The Graduate University for Advanced Studies
(SOKENDAI)
Japan

Scalable Approaches for Content-based Video Retrieval

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Informatics

by

Ngo Duc Thanh

2013

To My Family.

Acknowledgments

First and foremost I want to express my gratitude and thanks to my advisors, Prof. Shin'ichi Satoh and Assoc. Prof. Duy-Dinh Le. This dissertation would not have been completed without their valuable advice and endless encouragement. It is my honor to be their student and conducting my research under their supervision.

Thanks to my advisory committee members: Prof. Shin'ichi Satoh, Prof. Akihiro Sugimoto, Assoc. Prof. Imari Sato, Prof. Koichi Shinoda and Assoc. Prof. Duy-Dinh Le, for making their time available to help and to give detailed comments on my dissertation.

I would like to thank all my friends and colleagues at National Institute of Informatics (NII), Japan. I really enjoyed my discussions and collaboration with them. Finally, I cannot end without giving thanks to my family for their endless support and love.

Abstract of the Dissertation

Scalable Approaches for Content-based Video Retrieval

by

Ngo Duc Thanh

Doctor of Philosophy in Informatics

The Graduate University for Advanced Studies (SOKENDAI), Japan, 2013

Professor Shin'ichi Satoh, Advisor

With the development of modern technology, videos nowadays can be easily collected and stored. Because of the explosive growth of video data, dealing with scalability issue becomes crucial to any video retrieval system. This dissertation investigates scalable approaches for video retrieval, especially content-based video retrieval.

Content-based video retrieval (CBVR) generally refers to the task of retrieving relevant videos over a video corpus based on visual information derived from the videos themselves, given queries from users. In the most common way of searching visually similar videos, example images (or videos) representing what users want to search are provided as queries. We first follow this standard query-by-example paradigm to study scalable video retrieval approaches with a special focus on human face - an extreme meaningful source of information in videos.

Face retrieval in videos is a long-standing research issue. To utilize the variability of face appearances in video, sets of face images called face-tracks representing the appearance of characters in a shot, instead of individual face images,

are used in recent works. Although several approaches have been introduced, their extremely high computational cost limits their scalability to large-scale video datasets that may contain millions of faces of hundreds of characters. We present an efficient face-track matching approach which is scalable to such datasets while maintaining a competitive accuracy. Instead of using all the faces in the face-tracks to compute their similarity, our approach obtains a representative face for each face-track. As a result, the computational cost of matching is significantly reduced while taking into account the variability of faces in face-tracks to achieve competitive matching accuracy. Experiments are conducted on two large-scale face-track datasets obtained from real-world news videos. One dataset contains 1,497 face-tracks of 41 characters extracted from 370 hours of TRECVID videos. The other dataset provides 5,567 face-tracks of 111 characters observed from a television news program (NHK News 7) over 11 years. All face-tracks are automatically extracted using our proposed face-track extraction approach. We make both datasets publically accessible.

We then consider a general query-by-example scenario without limitation to face, in which users supply a query by cropping an object of interest from an input image. Having no prior knowledge about the retrieved database, users sometimes can not avoid selecting bad queries which return disappointed retrieval results due to the lack of relevant items in the database. In the second part of our dissertation, we tackle this problem by introducing an approach to facilitate users in formulating their queries. The key idea is to identify and recommend to users object instances actually exist in the database. We present an efficient approach for identifying and quantifying occurrences of multiple candidate object instances in a large-scale database. Instead of scanning the database for each candidate instance, we detect occurrences of multiple candidate instances simultaneously by investigating pairs of highly similar regions, knowing one pair is formed by

a candidate object region in the input image and a candidate object region in a video frame of the database. We formulate the problem of finding such pairs as an optimization problem, which can be efficiently solved by a branch-and-bound algorithm. Combing with inverted index technique, we achieve a novel and scalable approach for query recommendation in video retrieval.

Although query-by-example paradigm is intuitive and easy to use, one of its well-known limitations is that it can not handle video retrieval by semantic concepts. We address this problem in the final part of this dissertation. Within the scope of our work, the semantic concepts of interest are generic object categories. We propose an object categorization approach based on Multiple Instance Learning (MIL) to detect presences of the object categories in videos at frame-level (or image-level). To apply MIL, images are regarded as a bags and sub-windows in the images as instances of the bags. Learning a discriminative MI classifier requires an iterative solution. In each round, positive sub-windows for the next round should be selected. With standard approaches, selecting only one positive sub-window per positive image may limit the search space for global optimum; meanwhile, selecting all temporal positive sub-windows may add noise into learning. Our approach selects a subset of sub-windows per positive bag to avoid those limitations. Spatial relations between sub-windows are used as clues for selection. Our MIL-based approach is then combined with a global scene classifier in a generalized stacking framework to boost the categorization accuracy. Compared to other approaches required the same amount of annotation, it achieves a better balance between cost-effectiveness and accuracy.

Table of Contents

1 Introduction **1**

1.1 Background 1

1.1.1 Video Preprocessing 2

1.1.2 Content Analysis 5

1.1.3 Query and Retrieval 6

1.2 Motivations and Addressed Problems 8

1.3 Summary of Contributions 14

1.4 Outline of The Dissertation 15

2 Face Retrieval in Large-scale Video Datasets **16**

2.1 Introduction 16

2.2 Related Works 19

2.3 Framework Overview 22

2.4 Face-track Extraction 23

2.4.1 Face-track Extraction based on Tracking Points 24

2.4.2 Removal of Frames Containing Flash Lights 25

2.4.3 Point Generation and Tracking 27

2.4.4 Our Proposed Approach for Face-track Extraction 29

2.5 Matching Face-tracks 32

2.6 Experiments 36

2.6.1 Evaluation of Face-track Extraction 36

2.6.2 Evaluation of Face-track Matching 39
2.7 Summary 56

3 Query Recommendation for Video Retrieval 57
3.1 Introduction 57
3.2 Related Work 61
3.3 Framework Overview 63
3.3.1 Select Candidate Images Using Inverted Index 66
3.3.2 Select Candidate Regions in Each Image 66
3.3.3 Maximal Clique Analysis Algorithm 67
3.4 Branch-and-Bound Framework for Finding Top Region Pairs with the Highest Similarity Scores 68
3.4.1 Organizing Regions into Hierarchical Structures 73
3.5 Experiments 75
3.5.1 Datasets 75
3.5.2 Performance Evaluation 76
3.5.3 Results 78
3.6 Summary 81

4 MIL-based Object Categorization for Content Analysis 82
4.1 Introduction 82
4.2 Related Works 85
4.3 Support Vector Machine for Multiple Instance Learning 86
4.4 The Former Approaches of SVM-based Multiple Instance Learning 87

4.5 Support Vector Machine with Spatial Relation for Multiple Instance Learning . . . 88
4.6 Combining with a Global Scene Classifier . . . 92
4.7 Experiments . . . 95
4.7.1 Dataset . . . 95
4.7.2 Bag and Instance Representation . . . 96
4.7.3 Evaluated Approaches . . . 97
4.7.4 Experimental Results . . . 101
4.8 Summary . . . 102

