames that slice \(V^{\text{rgb}}\) along the \(t\)-axis, as the snapshots marked by green in Fig. 6.2, where \(t \in T = \{t_1, \ldots, t_N\}\) is the selected slice indices. These slices have served as the traditional input channels to learn the appearance features and a small amount of motion patterns. Another two slices are proposed to capture richer temporal information. The \(xt\)-slices \(s_{xt}^{\text{rgb}}(y)\) are captured by sectioning \(V^{\text{rgb}}\) across its height, where \(y \in Y = \{y_1, \ldots, y_N\}\). And the \(yt\)-slices \(s_{yt}^{\text{rgb}}(x)\) are orthogonal to the width of \(V^{\text{rgb}}\), where \(x \in X = \{x_1, \ldots, x_N\}\). The indices \(x\) and \(y\) also refer to the probe lines. Fig. 6.2 provide examples of the \(xt\)- and \(yt\)-slices and they are marked by the red and blue boxes, respectively. Since they capture the entire temporal evolutions of crowd along the \(x\) and \(y\) dimensions in a single snapshot, more discriminative motion features are able to be discovered.

Optical Flow Motion Channels. The 3-D volume is not necessarily limited to the
6.1.2 Analysis of the Motion Channels

RGB or gray-scale visual data. In fact, volumes storing various per-pixel features are able to serve as the hosts to generate the desired motion channels for different learning tasks. An important and popular feature that frequently adopted in the field of action recognition and video classification is the optical flow [140; 161; 162], which describes the per-pixel displacement between two adjacent frames. The motion contents recorded in the optical flow volume act as a complement to RGB volume by providing explicit representations of the motion in videos so that the network does not need to estimate motion implicitly during the learning procedure. This kind of motion representation is explored by slicing the optical flow volume \( V_{\text{flow}} \) which is similar to the way introduced for RGB volume \( V_{\text{rgb}} \). The resultant motion channels are denoted as \( s_{xy}^{\text{flow}}(t) \), \( s_{xt}^{\text{flow}}(y) \) and \( s_{yt}^{\text{flow}}(x) \) for \( xy \)-, \( xt \)- and \( yt \)-slices, respectively. Fig. 6.3 illustrates the video slices and the corresponding optical flow slices of a crowd video. The optical flow based \( xt \)- and \( yt \)-slices tend to expose the long-range crowd motion flows both spatially and temporally, and directly indicate the motion patterns that the crowd conveyed.

**Figure 6.2:** Examples of temporal-aware motion channels extracted from a crowd video volume (a). The (b) \( xy \)-slices, (c) \( xt \)-slices, and (d) \( yt \)-slices are marked by green, red, and blue, respectively. The horizontal direction in each image of (c) and (d) indicates the \( t \)-axis.

6.1.2 Analysis of the Motion Channels

The \( xy \)-slices are nothing special but a series of sampled frames (or feature maps). They are more like to offer spatial patterns rather than the motion ones, due to 1) the number of slices is limited, 2) the CNNs are not originally designed to handle temporal data. Actually, it is hard to recognize the activities or behaviors of a crowd video even via human eyes, if only inadequate, ambiguous and chaotic appearance information is provided. Take the \( xy \)-slices shown in Fig. 6.3(a) as an example, it is hard to detect or observe the temporal evolution of the crowd crosswalk from a single spatial slice.
Figure 6.3: Temporal-aware motion channels from a crowd video volume (a) and its optical flow volume (b). These slices are extracted at the same positions from both volumes. Although they are constructed from different features, they share similar spatial and temporal patterns.

However, surprisingly, the channels in $xt$ and $yt$ planes show prominent expressive ability.

Commons of $xt$- and $yt$-slices. Recently the study by Zhou et al. [80] have demonstrated that collectiveness is a common property of crowd systems. These studies often compute the collectiveness from each single frame (i.e. $xy$-slice), and average or concatenate the collectiveness along time domain. This section shows that both $xt$- and $yt$-slices encompass richer temporal information and can even convey the temporal evolution from just a single snapshot. For example, in comparison with the spatial feature map ($xy$-slice) of crosswalk in Fig. 6.4(b), one single $xt$- and $yt$-slice can indicate the temporal evolution of overall collectiveness. As shown in the top of Fig 6.4(c) and (d), the collectiveness of the crosswalk crowd has a evolution pattern of high-low-high along $t$-dimension, and the black boxes mark the time-interval with low collectiveness. Thus, they faithfully reveal both the temporal and spatial positions that two group of pedestrians are crossing each other, in which case the collectiveness should be lower.

Another example is shown in the bottom of Fig. 6.4. A crowd with marching motion is naturally of high collectiveness, but its $xy$-slice shown in Fig 6.4(b) output a sparse collectiveness distribution which cannot well express the motion pattern of marching. Fortunately, the selected $xt$- and $yt$-slices present a more stable and uniform collectiveness distribution along the $t$-axis, they effectively depict that the crowds are marching collectively and stably in a long duration. Therefore, these $xt$- and $yt$-based motion channels can robustly monitor the crowd dynamics with different user-designed features (RGB, optical flow, collectiveness and etc.) and provide novel interpretations about crowd motions.

Differences between $xt$- and $yt$-slices. The $xy$-stack is named as a collection of several $xy$-slices, which are constructed from sectioning slices in the same dimension
6.1.2. Analysis of the Motion Channels

(a) (b) (c) (d)

Figure 6.4: Temporal-aware motion channels extracted from the collectiveness volume. (a) Two crowd videos with different levels of collectiveness. (b)–(d) The $xy$, $xt$- and $yt$-slices. The video shown in the upper row is about “crosswalk” and the one in the lower row contains crowd “marching”. The different motion patterns between two videos can be easily depicted by examining the temporal evolutions of their collectiveness from $xt$- and $yt$-slices. Hot color represents high collectiveness.

but different probe locations. Similarly, $xt$- and $yt$-stacks refer to a collection of $xt$- and $yt$-slices, respectively. Generally, the $xy$-slices have low variances, which can be seen from the first row of Fig. 6.2. However, unlike $xy$-slices, there are high intra-variances among $xt$-slices within the same $xt$-stack. As shown in Fig. 6.2, the $xt$-stack (in red) contains four $xt$-slices that are extracted from four different locations along $y$-dimension, recording the motions of trees, the heads of pedestrians, their legs, and the grass, respectively. It is interesting to observe that the middle two slices, revealing the marching motion, behave like trajectories. Thus the patterns in the $xt$-slices always depict the long-term crowd motions. Compared to $xt$-slices, on the contrary, the intra-variances among $yt$-slices are much lower, as shown in the third row of Fig. 6.2.

It is commonly observed that the patterns in a $yt$-slice are more often referred to the duration that the pedestrians in the crowd cross the corresponding vertical probe lines and the moments that these pedestrians stand around it. This is because the videos are often captured from the viewpoint parallel to the ground (i.e. $x$-axis), and human in the crowd videos are always positioned straight and orthogonal to the ground. Hence, even though off-the-shelf CNN models have already been sufficient to handle the proposed
motion channels, the learning architectures should make reliable adjustments and modifications specific to the characteristics of different motion channels so as to unleash the discriminative potential of the proposed motion channels to their extent.

6.2 Spatial Temporal Deep Architecture

This section extends a fully-convolutional neural network (FCNN) structure to learn the spatio-temporal crowd features and employ it to the application of crowd attribute prediction. There are three benefits owned by FCNN that are particularly desirable for learning features from crowd videos: 1) a FCNN does not require the input of specific sizes, which can be directly applied to scenes with different sizes without resizing or cropping; 2) a FCNN can capture local information within the receptive fields and help to learn local crowd motion features; and 3) global crowd motion properties can be easily obtained from the feature maps by global summation or average pooling operations. Considering the discussed properties of motion channels (Sec. 6.1.2), the spatial temporal FCNN architecture requires a specific slice fusion and weight sharing scheme to effectively learn and combine the intermediate features during the learning procedure.
6.2.1 Network Configuration

The layer configuration of the proposed spatial temporal deep model (i.e. S-Vol) is schematically shown in Fig. 6.5(a). A simple annotation is employed to indicate the layer parameters: 1) \( \text{Conv}(N,K,S) \) indicates the convolutional layer with \( N \) outputs and a kernel of size \( K \) and the stride of size \( S \); 2) \( \text{Pool}(T,K,S) \) denotes the pooling layer with type \( T \), a kernel of size \( K \) (\( K \) can be \text{global} and it means a global pooling resulting in a single value) and the stride of size \( S \); 3) \( \text{FC}(N) \) represents a fully connected layer of \( N \) neurons; 4) \( \text{ReLU} \) and \( \text{Sig} \) represent the rectified linear unit and the sigmoid function respectively. The network structure can be summarized as \( \text{Conv}(80,7,1) - \text{ReLU} - \text{Pool}(\text{MAX},2,2) - \text{Conv}(160,7,1) - \text{ReLU} - \text{Pool}(\text{MAX},2,2) - \text{Conv}(320,3,1) - \text{ReLU} - \text{Conv}(320,3,1) - \text{ReLU} - \text{Conv}(320,3,1) - \text{ReLU} - \text{Pool}(\text{SUM},\text{global},1) - \text{FC}(1024) - \text{FC}(N_{\text{attr}}). \ N_{\text{attr}} \) is the number of attributes designed in the crowd attribute prediction. All convolution operations are properly padded to keep the same shape. And cross entropy is adopted here for each attribute as the loss function so as to fulfill the multi-task learning.

6.2.2 Slice Fusion

The sampled \( xy \)-slices of a video is piled into a \( xy \)-stack as the input channels for the deep architecture, similarly with the \( xt \)- and \( yt \)-stacks. To verify the effectiveness of the proposed input channels (i.e. \( xy \)-, \( xt \)- and \( yt \)-slices), and explore the intra-stack relationships between slices as well as the inter-stack relationships among three stacks, this section designs the following S-Vols with different types of slice fusion (motivated by [52]) and weight sharing strategies, as shown in Fig. 6.5(b).

1) Early fusion. The early fusion scheme combines information of all slices in a stack immediately at the first convolutional layer with separate weights. This early and direct connectivity is effective to capture local correlations among the slices. The separate weights capture low-level changes across the slices in a stack.

2) Late fusion. The late fusion scheme places a number of branches to extract features from each slice and then merges the responses from all branches into a fully connected layer \( \text{FC}(1024) \). Branches with both separate-weight and shared-weight schemes are investigated to explore the intra-stack relationships among slices.

▷ separate-weight scheme: separate-weight scheme treats input slices differently and captures different high-level patterns across the slices. This scheme implies translation variance along the sliced dimension.

▷ shared-weight scheme: branches with shared weights extract the same kind of features from the input slices and thus capture translation-invariant patterns along
### 6.3 Experimental Results and Discussions

#### 6.3.1 Settings

**Dataset and evaluation.** The proposed temporal-aware crowd motion channels and deep architectures are applied to crowd attribute recognition as stated in Chapter 4, which predicts the possible attributes related to crowd behaviors inside a test crowd video. The quality metric Area Under ROC Curve (AUC) is still applied to evaluate

<table>
<thead>
<tr>
<th>S-Vol</th>
<th>Motion</th>
<th>Stack-combined model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>optic-xy-stack</td>
<td>optic-xt-stack</td>
</tr>
<tr>
<td>Early fusion</td>
<td>0.76</td>
<td>0.79</td>
</tr>
<tr>
<td>Late fusion (separate-weight)</td>
<td>0.70</td>
<td>0.79</td>
</tr>
<tr>
<td>Late fusion (shared-weight)</td>
<td>0.72</td>
<td>0.76</td>
</tr>
<tr>
<td>Slow fusion</td>
<td>0.74</td>
<td>0.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S-Vol</th>
<th>Static+Motion</th>
<th>Stack-combined model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RGB-xy-stack</td>
<td>RGB-xt-stack</td>
</tr>
<tr>
<td>Early fusion</td>
<td>0.80</td>
<td>0.78</td>
</tr>
<tr>
<td>Late fusion (separate-weight)</td>
<td>0.78</td>
<td>0.80</td>
</tr>
<tr>
<td>Late fusion (shared-weight)</td>
<td>0.77</td>
<td>0.78</td>
</tr>
<tr>
<td>Slow fusion</td>
<td>0.79</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 6.1: Crowd attribute recognition results (AUC score) on the WWW Crowd Dataset with different temporal-aware input channels and fusion strategies. The best results for each kind of input are in bold. The results for the stack-combined models are shown on the right.