5 Conclusion . . . 103
5.1 Summary of Research . . . 103
5.1.1 Face Retrieval in Large-scale Video Datasets . . . 103
5.1.2 Query Recommendation for Video Retrieval . . . 104
5.1.3 MIL-based Object Categorization for Content Analysis . . . 105
5.2 Future Directions . . . 105

References . . . 107

List of Figures

1.1 An overview of a video retrieval system. 2

1.2 An illustration of video structure. 3

2.1 Faces in a face-track with different facial expressions and poses. . 17

2.2 Framework Overview. In the off-line stage (left, blue box), face-tracks in videos are extracted using our proposed face-track extraction approach. This process is performed once for a video dataset. Then, our face-track matching approach estimates the similarity between a given query face-track and each face-track in the dataset to return a ranked list as the output of the online retrieval stage (right, red box). 23

2.3 An overview of a face-track extraction approach based on tracking points. Dashed lines connecting points in frames represent point tracks. One color is for one individual point track. Point tracks with circles points indicate tracks whose points are successfully tracked throughout the shot. A track whose point can not be tracked at a frame and replaced by a new point is denoted with triangle points. The point track with yellow squares demonstrates a track with tracking error. A neutral threshold 0.5 is used in comparing faces for grouping in this example. Thus, all detected faces are grouped into one face-track. The figure is best viewed in color. 25

2.4 A real example of unreliable tracking results due to flash lights. Green points indicate track point positions in current frame. Red lines connecting green points and yellow points represent motions of points from the previous to the current frame. 26

2.5 Illustration of a problem that occurs when generation of track points is independent of face detection results. Two faces C_1 and C_2 of character C in this example have no track points passing through them. Thus, they are considered to be two single-face face-tracks. 28

2.6 A real example with simplified illustration of tracking errors due to occlusion. Although all track points are retained in such cases, their tracks cannot help to connect face D_3 with other faces D_1 and D_2 of the same character, given the threshold of CGM for grouping two faces is 0.5. 30

2.7 A step-by-step illustration of our approach for face-track extraction. 32

2.8 An illustration of our proposed k-Faces approach. In (a), each face-track is first divided into k equal parts ($k = 4$, in this example). The middle face (with bright-colored outline) is selected in each part to represent the part. Then, k selected faces (marked with stars) are used to compute a mean face (circle or triangle) on the feature space. The mean face now represents the whole face-track. Finally, in (b), the similarity of face-tracks is estimated based on the distance between their mean faces. 33

2.9 Statistical information on our datasets. (a) shows the distribution of face-tracks over their lengths; (b) and (c) present the number of face-tracks for the top 20 individual characters in each dataset. 40

2.10 Pair-wise based approaches. Based on the possible pair-wise distances of faces in face-tracks A and B, we have: $pair.min(A,B) = \min dist(a_i, b_j)$, $pair.max(A,B) = \max dist(a_i, b_j)$, and $pair.mean = M^{-1}N^{-1}\sum_i \sum_j dist(a_i, b_j)$, where $i = \overline{1,M}$, $j = \overline{1,N}$. Knowing that a_i and b_j are feature vectors, the function $dist(a_i, b_j)$ is used to compute their distance on the feature space. Different types of distances are used for *dist* such as L1 distance, Euclidean distance, and Cosine distance. In this illustration, $M = |A| = 4$ and $N = |B| = 3$. 43

2.11 MAP(s) of the approaches in the TRECVID dataset. 46

2.12 MAP(s) of the approaches in the NHKNews7 dataset. 47

2.13 Face-tracks of President George W. Bush recorded in 2001 (top) and 2009 (bottom). 48

2.14 MAP(s) of the evaluated approaches in Honda/UCSD (left) and Buffy (right) datasets. 53

2.15 Evaluation on *representativeness* of mean faces computed by k-Faces.$Temporal$ and k-Faces.$KMeans$ over TRECVID (top) and NHKNews7 (bottom) datasets. 55

3.1 Having no idea about the database, how do users know whether their intentional search item will return relevant results , without any trial search ? If not, which query should be used instead to search or to explore the database? Recommend-Me targets to answers these questions. In this example, it recommends a stop sign (green box), which occurs in 4 images of the database, rather than other candidate items (yellow boxes), which can not be found in the database. 58

3.2 Framework pipeline. (a) Step 1: a list of candidate images are returned by using inverted index. (b) Step 2: only rectangular regions (green boxes) that tightly bound segments throughout the hierarchy are selected. One node of the hierarchy represents one rectangular region. (c) Step 3: among all possible region pairs (blue arrows) between regions of the initial query image and regions of all images in the database, only the top TP pairs with the highest similarity scores (red arrows) are returned. (d) Step 4: overlapping rectangular regions in the initial query image are grouped using maximal clique analysis. (e) Step 5: for each group, all the of images such that one image contains at least one match of one member region of the group are counted. These numbers (green numbers) are used to rank the groups. One group represents one recommendation (green boxes). Best viewed in color. 64

3.3 Branching Step. The parameter space covered by $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ is divided into disjoint parts $\mathcal{P}(\mathcal{S}_l(n^A_1), \mathcal{S}_l(n^B))$,..,$\mathcal{P}(\mathcal{S}_l(n^A_{c_A}), \mathcal{S}_l(n^B))$, regarding 3.2. 71

3.4 Bounding Step. A parent region (orange) covers its child regions (green and blue) on the image space. Thus, a code-world detected inside the child regions is also inside the parent region. 74

3.5 Evaluation results on two datasets (Oxford: left column, MQA-1M: right column). Top figures on both columns demonstrates that our proposed approach is significant better than the standard inverted index approach based on indexing regions in term of accuracy. Other bottom figures show the trade-off accuracy vs. efficiency of our approach at different values of $\mathcal{TP}$. 78

3.6 One image is one initial query image. In these examples, Recommend-Me successfully identified the items. 79

3.7 Examples of region pairs found by our approach. A pair is form by a region in the input query image and a region in an image of the database. The first image (from left to right) is the input query image, the others are images of the MQA-1M database. 80

4.1 Illustration of positive candidate selection for the next learning round by different approaches. mi-SVM selects all temporal positive instances (*orange*). MI-SVM selects only the most positive instance per positive bag (*dash-purple*). Meanwhile, our approach selects a subset of spatially related instances (*green*) per positive bag based on their overlap degree with the most positive instance of the bag. 90

4.2 Example images taken from Caltech 101. From top to bottom are images of airplanes, cellphones, faces and motorbikes respectively. 96

List of Tables

2.1 Detailed information on the videos used in our experiment. FT-F indicates face-tracks having flash frames. FT-OM indicates face-tracks containing occlusions or that do not appear at the beginning of the shot. O stands for occlusion and, M for middle. 37

2.2 Performance of the evaluated approaches. 38

2.3 Processing time of the evaluated approaches. 38

2.4 Statistical comparison between our datasets and other benchmark datasets. 42

2.5 Mean Average Precision and processing times (in seconds) of the evaluated approaches. Note that the preprocessing process is only performed once for a given dataset. And, k of the approaches in this table is equal to 20. 50

3.1 The number of images for each type of item in our dataset. 76

4.1 Average classification accuracy of the evaluated approaches on Caltech 4. Note that the performance of MA is computed on 3 categories (airplanes, faces and motorbikes) only due to the lack of ground-truth object box of the category cars_brad. 98

4.2 Average classification accuracy of the evaluated approaches on Caltech 101. 98

4.3 Average classification accuracy of the evaluated approaches on 10 categories of Caltech 101. 99

4.4 Average classification accuracy of all evaluated approaches on 101 categories of Caltech 101. 99
4.5 Average classification accuracy of the evaluated approaches on 10 categories. 100

List of Abbreviations

AP	Average Precision.
BB	Branch and Bound.
BoW	Bag of Words.
CBVR	Content-based Video Retrieval.
CMSM	Constrained Mutual Subspace Method.
ESR	Efficient Subimage Retrieval.
ESS	Efficient Subwindow Search.
GMM	Gaussian Mixture Model.
MAP	Mean Average Precision.
MIL	Multiple Instance Learning.
MSM	Mutual Subspace Method.
SIFT	Scale Invariant Feature Transform.
SVM	Support Vector Machine.