3) **Slow fusion.** The slow fusion model merges information from input slices in multiple steps. The shared-weight scheme detects correlations among slices within a receptive field and gradually increases the receptive fields through multi-step merging. The nearby slices firstly are fused together to capture local information, and then progressively merged into fewer branches such that the higher layers get access to more global information. As shown in Fig. 6.5(b), the first convolutional layer is applied to every 4 slices with stride 2 to produce 4 response maps. In the same way, the second and third convolutional layers perform on every 2 branches. Thus the third convolutional layer carries information from all slices and is then connected to upper convolutional layers.
Data augmentation. The original size of crowd videos in WWW dataset is $360 \times 640 \times \tau$, where $\tau$ is the number of frames in each video. In order to involve more temporal information and make the spatial and temporal dimensions balanced, the number of frames $\tau$ in each video is fixed to 200. If $\tau < 200$, empty frames are padded by duplicating a batch of available frames to their locations. Thus the size of the video volume used in the following experiments is fixed to $360 \times 640 \times 200$. Five sub-volumes are spatially sampled at four corners and the center from each video volume. The size of the sub-volumes is fixed to $256 \times 256 \times 200$.

Training, validation and test data set. Following the same data setting in the Section 5.3.1 of Chapter 5, the WWW dataset is divided into training, validation, and test sets with a ratio of $7 : 1 : 2$. Different from Chapter 5, due to data augmentation, the total number of videos is $5 \times 10,000 = 50,000$. It is guaranteed that the five augmented sub-volumes from one volume are distributed to the same set.

6.3.2 Evaluation on Different Input Channels

In the following experiments, the input channels are denoted as $xy$-, $xt$-, and $yt$-stacks, including 10 $xy$-, $xt$-, and $yt$-slices, respectively. The experiments are constructed by inputting 6 kinds of temporal-aware motion channels, i.e., RGB-$xy$-, RGB-$xt$-, RGB-$yt$-stacks and optic-$xy$-, optic-$xt$-, optic-$yt$-stacks, into deep models with 4 different fusion schemes as introduced in Sec. 6.2. All the results are shown in Table 6.1, which will be discussed as follows.

1. $xy$-stacks. As shown in Table 6.1, both optic-$xy$- and RGB-$xy$-stacks obtain the best performances with the early fusion model. In addition, the optical flow stacks always have lower performances against those for RGB stacks since the textures provided by RGB frames are more informative than the displacement fields in the optical flow maps. The worst model in general employs late fusion with separate weights scheme, which fuses the intermediate features that separately learned in each slice after the last convolutional layer, and it may result in capturing too much local spatial patterns but failing to interpret the temporal connections between slices. The poor performance is also partially caused by the loose temporal connections between $xy$-slices, which are sparsely sampled from a video (i.e. sample one slice every 20 frames). The upper-layer fusion fails to extract their temporal patterns given the weak inputs. Grouping all the channels or partial channels at the first layer (i.e. early fusion and slow fusion) is a good choice for $xy$-slices to learn spatio-temporal patterns.
2. *xy-stacks versus xt- and yt-stacks.* By introducing the xt- and yt-stacks into these learnings models, their performances are observed to be superior to those by xy-stacks, which proves the effectiveness of the proposed motion channels in learning the spatio-temporal patterns. Although the results by RGB-xy-stacks significantly outperform those by optic-xy-stacks in all the deep models, interestingly this gap becomes tremendously narrower when the comparisons are conducted on these two kinds of yt- or xt-stacks. They even share comparable results under their best settings. On one hand, it reveals that the crowd attribute recognition relies more on the temporal information than the spatial or appearance patterns. On the other hand, it also means that the learning potential for the spatio-temporal patterns of the input videos is effectively reached with the special structures contained in xt- and yt-slices. Therefore, the lack of texture information conveyed in optical flow-based motion channels does not result in noticeable performance degradations in the application of crowd attribute recognition.

3. *xt-stacks versus yt-stacks.* Different from the conclusion drawn from xy-slices, yt-slices perform worst by early fusion but best by the late fusion with shared weight scheme. The reason might be the high correlations between yt-slices in one yt-stack. In this case, similar slices will output similar feature responses after the last convolutional layer if the shared weight scheme is adopted. By fusing the feature responses of a video together as late fusion suggests, the temporal patterns are thus presented implicitly inside the gradually varying feature patterns. On the contrary, xt-slices receive the best results if learned by the separate-weight strategy. Due to high variance among slices inside the xt-stack, it suggests that it is better to learn the weights separately to extract the patterns for each slice, which is similar to the case of xy-slice.

Fig. 6.6 compares the feature responses of optic-xt-slices and optic-yt-slices calculated before the first fully connected layer. The deep models adopt the late fusion with shared- and separate-weight strategies. The xt-slices are different in visualization but yt-slices are generally share similar patterns. The feature responses by shared-weight strategy shown in Fig. 6.6(c) present similar patterns among all the ten slices, which makes sense to yt-slices but fails to capture the intra-variances among xt-slices. But in contrast, the results by separate-weight strategy in Fig. 6.6(d) show distinct responses for each slice, no matter whether they are similar. The correlations between yt-slices are thus lost so that fewer temporal patterns can be extracted in comparison with that by shared-weight strategy.