CHAPTER 1

Introduction

1.1 Background

Video retrieval refers to the task of retrieving the most relevant videos in a video collection, given a user query. A robust video retrieval system can bring benefits to a wide range of multimedia applications such as news video analysis, video-on-demand broadcasting, commercial video analysis, digital museums or video surveillance. In the past, when video collections are relatively small, video retrieval can be done using keywords manually annotated by specialist. However, due to recent exponential growth of video data supported by advances in multimedia technology, manual annotation has been no longer tractable. Consequently, it creates a great demand on automatic video retrieval systems.

In general, a video itself contains multiple types of information including: i) embedded video metadata such as title, description, creation date, author, copyright, duration, video format; ii) audio content, from which transcripts can be obtained by applying speech recognition techniques; and, iii) visual content. In the context of this dissertation, we address video retrieval systems based on information derived from visual content of videos. Such systems are called content-based video retrieval system.

Building a content-based video retrieval system requires solutions to several problems. We briefly introduce parts of a typical content-based video retrieval

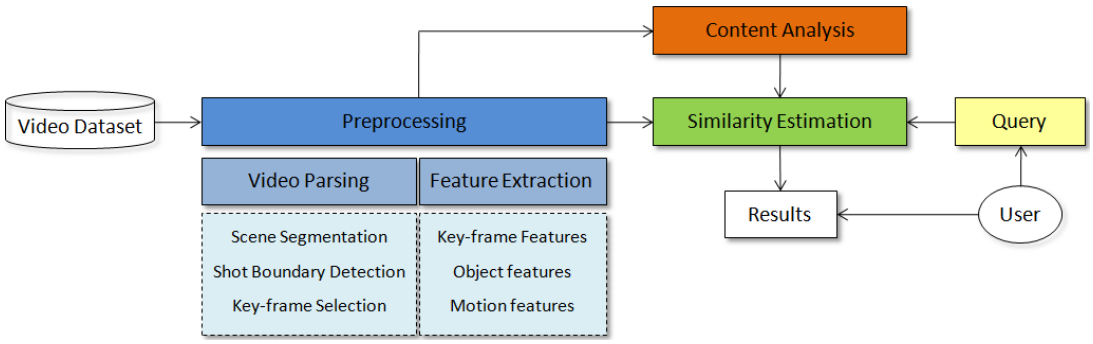


Figure 1.1: An overview of a video retrieval system.

framework (summarized in Figure 1.1) in the following.

1.1.1 Video Preprocessing

1.1.1.1 Video Parsing

Videos are usually organized as a hierarchical structure of scenes, shots, and frames (illustrated in Figure 1.2). The goal of video parsing is to divide a video into a set of such structural elements. Depending on a specific application, video elements of a corresponding type will be used as the fundamental processing units. For instance, object based video retrieval may need to analyze videos at frame level. Meanwhile, event based video retrieval mainly targets to shots.

Video parsing is a prerequisite step towards video content analysis. Approaches for video parsing includes scene segmentation, shot boundary detection, and keyframe selection.

Shot boundary detection. A shot is defined as a sequence of frames captured by a single camera operation. The interruptions between camera operations indicate the shot boundaries, thus make frames in a shot strongly correlated each other. There are two basic categories of shot boundaries, depending on the tran-

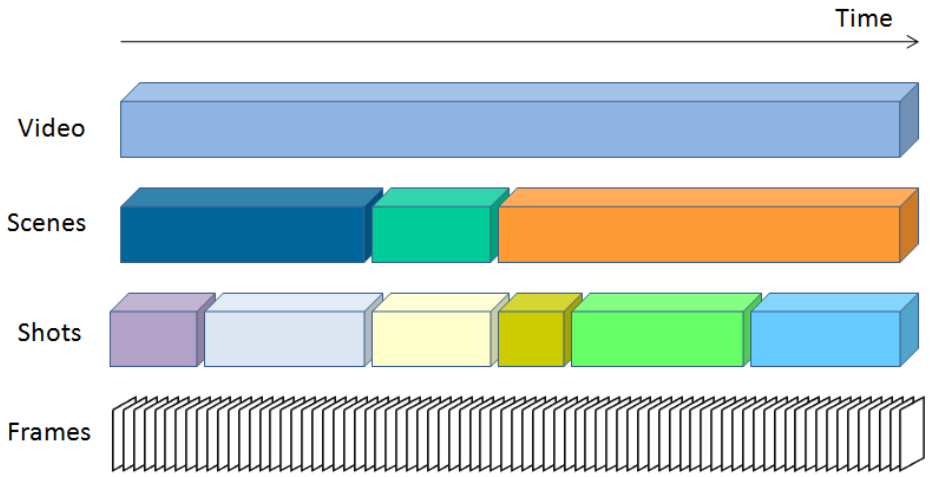


Figure 1.2: An illustration of video structure.

sitions between shots. A shot boundary is categorized as CUT if the transition between shots is abrupt. An abrupt transition occurs in a single frame only. Otherwise, if a transition spreads over a number of frames, the shot boundary is called gradual transition (GT). Gradual transitions are mainly created by editing effects. Shot boundary detection (SBD) aims at detecting such transitions between consecutive shots.

Shot boundary detection approaches are usually based on measuring dissimilarities between frames of which visual features are extracted. The dissimilarities between pairs of consecutive frames or between frames within a window [1] can be measured using several types of distance such as 1-norm cosine distance, Euclidean distance, histogram intersection distance, the chi squared distance [1, 2], or the earth mover's distance [3]. Using the measured dissimilarities, shot boundaries are detected by thresholding [4, 5], graph partitioning [6], or applying learned classifiers.

Keyframe selection. Consecutive frames within a sequence (i.e. a video or

a shot) are highly redundant. Thus, a set of certain frames which best reflect content of the sequence should be selected to represent the sequence. Such frames are called keyframes or representative frames. The ultimate goal of keyframe selection is to eliminate redundant frames while reserving salient frames as much as possible.

Recent approaches to keyframe selection mainly target to minimize the dissimilarities between each selected keyframe with its neighboring frames, or maximizing dissimilarities between the selected frames. Approaches of the first strategy include clustering-based and curve simplification-based approaches. Meanwhile, approaches following the second strategy consist of those based on sequentially selecting a keyframe which is significant different to the previous selected keyframe [7, 8], or minimizing the correlations between keyframes within the selected set [9].