4. *Stack-Combined model.* Three types of stacks (i.e. xy-, xt-, and yt-stacks) are combined as the input of a stack-combined deep model, named as xyt-stack. It
6.3.3 Comparison with State-of-the-Art Methods

1) Static model evaluation. Due to the indispensability of the static features, the static patterns are learned by inputting a single frame (SF) of a video into the same

Figure 6.6: Comparison of the feature responses of optic-$xt$-slices and optic-$yt$-slices before the first fully connected layer and learned by late fusion with shared- and separate-weight strategies. (a) the input video snapshot and its corresponding optical flow displacement field. (b) ten selected optic-$xt$-slices (upper) and optic-$yt$-slices (lower). (c) feature responses learned by shared-weight strategy. (d) feature responses learned by separate-weight strategy. The feature responses shown in upper column come from optic-$xt$-slices, and those in lower column are from optic-$yt$-slices. The feature response of a single slice is of dimension 32 and the responses are normalized into [0, 1].

employs a late fusion scheme that fuses three separate models that are optimal for $xy$-, $xt$- and $yt$-stacks. As shown in Table 6.1, the performance of the slice-combined model for RGB or optical flow only improves 1% in comparison with those of single tracks. It might attribute to the correlation among three stacks should be learned from lower layers, otherwise their local spatio-temporal patterns cannot be well extracted.
Table 6.2: Compare with baselines. Best are in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Static (no pretrain)</th>
<th>Motion</th>
<th>Static (no pretrain) + Motion</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baselines-HC</td>
<td>SFH</td>
<td>0.68</td>
<td>MDH</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DenseTrack [140]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SFH + DenseTrack</td>
<td>0.63</td>
</tr>
<tr>
<td>Baselines-DEEP</td>
<td>DLSF (no pre-train)</td>
<td>0.74</td>
<td>DLMF [Chp. 5]</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>[Chp. 5]</td>
<td></td>
<td>DLSF + DLMF (no pre-train)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Slow Fusion [52]</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Two-stream [53]</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3D-CNN [51]</td>
<td>0.78</td>
</tr>
<tr>
<td>Our Methods</td>
<td>S-Vol + SF</td>
<td>0.81</td>
<td>S-Vol + optic-xyt-stack</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>(no pre-train)</td>
<td></td>
<td>S-Vol + SF + Optic-xyt-stack (no pre-train)</td>
<td>0.88</td>
</tr>
</tbody>
</table>

In addition, the same baseline in the Section 5.3.2 of Chapter 5 are adopted to be the comparison method (i.e. the bag-of-words [151] of Dense SIFT [157], GIST [40], HOG [158], self-similarity [159], and local binary patterns [160]). As shown in Table 6.2, the proposed method outperforms the handcrafted static features with a large margin.

2) **Motion model evaluation.** In this part, the performance of the proposed temporal-aware crowd motion channels is evaluated (i.e. optic-xt and yt-stacks) in comparison with two baselines: the motion branch model (i.e. Deeply Learned Motion Features (DLMF)) presented in the last Chapter 5 and the proposed optic-xyt-stack, which is analogous to optical flow stacking in [53]. The optical flow-based motion channels in this evaluation is applied here since they represent the pure motion information.

**Quantitative results.** The mean AUC of DLMF is 0.68 [Chp. 5] that is much lower than that of stack-combined motion model optic-xyt-stack (0.81), and even worse than single branch models (i.e. optic-xy-, optic-xt- and optic-yt-stack) as shown in Table 6.1. It demonstrates the proposed temporal-aware motion channels are more effective to extract the temporal patterns than the baseline. Last but not the least, by taking optic-xyt-stack as the input of the motion branch of the deep architecture proposed in
6.3.3. Comparison with State-of-the-Art Methods

Figure 6.7: Attribute prediction examples by optical flow-based temporal-aware motion channels. For each video, the ground-truth attributes are depicted next to it. The length of bar from the left origin to its end indicates the prediction probability of an attribute either by \( xy \)-, \( xt \)- or \( yt \)-slices (marked by red, blue and purple) as well as the preceded method DLMF in Chapter 5] (marked by green). The rightmost two examples marked by a black box are the failure cases.

the Section 5.1 of Chapter 5, the AUC is 0.77, which also indicates the spatial temporal deep models contain a better interpretation for the proposed motion channels.

This section also compares the proposed method with another two baselines with handcrafted features. One is the histogram of motion descriptor (MDH) introduced and also compared in the Section 5.3.3 of Chapter 5. And another is dense trajectory [140] showed state-of-the-art result in action recognition. Both baseline features are trained with independent classifiers via linear SVM similar to the SFH baseline.

According to the results shown in the fourth column of Table 6.2, S-Vol + optic-\( xyt \)-stack outperforms the other two baselines by 23% and 18% on mean AUC respectively.

**Qualitative evaluation.** The section also conducts the qualitative evaluation for a list of exemplar crowd videos shown in Fig. 6.7. The proposed motion channels can successfully predict the attributes that are characterized by motion, such as “run”, “runner”, “walk” and “parade”. Interestingly, the \( xt \)- and \( yt \)-slices can well predict attributes with complex and long-term motion patterns, like “model” and “runaway”, “marathon” and “rink”, as well as “fight” and “mob”. However, some attributes cannot be well predicted by motion channels, even though they can be easily recognized by appearance clues. For instance, two crowd videos inside the black box in Fig. 6.7 contain several attributes such as “beach”, “soldier”, which have extremely low prediction probabilities by all the motion channels. Some other failure cases result from various issues, from the optical flow errors to complex camera motions, as well as the scale of pedestrians in crowd. In addition, the reference DLMF generally performs poorer than the proposed methods.