Scene segmentation. A video usually consist of scenes, where each scene may contain one or more shots. Shots of a scene are about the same subject or theme. Thus, scenes are also known as story units and they are at higher semantic level than shots. Scene segmentation is to decompose a video into scenes. Regarding to [10], scene segmentation approaches can be divided into four categories mostly depending on their strategy such as merging, splitting, statistical modeling, and boundary classification.

Merging based approaches gradually merge similar shots to form a scene following a bottom up style [11]. In contrast, splitting based approaches split the whole video into separate coherent scenes using a top down style [12]. With approaches based on statistical model, they aim at constructing statistical models of shots such as stochastic Monte Carlo [13], GMM [14], or a unified energy minimization framework [15] for scene segmentation. With boundary classification

based approach, they extract features of shot boundaries and then use them to classify the shot boundaries as scene or non scene boundaries [16].

1.1.1.2 Feature Extraction

Given video elements parsed by video parsing approaches, the next crucial step to construct a video retrieval system is to extract features from the video elements so that they can be used for video content analysis. Common features include features extracted from static keyframes and motion features extracted from sequences of frames.

Static keyframes are the basic video elements reflecting video content. Feature of static keyframes are basically derived from colors, textures, and shapes in keyframes or their regions. Recent works usually employ Bag-of-Visual-Word (BoVW) model borrowed from text retrieval for feature presentation. *Visual words* (i.e. salient regions) are first extracted from keyframes. They are then used to compose histograms of *visual words* representing individual keyframes or shots. Because such features are extracted from static keyframes, they can not capture motions which are mainly caused by camera movements and foreground object movements in videos. Motion features play a more significant role in video indexing and retrieval by events or human actions [17–20].

1.1.2 Content Analysis

The aim of content analysis is to analyze videos so that they can be indexed and retrieved by their content. The indexed terms are of several types including common patterns [21–23], video genres [24–26], events [27–29], human actions, object categories or concepts appearing in the videos [30, 31]. Video content analysis requires techniques for video data mining, video classification and annotation.

Recent approaches for video classification and annotation follow a typical strategy. First, low level features are extracted. Then, related category or concept classifiers are trained. Finally, the classifiers are used to map the features of video elements (e.g., videos, shots, frames) to the corresponding labels of the concepts or categories. Basically, the main challenge is to handle the *semantic gap* between low level feature extracted from videos and semantic concepts perceived by human being (e.g., video genres, events, object categories). However, it is well known that bridging the semantic gap is challenging due the variations in visual appearance of semantic concepts. Furthermore, human participation, in the form of manual annotation, is always required in order to train classifiers. This makes video annotation and classification approaches inflexible as they are applied to different domains.

1.1.3 Query and Retrieval

Query formation is at the online stage of a video retrieval system. As a query is given, the retrieval system perform retrieval by applying similarity estimation approaches or simply scanning over an index table to return most relevance video elements in accordance with the query.

1.1.3.1 Types of query

To formulate a query, users usually submit an example which visually represents what they want to search. This type of query is called query by example. Depending on the application as well as users' interest, the example can be a whole image, a bounding region of object of interest in an image, or a sketch [32]. Query by example is very useful when users want to search for the same object or scene under slightly varying circumstances and when the example images are available

indeed. If proper example images are unavailable, it is impossible to perform searching. Query by example is regarded as non semantic based video query type since it can not capture the semantic search intention from the example. To narrow such a semantic gap, one way is to use textual keywords in queries that precisely describe what the user is looking for. Nevertheless, this would require a thorough and correct textual annotation of the entire video corpus, which is not only very tedious but in fact, impossible for humans to perform.

Query by concept paradigm is yet another way to bridge the semantic gap between users' search intention and visual video content. With this paradigm, users can select a predefined concept, after which the retrieval system return relevant video elements based on presence of the concept detected by concept detectors or video annotation and classification approaches. By doing this, the semantic consistence between users' search intention and visual content of videos is kept. Furthermore, it bypasses the limitation of query by example paradigm such as needed existence of example image. However, because the concepts are predefined, the retrieval system cannot support searching concepts out of the cope. And, human participation is required in order to train the detectors.

1.1.3.2 Similarity Estimation

Video similarity estimation play an important role in a content based video retrieval system. The choice of approaches depends on the query type.

Feature-based Similarity Estimation. This is mostly for query by example paradigm. The most direct measure of similarity between video elements and a query is the distance between their extracted features. According to different user's demands, features of static keyframes, object features can be used to measure their similarity. However, selecting appropriate types of feature is one of the

most critical problems. Furthermore, the estimation process is costly and time consuming if the dataset is huge.

Concept-based Similarity Estimation. Matching the name of each concept with query terms is the simplest way of finding the videos that satisfy the query. Basically, if the concept detection is done for all videos of the retrieved dataset in the offline stage, the retrieval system can respond to users' search request in constant time by scanning an inverted index table. If users retrieve multiple concepts simultaneously, returned videos elements can be ranked by voting. The limitation of this approach is that it only supports searching certain concepts with corresponding trained classifiers in advance.

1.2 Motivations and Addressed Problems

Firstly, scalability is no longer a plus feature but a definite requirement of nowadays video retrieval systems due to the exponential growth of video data. Applications involving video retrieval can hardly be practical if the system is not scalable. Investigating scalable approaches is therefore importance to content-based video retrieval.

A system is defined to be scalable if it can be easily extended to handle a much larger amount of data and its overall consumption of resources increase gracefully with the size of the database. In other words, the key factor that affects scalability of a system is its cost-effectiveness. There are two main types of cost possibly consumed by a content-based video retrieval systems: computational cost and human annotation cost. Human annotation cost can be regarded as the amount of manual annotation needed in developing the system such as annotation for training classifiers in content analysis approaches. Minimizing these costs is

essential to achieve scalable retrieval systems. In accordance with that, following issues come into being: 1) how to reduce the costs in most expensive processes e.g. similarity estimation and content analysis ? and 2) how to balance cost-effectiveness and accuracy of a content-based video retrieval system while cost-effectiveness is usually inversely proportional to accuracy ? These issues must be considered in developing scalable approaches.

Secondly, despite a great deal of progress has been made in some of the core aspects of video retrieval, there is still much more room for improvement especially when scalability is taken into account. In the following, we present the scopes of the problems addressed and overview of our approaches.

Face retrieval in large-scale video datasets. Face retrieval plays important role in content-based video retrieval due to the fact that human face is an important source of information in video. We target scalable face retrieval approaches which can deal with real-world datasets of such scale that have never been considered in the literature.

In the manner of query-by-example paradigm, a video retrieval system based-on face usually consists of two main processing steps: face detection and face matching. Faces in the retrieved video database is first detected. They are then matched with the query face to return relevant results. Whereas conventional approaches take into consideration single face images as the basic units in extracting and matching, recently proposed approaches have shifted toward the use of sets of face images, called face-tracks, to utilize multiple face images of characters within a video shots for accurate representations. By exploiting the plenteous information from face-tracks, face-track-based approaches achieve a more robust and stable performance. However, in order to use face-tracks, we have to face two main challenging problems. First, face-tracks in videos should be robustly

extracted. A noisy face-track may result a very poor representation of a character appearance. Second, high computational cost is required for face-track matching. Addressing these problems is important to video retrieval systems using face-tracks.

We follow a common strategy of existing approaches for face-track extraction that use trackers to establish the connections between detected faces of the same characters. A point tracker is used instead of region trackers for efficiency. The basic idea is that if two faces detected in different frames share a large amount of similar point tracks (i.e. trajectories of tracked points) passing through both of them, they are likely to be faces of the same character. Although using point tracker is efficient, the reliability of its resulting point tracks is sensitive to sudden illumination changes caused by flash lights. We handle this problems by detecting and eliminating frames containing flash lights. In addition, we introduce yet another techniques to keep point tracks robust against partial occlusions and scattered appearances of characters by making point generation adapted to face appearances and replacing bad tracked points.

To reduce the computational cost for face-track matching, we argue that using all faces in a face-track for learning a representation is redundant since faces detected in relatively closed frames are highly similar. We thus propose to sample a set of faces from the original face-track following their temporal order of appearance. We use the mean face of the sampled face as the representative face for the whole face-track. Matching between face-track is done by comparing their representative faces. As a result, we still take into account information from multiple exemplar faces of face-tracks while significantly reduce computational cost. Experiments are conducted on two large-scale face-track datasets obtained from real-world news videos. One dataset contains 1,497 face-tracks of 41 characters

extracted from 370 hours of TRECVID videos. The other dataset provides 5,567 face-tracks of 111 characters observed from a television news program (NHK News 7) over 11 years. All face-tracks are automatically extracted using our proposed face-track extraction approach. We make both datasets publically accessible.

Query recommendation for video retrieval. Face retrieval, to some extent, is a particular example of retrieval approaches following query-by-example paradigm. In a more general query-by-example scenario without limitation to face, users usually supply a query by cropping an object of interest from an input image. The retrieval system then returns a list of relevant images, assuming images are video frames sampled by key-frame selection approaches. The images are expected to contain instances of the object of interest. However, despite the robustness of the retrieval system, there are cases in which users are disappointed with their search results due to the fact the object has no other instances in the retrieved database. Without prior knowledge about the database, users of a retrieval system have no clues to avoid such cases. This motivates us to tackle a query recommendation approach to facilitate users in forming their queries. Query recommendation purely based on visual content has not been considered in existing retrieval systems.

Given an input image from users, our aim is to automatically recommend object instances that both the input image and the retrieved database have in common. An object with higher number of occurrences will be ranked higher in the recommendation list. In order to do that, we need to address several issues. First, the number of candidate object instances in the input image is huge. Basically, any region in the image can be a candidate object instance. Second, even if a candidate object instance is selected, quantifying its occurrences in the database is not trivial. A naive approach is for each candidate region in the input

image R_k^Q, scan all regions R_n^I in images in the database; and compute similarity of region pairs (R_k^Q, R_n^I) to find the number of matches. This process is repeated for all candidate regions in the input image. Therefore, the main problem in building such a system is to handle such a huge number of similarity evaluations of region pairs that are computationally too expensive.

To address this problem, we propose a two-stage approach that uses different treatments for different region pairs. Specifically, in the first stage, we use inverted index to quickly filter out a large number of images in the database that are not relevant to any region of the input image. In the second stage, more complex processing is used for promising region pairs. We propose a branch-and-bound (BB) framework that allows to significantly reduce the number of similarity evaluations compared with exhaustive search to identify the region pairs of highest score. We demonstrate the scalability and performance of our system on two public datasets of over 100K and 1 million images. The approach can be easily applied to video datasets by applying key-frame selection approaches to select frames from videos and regard them as images.

Object categorization for content analysis. Although query-by-example paradigm is intuitive and easy to use, one of its well-known limitations is that it can not handle semantic search due to the gap between low-level visual feature and semantic concepts perceived by human being. Query-by-concept paradigm is a way to narrow the gap. In this paradigm, presence of predefined concepts are first detected in video elements e.g. video frames. Users retrieve the database by selecting concepts of interest and then obtain corresponding videos having the concept detected. Among possible concepts, we target generic object categories since they widely appear in our daily life as well as video content.

To detect presence of object categories in frames, object detectors (e.g., face

detector [33], pedestrian detector [34], etc.) can be used. However, a huge amount of annotated samples are required in order to train the detectors. Preparing such training samples is tedious since annotators must specify locations and bounding regions of the positive objects in images. In terms of human annotation cost, it is not cost-effectiveness; thus, unscalable as the number of object categories is sifnificantly increased. To reduce the cost, object categorization approaches using image-level annotations should be employed, instead of object detection approaches. However, image-level annotations are ambiguous for learning a robust object model since the object region and background region within one training image share the same label. To eliminate labeling ambiguity, we investigate Multiple Instance Learning (MIL) based object categorization approaches.

In MIL setting, groups and their samples are usually called bags and instances. A training group is labeled positive, if it has at least one positive instance. Otherwise, it is labeled as a negative bag. Given training labels are for groups, MIL approaches can learn to classify samples of the groups. If we consider a bag as an image and instances in the bag as sub-windows in the image, MIL approaches can be applied to detect object regions. However, one drawback of existing MIL approaches as they applied to our scheme is that they disregard the correlations or spatial relations between sub-windows (i.e. instances) within an image (i.e. bag) in their iterative learning process. Such relations between sub-windows can be clearly observed as: if a sub-windows is said containing an object, its highly overlapped sub-windows should contain the object also. We thus propose to improve the approaches by incorporating spatial information into learning. The proposed MIL-based approach is then combined with a global scene classifier in a generalized stacking framework to boost the accuracy.

1.3 Summary of Contributions

The original contributions of this dissertation are as follows:

- We introduce approaches for face retrieval in large-scale video datasets. First, we present a robust and efficient approach to extract face-track from real-world videos which are news video recorded over years. Second, a scalable approach for face-track matching is proposed. The matching approach achieved competitive accuracy compared to state-of-the-art approaches while it is hundreds to thousands times faster than others.

- We prepare two real-world face-track datasets of such scales that have never been considered in the literature. One dataset contains 1,497 face-tracks of 41 characters extracted from 370 hours of TRECVID videos. The other dataset provides 5,567 face-tracks of 111 characters observed from a television news program (NHK News 7) over 11 years. All face-tracks are automatically extracted using our proposed face-track extraction approach. We make both datasets publically accessible by research community.

- We present a novel approach for query recommendation to facilitate users in forming their queries. The recommendations help users to select good search query which certainly return relevant results, to rapidly refine the input query image or to explore the database using the recommendation as hints. An efficient solution based on a two-stage approach using branch-and-bound and inverted index techniques was also presented. To the best of our knowledge, the approach is the first attempt toward visual query recommendation.

- We propose an object categorization approach based on Multiple Instance Learning (MIL) to detect presences of predefined object categories in videos

at frame-level. By incorporating spatial information between sub-windows into learning, it is more accurate than original MIL approaches as well as standard categorization approaches. Compared to other approaches consuming the same amount of annotation cost, it achieves better balance between cost-effectiveness and accuracy.

1.4 Outline of The Dissertation

The remaining of this dissertation are organized as follows:

- **Chapter 2** presents our proposed approaches for face-track extraction and face-track matching for face retrieval in large-scale video datasets.
- **Chapter 3** presents our proposed approach for query recommendation based on Branch-and-Bound (BB) and Inverted Index techniques.
- **Chapter 4** introduces our proposed object categorization based on Multiple Instance Learning.
- **Chapter 5** gives summary of our dissertation and future works.