3) **Static model vs. Motion model.**
Figure 6.8: Attribute prediction examples by optical flow-based tempoal-aware motion channels and by the appearance channels separately. Each example has the positive attribute results by optic-\textit{xyt} shown on the left and by the appearance (RGB) shown on the right. The groundtruth attributes are listed under each examples. The heights represent prediction scores. Orange indicates correctly predicted attributes, and blue indicates wrong prediction (false alarm).

**Quantitative evaluation.** As reported in the Chapter 5 that the motion model (DLMF) performs worse than static model (DLSF) no matter with or without pre-training on ImageNet. The performance drop is at least 6%. However, the proposed motion model optic-\textit{xyt}-stack has a comparable result with static model.

**Qualitative evaluation.** Some attributes can be more easily recognized by appearance information. As examples shown in black box in Fig. 6.7, some failure attributes like “beach”, “soldier” and “newly-wed couple” are easily recognized by appearance information. Thus appearance features are essential complements of the motion features to achieve satisfactory attribute recognition performance. In Fig. 6.8, the left example has a good prediction result via optic-\textit{xyt} motion features and its appearance clue from single frame may cause ambiguities. On the contrary, the appearance features outperform motion features in the right example. It is much easier to recognize “newly-wed couple” and “runaway” by appearance but it requires motion features to identify “fight” and “mob”. It also provides a hint of why RGB-stacks generally have a better performance than optic-stacks as RGB-stacks contain more appearance features.

4) **Static + Motion model evaluation.** The combined model is evaluated by fusing the static and motion features together, and compare with three handcrafted baselines and another four methods by deep learning. The handcrafted baselines are the same to those adopted in Chapter 5, i.e. the combination of the static feature (SFH) and the motion features (MDH), the combination of the SFH and dense trajectory, and the spatio-temporal motion patterns (STMP). Among four deep learning approaches, besides the Two-stream and Slow-Fusion models, another two baselines include the preceded two-branch DLSF+DLMF model introduced in Chapter 5 and a 3D-CNN [51] that has been successful in action recognition by involving 3D convolutions to simultaneously learn static and motion features.
According to the last row in Table 6.2, the proposed combined model outperforms all the baselines while STMP is the worst. Slow Fusion [52] inputs multiple frames to the deep model which might lead to a larger net structure with many more parameters, and also cannot well learn temporal dynamic information. Meanwhile, optical flow itself in the two-stream structure [53] cannot characterize common features across different scenes. 3D-CNN requires very large memory to capture long-range dynamics. Taking 200 frames as input, its model size is 243 Gb, while ours is 11 Gb. With 10 frames and 32 kernels its AUC is 0.78, which is 10% lower than ours. In the last row of Table 6.1, without pre-training, the proposed combined model (i.e. S-Vol+SF+optic-xyt-stack) outputs the mean AUC of 0.88 which is much higher than DLSF+DLMF (no pre-train). While after pre-training on ImageNet, the proposed combined model obtains the AUC of 0.89 which outperforms that of DLSF+DLMF (pre-train) [Chp. 5] (i.e. 0.88). All these results demonstrate the effectiveness of the proposed method. Moreover, in comparison with the static model (S-Vol+SF), the proposed combined model improves 7% thanks to the fusion of motion features, it also proves the robustness of the proposed motion representation. Furthermore, RGB-xy-stack assembles multi-frames as [52] that also represents both the static and motion features. Its AUC is 0.81, and the proposed combined model improves it by 7%. From Table 6.1, even with only motion features (e.g. optic-xyt-stack), the proposed method can get a comparable performance with the combined model of RGB-xy-stack or RGB-xyt-stack.

6.4 Summary

In this chapter, a set of temporal-aware motion features is designed by slicing the crowd video volumes from different dimensions. Multiple deep networks with different data fusion strategies and weights sharing schemes are also proposed to deeply learn temporal-aware features. The application of crowd attribute recognition demonstrates that the proposed motion features preserve rich temporal information for crowded scene understanding, and different deep structures perform distinctively for different types of motion features. These temporal-aware slices have a potential to be generally operated on many other volumes to capture different clues from xy, xt, and yt. Besides, it is observed that when crowds move horizontally, the yt-slices will perform similar patterns as xy-slices, while the yt-slices look quite different from xy-slices when crowds move vertically or diagonally. It is an interesting future work on exploring slicing strategies, which in this chapter is just horizontal and vertical slicing.
Chapter 7

Slicing Convolutional Neural Network

The preceding Chapter shows that besides representing a video volume as a stack of spatial \(xy\)-slices cut along the dimension \(t\), 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 the video-tasks. However, the approach in Chapter 6 extracts 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. 7.1. Moreover, the internal properties and connections among different slices were not well learned but just handled independently.

To overcome the limitations listed above, this chapter proposes a Slicing CNN (S-CNN) model with three branches by extracting 3D feature cuboids into 2D spatio- and 2D temporal-slices. Compared slicing the raw video volume, the feature cuboid generated by spatial filter describes the motion of a particular semantic unit. An example is shown in Fig. 7.2, 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. With innovative model design, appearance and dynamic informations can be effectively learned from semantic levels, separately and interactively. In addition,

![Figure 7.1: 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.](image-url)
the proposed model is capable of extracting appearance and dynamic informations from long-range videos (i.e. 100 frames) without sampling or compression.

The structure of this chapter is designed as follows: Section 7.1 profiles the unique advantages of the feature maps at semantic-level. The model implementation details can be found in Section 7.2. The complete S-CNN model is shown in Fig. 7.7, and the detailed architecture of the single branch (i.e. S-CNN-xy, S-CNN-xt, and S-CNN-yt) is shown in Fig. 7.6. Section 7.3 provides the ablation study of S-CNN and the comparison results with state-of-the-arts. The conclusion of this work is presented in Section 7.4.

### 7.1 Semantic Feature Maps Profiling

Existing deep learning frameworks often use motion maps as the input of 2D CNN to learn dynamic features. However, much information has been lost at the input layer since they compress the entire temporal range by sub-sampling frames or averaging spatial feature maps along time dimension. Alternatively, dynamic feature representations can also be described from 2D temporal slices that cut across 3D video volume from another two orthogonal planes, as xt- or yt-slices shown in Fig. 7.3. When different types of subjects cross the line at different speed and in different directions, various visual patterns are presented in the xt- and yt-slice. Given multiple temporal slices extracted from different horizontal and vertical probe lines and fed to multiple 2D CNNs, their dynamic features can be combined to capture long-range spatio-temporal dynamics.

It is a general case that a xt- or yt-slice captured from 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. On the other hand, it’s good to see the proposed model can effectively include crowd-related dynamic patterns while reject motions from background irrelevant contents. It is thus essential to 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. It also expects the dynamic representations in one temporal slice to be the concatenation of dynamic representations for all the available semantic patterns.

7.1.1 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 [163]. 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
7.1.2 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. 7.5, 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? These questions can be answered by pruning spatial filters that generate “irrelevant” feature maps and investigate its impact to the attribute recognition performance.

**Figure 7.4:** 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.

Specific object. From the example shown in Fig. 7.4, convolutional layers of the VGG model [122] 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. 7.5(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.

Motivated by the aforementioned observations, exploiting such feature cuboids is capable 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. Therefore, it is desirable to extract the 3D feature cuboids of spatial filters that responds to semantic patterns, as the input of xt- and yt-slice filters to extract dynamic features of different semantic units.
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 [24], denoted as $S_i$ for a query image $i \in \mathcal{I}$, which is then resized to match the resolution of the extracted feature maps. Two scores (i.e. affinity score and conspicuous score) are adopted to measure the “relevance”.

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

$$\alpha^n_i = \|1_{[F^n_i > 0]} \cdot S_i \|_1 / \|S_i \|_1,$$

where $1_{[\cdot]}$ is an indicator function that returns 1 when its input argument is true. $\cdot$ denotes the element-wise multiplication.

**Conspicuous score.** The conspicuous score $\kappa^n_i$ calculates the feature’s energy inside the crowd foreground annotated in the mask $S_i$ against its overall energy,

$$\kappa^n_i = \|F^n_i \cdot S_i \|_1 / \|F^n_i \|_1.$$

A histogram $H$ with respect to the filters is then constructed in a certain layer. For filter $\# n$, if the feature map $F^n_i$ satisfies either $\alpha^n_i > \tau_\alpha$ or $\kappa^n_i > \tau_\kappa$, given two thresholds $\tau_\alpha$ and $\tau_\kappa$, the value of its histogram bin will be as

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

By sorting $H(n)$ in a descending order, the first $r$ spatial filters are retained but the left are pruned. The reserved filters are denoted as $N_r$.

### 7.1.3 Semantic Temporal Slices

Existing studies typically learn dynamic features from raw video volumes [52] or hand-crafted motion maps [53; 54]. However, much information is lost at the input layer since they compress the entire temporal range by sub-sampling 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. 7.1. 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. 7.1 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, this section proposes Semantic Temporal Slice (STS) extracted from semantic feature cuboids, which are obtained from the $xy$ convolutional layers, as shown in Fig. 7.7. 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.
7.2 S-CNN Deep Architecture

This section proposes a new end-to-end model named as Slicing CNN (S-CNN) consisting of three branches (i.e. S-CNN-xy, S-CNN-xt, S-CNN-yt) which benefits from the spatio-temporal structures and the semantic meanings of the CNN feature cuboids. Firstly, a collection of semantic feature cuboids are obtained after leaning appearance features by a 2D CNN model on each frame of the input video volume. Each feature cuboid captures a distinct visual pattern, or an object instance/category. Based on the extracted feature cuboids, three different 2D spatio- and temporal-filters (i.e. xy-, xt-, and yt-) are introduced 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.

7.2.1 Single Branch of S-CNN Model

The 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. The VGG architecture [122] is employed here because of its excellent performance in image-related tasks. As shown in Fig. 7.6, for an input raw video volume, following the original settings, the lower layers of VGG-16 from conv1_1
### 7.2.1. Single Branch of S-CNN Model

#### Figure 7.7: 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 adapt 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.

To conv4-3\(^1\) are used to extract spatial semantic feature maps. The size of the feature cuboid \(F_i^s\) 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 \(\tau\).

**S-CNN-xy branch.** The S-CNN-xy branch learns spatio-temporal features from the xy-plane by xy-convolational filters. Based on the spatial feature cuboids \(\{F_i^s\}_{i=1}^{\tau}\), the S-CNN-xy branch continues 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 \(\tau \times c \times h_t \times w_t\). In other words, there are \(c\) xy spatio-temporal feature cuboids.

---

\(^1\)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.
each of which is $\tau \times h_t \times w_t$. A $1 \times 1$ filter is then adopted on each $F_{xy}^{i}$ to fuse the temporal information from different frames. The spatio-temporal feature maps $F_{xy}^{t}$ 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 $\{F_{s}^{i}\}_{i=1}^{7}$ by xt- or yt- branch, the S-CNN-xt/yt branch first swaps dimensions of the original $xy$-plane to the corresponding xt- or yt-plane. Take xt-branch as an example, as shown in Fig. 7.6, the semantic feature cuboids turn to be $h_{s} \times c \times \tau \times w_{s}$ after swapping dimensions. The $xy$-convolutional filters used in $xy$-branch are then substituted with xt-filters for conv5_1 to conv5_3 layers. Before temporal pooling at the last stage, the dimensions of xt-plane need to be swapped back 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.

### 7.2.2 Combined S-CNN Model

After training each branch separately, the features learned from different spatial and temporal dimensions are then fused together by concatenating the spatio-temporal feature maps (i.e.$F_{xy}^{t}$, $F_{xt}^{y}$, and $F_{xy}^{t}$) from three branches with $\ell_{1}$ normalization. Linear SVM is adopted as the classifier for the sake of its efficiency and effectiveness on high-dimensional feature representations. A SVM is trained independently for each attribute, thus there are 94 models in total. To train each SVM, the videos containing the target attribute are regarded as the positive samples and the rest are the negative samples. The complete S-CNN model is visualized in Fig. 7.7.