CHAPTER 2

Face Retrieval in Large-scale Video Datasets

2.1 Introduction

Developing an accurate face retrieval system is not a trivial task because of the fact that the imaged appearance of a face changes dramatically under large variations in poses, facial expressions, and complex capturing conditions. Besides accuracy, efficiency is also an issue in such a face retrieval system because the scales of available datasets are rapidly getting larger, for instance, exceeding thousands of hours of videos with millions of faces belonging to hundreds of characters. Thus, accurate and efficient approaches to face retrieval are always required.

A face retrieval system generally consists of two main steps. The first step is extracting the appearance of faces in videos. The second step is matching the extracted ones with a given query to return a ranked list. Whereas conventional approaches take into consideration single face images as the basic units in extracting and matching [35–37], recently proposed approaches have shifted toward the use of sets of face images called face-tracks. A face-track contains multiple face images belonging to the same individual character within a video shot. The face images in a face-track may present the corresponding character from different viewpoints and with different facial expressions (as shown in Figure 2.1). By exploiting the plenteous information from the multiple exemplar faces in the face-tracks, face-track-based approaches are expected to achieve a more robust

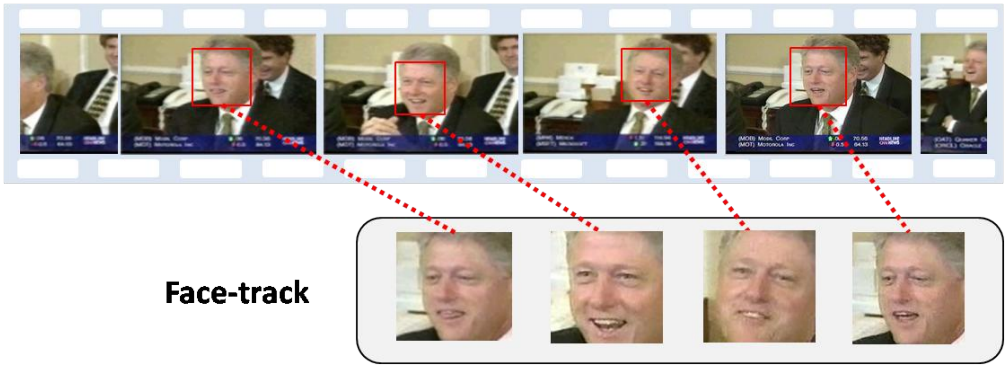


Figure 2.1: Faces in a face-track with different facial expressions and poses.

and stable performance.

Once all the face-tracks in the video shots are extracted, they are matched with the query to return a ranked list as the output of the face retrieval system. Because each face-track is a set of face images, matching face-tracks can essentially be thought of as a problem of matching image sets. Several approaches have been introduced to deal with this problem [38–43]. They differ in the ways in which the sets are modeled and the similarity between sets is computed. Using these approaches, image set has been modeled in different ways, such as distributions [38], subspaces [41–43], a convex geometric region in a feature space [39], or more general manifolds [40]. Although these approaches have shown promising results in benchmark datasets, they require high computational costs to characterize the representation of face-tracks, such as computing the convex geometric region in [39], the probability in [41], and the eigenvectors in [41–43]. Their complexity in modeling face-tracks and estimating the similarity between face-tracks limits their practicability in large-scale datasets.

This work provides a threefold contribution toward solving the above problems.

Robust face-track extraction from news video. Face-tracks should be extracted in advance to perform face-track matching. For this purpose, we propose a point tracker based face-track extraction approach, which is very efficient compared to approaches using an affine covariance region tracker or face clustering. The basic idea is that if two faces detected in different frames share a large amount of similar point tracks (i.e. trajectories of tracked points) passing through both of them, they are likely to be faces of the same character. To make point tracks reliable and sufficient in number for grouping faces of multiple characters throughout a shot, we introduce techniques to handle problems due to flash lights, partial occlusions, and scattered appearances of characters. All of these problems have not been carefully considered in former face-track extraction approaches, especially within the domain of news videos. By combining these techniques, our approach achieves a significant improvement to accuracy compared to a state-of-the-art approach [44].

Efficient face-track matching. We introduce an approach that significantly reduces the computational cost for face-track matching while maintaining a competitive performance with state-of-the-art approaches. Based on the observation that face-tracks obtained by tracking provide highly similar faces in consecutive frames, we argue that it is redundant to use all the faces in a face-track for learning the variation of faces. Thus, a set of faces is sampled from the original face-track for matching. The size of the set is much smaller than that of the original face-track. The mean face of the sampled faces in the set is then computed. The similarity between two face-tracks is based on the distance between their mean faces.

Large-scale face-track datasets from real-world news videos. We investigate the problem of face-retrieval in news video datasets whose scales have

never been considered in the literature. Our first dataset is from 370 hours of TRECVID news videos and it contains 405,887 detected faces belonging to 41 individuals. The second dataset includes 1.2 million faces of 111 individuals observed in the NHK News 7 program over 11 years. The total number of available face-tracks is 5,567. The number of occurrences of each individual character varies from 4 to 550. Both datasets are published to assist the research community.

2.2 Related Works

Face-track extraction. Face-track extraction is a key step in video-based face retrieval systems. Existing studies on automatic face-track extraction follow a standard paradigm that consists of two basic steps, detecting faces in frames and grouping faces of the same character into face-tracks. In the first step, the Viola-Jones detector is usually used to detect near frontal faces in frames of videos. In the second step, the detected faces of the same character are grouped by using either clustering [45] or tracking approaches [44, 46, 47]. In [45], Ramanan et al. built a color histogram for the hair, face, and torso associated with each detected face in a frame. A concatenated vector of the normalized color histogram represented the face. They then clustered all vectors to obtain groups of similar faces, using agglomerative clustering. The limitations of this approach include its high computational cost for constructing and clustering high-dimensional representation feature vectors and its dependence on determining a reasonable threshold for the clustering algorithm to ensure that no group contains faces of multiple characters and the groups are not over-fragmented.

On the other hand, Everingham et al. in [44] and Sivic et al. [46] proposed the use of tracking approaches to associate the detected faces of the same character. In [46], an affine covariance region tracker of [48] is used. This tracker can

develop tracks on deforming objects, where the between-frame region deformation can be modelled by an affine geometric transformation plus perturbations. The outcome is that a face can be tracked (by the collection of regions on it) through significant variations in poses and changes in expressions, allowing distant detected faces to be associated. One of the main drawbacks of this approach is its high computational cost for locating and tracking affine covariant regions. In contrast, Everingham et al. in [44] used a more efficient tracker, which is Kanade-Lucas-Tomasi (KLT) tracker [49], to create a set of track points starting at some frames in a shot and continuing until some later frames. Grouping faces in different frames for one character is based on enumerating the track points shared between faces. However, because the KLT tracker is sensitive to illumination changes and partial occlusions, additional techniques are required to obtain accurate face-track extraction results. Another way of using a tracker for face-track extraction was recently introduced by Merler et al. in [47]. Instead of using tracking results to connect detection results, they combine both of these to estimate the optimal positions of faces. Their online multiple instance learning tracker is expensive and the linear combination is sensitive to parameter changes.

Face-track matching. There are two major categories of approaches to using multiple-exemplars of faces in face-tracks (i.e. sets of face images) for robust face matching and recognition. The approaches in the first category [50–53] make use of both face images and the temporal order of their appearances. The face dynamics within the video sequence are modeled and exploited to improve recognition accuracy. For instance, Li et al. [54, 55] introduced an approach to modeling facial dynamics by constructing facial identity structures across views and overtime in the Kernel Discriminant Analysis feature space. Edwards et al. [56] proposed learning the mutation of individual faces through video sequences by decoupling sources of image variations, such as poses, facial expressions and

illumination. They then used the trained statistical face model to incorporate identity evidence over a sequence. In [52], Liu and Chen used an adaptive Hidden Markov Model (HMM) for this face recognition problem. In the training phase, they created a HMM for each character to learn the statistics and temporal dynamics using the eigen-face image sequence. The implicit constraint of these approaches is that the dynamics of faces should be temporally consecutive. In general, this constraint is not always satisfied.

Without relying on temporal coherence between consecutive images, the approaches in the second category use multiple face images only and treat the problem as a set-matching problem. These approaches are differentiated based on the ways in which the sets are modeled and the similarity between sets is computed. Shakhnarovich et al. [38] modeled a face sequence using a probability distribution. However, to make the computation tractable, they made the assumption that faces are normally distributed, which may not be true [57]. Cevikalp and Triggs [39] claimed that a face sequence is a set of points and they discovered a convex geometric region expanded by these points. The min-min approach [44, 46, 47] considered a face sequence as a cluster of points and measured the distance between these clusters. Subspace approaches [41–43] viewed a face sequence as points spread over a subspace. Although these methods can be highly accurate, a lot of computation is needed to represent the distribution of the face sequence, such as computing the convex hulls in [39], the probability models in [41], and the eigenvectors in [41–43]. For this reason, they are not scalable to large-scale video datasets.

Face datasets. In evaluating the performance of face matching approaches, most of the previous works on face retrieval in video use two benchmark datasets: MoBo (Motion of Body) [58] and Honda/UCSD [59]. The scales of these datasets

are limited, varying from hundreds to thousands of face images of tens of individual characters. Particularly, Honda/UCSD consists of 75 videos involving 20 individual characters. Each video contains approximately 300-500 frames. Meanwhile, Mobo provides 96 image sets of 24 individual characters. Hence, there are only 4 image sets for each character. One of the largest face datasets recently available is the YouTube Faces dataset [60], which provides 3,425 videos of 1,595 individual characters. However, each character has only around 2.15 videos. Such a small number of samples for each character is insufficient to stably evaluate a face matching or recognition approach, which is an important part of a face retrieval system. In addition, there are no face datasets related to real-world news videos, which is our targeted domain. In view of all the above mention considerations, we prepare new datasets for evaluating the approaches.

2.3 Framework Overview

Figure 4.1 illustrates the overview of our framework. In the off-line stage, the face-tracks in all video shots are extracted using our face-track extraction approach (described in Section 4). Each extracted face-track contains multiple face images of one individual character, varied under different viewpoints, illumination conditions, and expressions within a shot. Each single face image in a face-track is represented by a feature vector. The process consisting of face-track extraction and face image representation is performed once for the entire video dataset. Our main contribution here is making the face-track extraction approach robust against flash lights, scattered appearances of characters, and occlusions.

Given a face-track as an input retrieval query, the online stage of our system starts by using our proposed face-track matching algorithm (described in Section 5) to estimate the similarity between a query face-track and each face-track in the

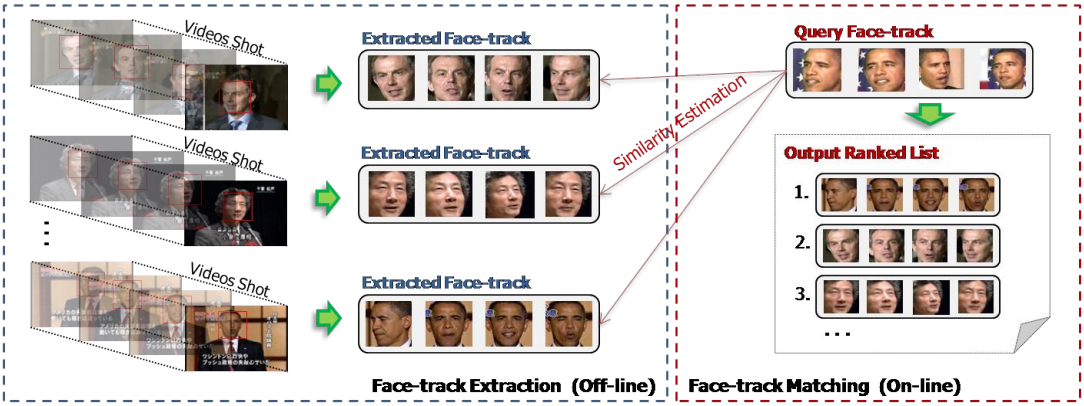


Figure 2.2: Framework Overview. In the off-line stage (left, blue box), face-tracks in videos are extracted using our proposed face-track extraction approach. This process is performed once for a video dataset. Then, our face-track matching approach estimates the similarity between a given query face-track and each face-track in the dataset to return a ranked list as the output of the online retrieval stage (right, red box).

retrieved set containing all face-tracks extracted from the dataset in the offline stage. A ranked list of the evaluated face-tracks is returned as the retrieval result of the online stage. Because the retrieved set is huge, our approach targets an extremely efficient face-track matching strategy while maintaining a competitive performance with state-of-the-art approaches.

2.4 Face-track Extraction

A common strategy in the existing approaches for face-track extraction consists of detecting faces in frames and grouping the detected faces of the same character. While detecting faces is done by using a standard face detector (e.g. Viola-Jones face detector) [44–46], grouping detected faces requires comprehensive techniques

to identify faces of the same character.

2.4.1 Face-track Extraction based on Tracking Points

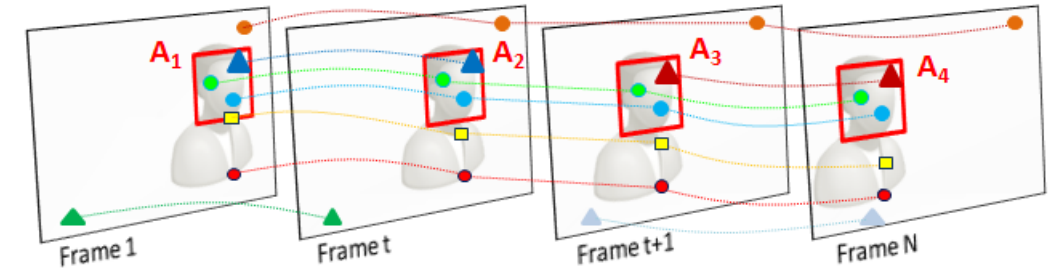
To group detected faces into face-tracks, connections should be established between faces belonging to the same character in different frames. A point tracker can be used for this purpose.