### 7.3 Experiments

#### 7.3.1 Settings

**Dataset.** To demonstrate the effectiveness of the proposed S-CNN deep model, this section investigates it on the task of crowd attribute recognition with the WWW Crowd Dataset introduced in Chapter 4, 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). The experiment setup follows the original one in Chapter 5 and guarantees that the attributes are learned scene-independently.

**Evaluation Metrics.** To well evaluate the proposed methods, both Area Under ROC Curve (AUC) and Average Precision (AP) are adopted as the evaluation metrics. 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. 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 \( \text{mAP}_{lb} = \frac{1}{N_{\text{attr}}} \sum_{k=1}^{N_{\text{attr}}} |T_k| / |T| \), where \( N_{\text{attr}} \) is the number of attributes, \( T \) is the test set, \( T_k \) is the set of samples with the attribute indexed by \( k \). In the experiments, the theoretical lower bound is 0.067.

**Model Pre-training.** As a common practice in most deep learning frameworks for visual tasks, the proposed S-CNN models are initialized with the parameters pre-trained on ImageNet. This is necessary since VGG requires diverse 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, the following experiments 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. The first two fc layers with 4096 neurons followed by Rectified Linear Units (ReLUs) and Dropout are kept while the last fc layer with 94 dimensions (attributes) followed by a cross-entropy loss function. If no specific clarifications are stated, this strategy is applied to initialize all experimental models.

### 7.3.2 Ablation Study of S-CNN

This section compares the recognition performance raised from different types of features learned by the proposed S-CNN models, and find that the semantic temporal slices can effectively learn the long-range crowd motion patterns that are essential to recognize the crowd attributes.

**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, the recognition performance is compared between 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 \( \tau \) has three ranges: 20, 50, and 100 frames, denoted as S/(R)TS\( _{\tau} \). The hardware limitations of current implementation cannot afford RTS\( _{100} \) with full spatial information.

- **Low-level v.s. Semantic-level Temporal Slices.** In comparison with the
Table 7.1: Results of S-CNNs learned from raw- and semantic-level with different temporal ranges (bolds are the best).

<table>
<thead>
<tr>
<th>methods</th>
<th>(\tau)</th>
<th>mean AUC</th>
<th>mean AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS-(xy)</td>
<td>20</td>
<td>91.06</td>
<td>49.56</td>
</tr>
<tr>
<td>RTS-(xy)</td>
<td>50</td>
<td>92.28</td>
<td>52.05</td>
</tr>
<tr>
<td>RTS-(xy)</td>
<td>100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>STS-(xy)</td>
<td>20</td>
<td>91.76</td>
<td>54.97</td>
</tr>
<tr>
<td>STS-(xy)</td>
<td>50</td>
<td>92.39</td>
<td>55.31</td>
</tr>
<tr>
<td>STS-(xy)</td>
<td>100</td>
<td>92.52</td>
<td>55.67</td>
</tr>
</tbody>
</table>

Table 7.2: Results of S-CNNs learned from semantic-level with short- and long-range temporal slices. Results in bold are the best.

<table>
<thead>
<tr>
<th>methods</th>
<th>(\tau)</th>
<th>mean AUC</th>
<th>mean AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>STS</td>
<td>20</td>
<td>91.76</td>
<td>54.97</td>
</tr>
<tr>
<td>STS</td>
<td>100</td>
<td>92.52</td>
<td>55.67</td>
</tr>
</tbody>
</table>

results by RTS\(_{\tau}\), STS\(_{\tau}\) (\(\tau = 20, 50\)) is superior especially in mAP scores, as shown in Table 7.1. The results of \(xt/yt\) – semantic slices in Table 7.2 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 7.2, STS\(\_{[100]}\) performs the best and beats the other variants under both evaluation metrics. It demonstrates 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. 7.8(c).

Pruning of Features

Feature pruning is discussed in Section 7.1.2. 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 a significant drop on the performance of crowd attribute recognition. The detailed experiment in this section is first pruning 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 7.1.2, and then re-training the proposed deep models under the same setting as that of STS\(\_{[100]}\)\(^2\). Their mAUC and mAP are reported in comparison with the results by the default STS\(\_{[100]}\) in Table 7.3.

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) without other notations, STS\(\_{[100]}\) denotes the 100 frames-based S-CNN without feature cuboids pruning.

\(^2\)Without other notations, STS\(\_{[100]}\)
§ 7.3.2. Ablation Study of S-CNN

| $|N_r|$ | 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 7.3: 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.

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 [Chp. 5] and 3D-CNN [51]), 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 [53]. Besides, as shown in Table 7.3, compared with randomly pruning half of the filters ($|N_r| = 256$ _rnd_), 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 $H(n)$, 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 fulfill crowd attribute recognition.

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 94.04% and mAP score 62.55%, which improve the optimal results of single-branch models by 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 7.3, the S-CNN with $|N_r| = 100$ and $|N_r| = 256$ 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. 7.8(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...
7.3.3 Comparison with State-of-the-Art Methods

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

1. **DLSF+DLMF** [Chp. 5]. The DLSF+DLMF was originally proposed in Chapter 5 for crowd attribute recognition. It is a two-branch model with a late fusion scheme. The temporal feature learning in DLSF+DLMF is conducted by proposing three motion channels with temporal average-pooling. This method is employed as a reference with its default setting.

2. **Two-stream** [53]. The Two-stream contains two branches as a spatial net and a temporal net. Spatial nets are designed for capturing static appearance information, which trained on the single frame, while temporal nets for motion feature learning, trained on the optical flow volumes. The setting is adopted as same as that in [53] by inputting 10-frame stacking optical flow maps for temporal net. Besides, the parameters for temporal nets are also initialized with the VGG-16 model, as that in [54] for action recognition.

3. **3D-CNN** [51]. The 3D-CNN uses 3D kernels to learn both appearance and temporal features simultaneously. It requires very large memory to capture long-range dynamics. As [51] applied 3D kernels on hand-crafted feature maps, for fair comparison, the implementation is extracting features in lower layers of STS$_{[100]}$, and substituting $3 \times 3 \times 3$ 3D kernels for all 2D kernels after conv4$_{3}$ layer and cut off half kernel numbers.

---

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

---

**Figure 7.8**: 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.

etc, S-CNN presents a remarkable discriminative power for identification.
### § 7.3.3. Comparison with State-of-the-Art Methods

#### Figure 7.9: 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.

**Quantitative Evaluation**

As shown in Fig. 7.9, 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 methods 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
Figure 7.10: 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.

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 [53].

The performance of each attribute is also shown in Fig. 7.10. It depicts 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, “walk” is always with similar actions like “run” and “marching”.

Qualitative Evaluation

This section conduct qualitative evaluations, as shown in Fig. 7.8(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. The proposed method tends to capture the attributes more discriminatively with distinct motion patterns like “walk” but might feel difficulties in predicting attributes with subtle motions like “sit”.

7.4 Summary

This chapter presents a novel Slicing CNN (S-CNN) for crowd video understanding, with only 2D filters. It shows that the spatial \((xy-)\) filters capture appearance information, while temporal-slice \((xt-\text{ and } yt-)\) 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. The spatial feature cuboids pruning is further demonstrated to reduce redundancy leading to a sparser network. It is interesting to explore more strategies on feature cuboids selection in the future work.
Part III

Conclusions
Chapter 8

Conclusions

This thesis aims to explore the possibility of universal descriptions of the crowd dynamics across different crowded scenes in surveillance videos. In particular, this thesis attempts to solve three problems: (1) scene-independent properties for dynamic groups, (2) scene-independent crowd attributes from a large-scale crowd video set, and (3) spatial-temporal crowd motion patterns learned from the convolutional neural networks. These problems are not trivial since the crowd dynamics are complex and ambiguous, and the crowded scenes vary in densities, perspectives, and scales. It brings enormous challenges in capturing common properties and/or attributes from different crowded scenes. Moreover, it is not well-studied to apply the convolutional neural networks for video tasks.

The thesis first demonstrates that fundamental dynamic properties of groups, which are the primary units of a crowd, can be systematically quantified by visual descriptors. The thesis introduces a novel Collective Transition (CT) prior to capture the underlying distinct dynamics owned by groups. Based on the CT prior, a robust group detector is proposed and it reaches the state-of-the-art performance in comparison with the prior arts. Apart from the group detection, the proposed approach further characterizes and quantifies four types of group properties, collectiveness, stability, uniformity and conflict, from vision point of view. In particular, these group-level descriptors are scene-independent, and can be directly applied to novel scenes with diverse crowd densities and distributions. To demonstrate the effectiveness of the proposed group properties, this thesis constructs a group-based crowd dataset, named as CUHK Crowd Dataset.

In the second place, this thesis profiles the macro-level crowd properties instead of the group-level ones, in which the crowd attributes informatively describe a crowd video according to a series of basic elements referring to the locations, characters and events. A new large-scale crowd dataset (i.e. WWW (Who do What at someWhere) Crowd Dataset) is established in this thesis, including 10,000 crowd videos captured from various sources. A comprehensive set of crowd-related attributes are defined on the WWW crowd dataset, covering the common crowded scenes, subjects, and
actions. From the perspective of modeling, the attribute learning framework is a multi-
task learning deep model that jointly learns the appearance and motion features with
effective fusion schemes. Instead of directly inputting multiple frames for motion feature
learning, the proposed approach specifically designs a set of crowd motion channels
based on the group-level descriptions. This attribute-centric crowd analysis allows a
better performance in crowded scene understanding and provides potential abilities in
cross-scene event detection, crowd video retrieval, and crowd video classification.

At last, efficiently learning motion features with convolutional neural networks
(CNNs) is still understudied and the existing methods are inherently limited. To ad-
dress this problem, this thesis designs two novel spatio-temporal CNN architectures.
The first approach uniformly slices the raw 3D video volume from different dimensions
into 2D spatio- and 2D temporal-channels (i.e. $xy$, $xt$, and $yt$-stacks) as the input
of CNN model, named as Slicing Volumetric (S-Vol). It learns separated appearance
and dynamic features and effectively combines them. However, due to sampled selec-
tion from the whole video phase, it may lose some key information of either spatial
or temporal aspect. To overcome this drawback, this thesis then introduces another
more general approach, decomposing the 3D feature cuboids instead of video volume,
named as Slicing CNN (S-CNN). The S-CNN model can capture dynamics of different
semantic units, e.g. speed and acceleration of the groups or individual objects. The
semantic selectiveness can also guide to prune irrelevant filters respect to background
clutters.

There are several extensions in the future work:

(1) The thesis only considers group-level and macro-level crowd analysis. The methods
based on micro-level (i.e. pixel-level) can be explored in the future work. For ex-
ample, the WWW crowd dataset proposed in this thesis contains a set of attributes,
covering the scenes, subjects, and actions. Actually, it can be extended to a problem
of semantic-segmentation from pixel-level. It can further help scene understanding of
Who do What at someWhere (WWW).

(2) The Slicing CNN proposed in the Section 7 is not only limited to crowded scenes
but can also extended to other general scenes. And except this proposed formulation,
there are several variants which can also simulates 3D CNN by 2D CNN, such as
alternatively learn $xy$, $xt$, and $yt$-slice features in a single branch.

(3) Although CNN has good performance in crowd video analysis, it is worthy to
further explore whether RNN can provide more hints on crowd sequence formulation
or not in the future work. More analysis on how temporal information help spatial
information also deserves further study.