Assuming some points are generated and tracked through frames of a shot, we have the output of the tracking process as a set of tracking trajectories. One trajectory is for one generated point. We call such trajectories point tracks. Given two faces A and B in different frames and the set of point tracks, there are four types of point tracks regarding their intersection with the faces: (a) point tracks that pass through both A and B, (b) point tracks that pass through A but not B, (c) point tracks that pass through B but not A, and finally, (d) point tracks that do not pass through either A or B. A point track passes through a face if its point lies within the face bounding box in the corresponding frame.

A confidence grouping measure (CGM) that the two faces A and B belong to the same character can then be defined as:

$$CGM(A, B) = \frac{N_a}{N_b + N_c} \tag{2.1}$$

where N_a, N_b, and N_c are the number of tracks of types (a), (b), and (c). If $CGM(A, B)$ is larger or equal to a certain threshold, the two faces, A and B, are grouped into one face-track. Figure 2.3 presents an overview of a face-track extraction approach based on tracking points.



	(A_1, A_2)	(A_1, A_3)	(A_1, A_4)	(A_2, A_3)	(A_2, A_4)	(A_3, A_4)
Tracks of type (a)						
Tracks of type (b)						
Tracks of type (c)						
Tracks of type (d)						
CGM	3 / 1	2 / 3	2 / 3	2 / 2	2 / 2	3 / 0

Figure 2.3: An overview of a face-track extraction approach based on tracking points. Dashed lines connecting points in frames represent point tracks. One color is for one individual point track. Point tracks with circles points indicate tracks whose points are successfully tracked throughout the shot. A track whose point can not be tracked at a frame and replaced by a new point is denoted with triangle points. The point track with yellow squares demonstrates a track with tracking error. A neutral threshold 0.5 is used in comparing faces for grouping in this example. Thus, all detected faces are grouped into one face-track. The figure is best viewed in color.

2.4.2 Removal of Frames Containing Flash Lights

Although grouping faces based on a point tracker is efficient, applying the point tracker to news videos results in poor accuracy due to the occurrences of flash lights. The reason is that point trackers usually rely on intensity information to compute the image motion to find the correspondence between points in different frames. When flash lights occur in a frame, they significantly change the intensity

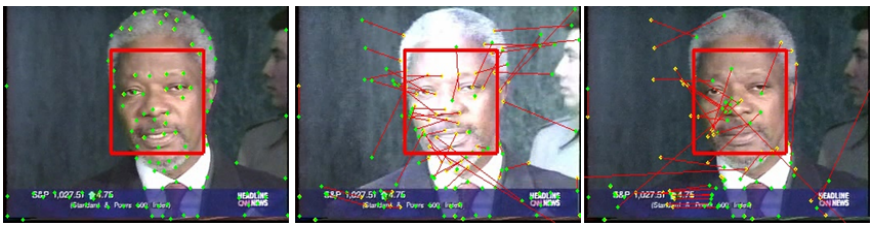


Figure 2.4: A real example of unreliable tracking results due to flash lights. Green points indicate track point positions in current frame. Red lines connecting green points and yellow points represent motions of points from the previous to the current frame.

of the frame. Thus, the tracker cannot track points properly (there is an example in Figure 2.4). To handle such situations that happen frequently in news videos, frames containing flash lights should be removed. We call such frames flash-frames.

To identify flash-frames, we measure the luminosity of the frames in the video shot. If the luminosity of a frame is significantly increased compared with its neighbors, the frame is declared to be a flash-frame. Particularly, given a set of consecutive frames S:

$$S = \{fr_s : s = \overline{t, t + \mathcal{W}}\} \tag{2.2}$$

where t is a frame index and $\mathcal{W}$ is the potential length of a flash light (i.e. the number of consecutive frames affected by a flash light). The frames in S are determined to be flash-frames if $\forall fr_s \in S$, we have:

$$\begin{cases} \mathcal{L}(fr_s) & > \gamma \mathcal{L}(fr_{t-1}) \\ \mathcal{L}(fr_s) & > \gamma \mathcal{L}(fr_{t+\mathcal{W}+1}) \end{cases} \tag{2.3}$$

given $\mathcal{L}(fr_x)$ is the computed luminosity of frame fr_x and γ is a predefined luminosity sensitive threshold. In our experiments, we found that $\gamma = 1.25$ and $\mathcal{W} = \{1, 2, 3\}$ are optimal to detect all flash-frames with a low false alarm rate.

By removing flash-frames, faces in these frames are also eliminated in grouping into face-tracks. However, such faces can not enrich information on their corresponding face-tracks but only add noise since their visual identity characteristics are often lost due to overlighting. And, two over-lightened faces of two characters may look very similar each other. Hence, eliminating them brings benefit to face-track matching.

2.4.3 Point Generation and Tracking

There are two main processing steps of a point tracker, which are point generation and point tracking. Once points are generated, they can be tracked through a sequence of frames. With a state-of-the-art point tracker such as the KLT tracker, points are generated by using an approach introduced by Shi and Tomasi [49]. The approach selects optimal points for tracking without any constraints on the positions of points. Then, points are tracked by computing optical flow between frames.

The requirement to use tracking results (i.e. point tracks) for grouping faces of the same character is that the faces must have some point tracks passing through all of them. There are two cases where the requirement is not met:

(i) Faces of new characters are detected in frames in which points are not generated or they are generated but not inside the faces (there is an illustration in Figure 2.5). To bypass this shortcoming, we generate track points for faces that are considered to be the faces of new characters. Faces of new characters in a certain frame are faces that cannot be grouped into any existing face-tracks.

In particular, we process a given shot frame-by-frame. Each face in the

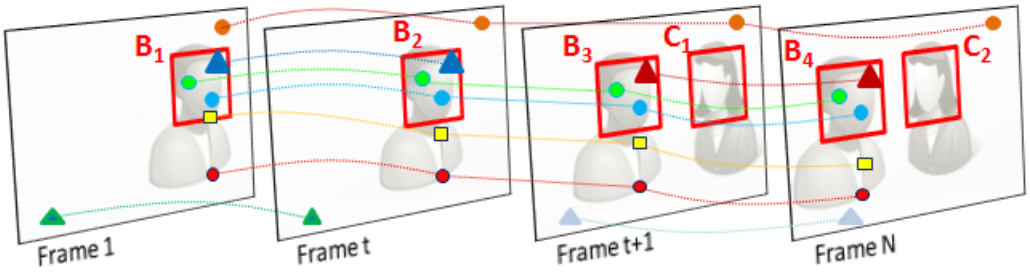


Figure 2.5: Illustration of a problem that occurs when generation of track points is independent of face detection results. Two faces C_1 and C_2 of character C in this example have no track points passing through them. Thus, they are considered to be two single-face face-tracks.

current frame of processing is checked against all existing face-tracks formed in the previous frames to find which face-track the face belongs to. Checking between a face and each face-track is based on computing the confidence grouping measure (CGM, presented in Subsection 4.1) between the face and the last face of the face-track. The face will then be grouped into a face-track whose CGM is largest and larger or equal to a certain threshold (0.5 in our experiments). A face having no CGM larger than the threshold is set as the initial face of a new face-track (i.e. a new face for a new character). Track points are then only generated inside the face bounding boxes.

(ii) Points are incorrectly tracked due to occlusions. Track points move from inside the face to outside it after occlusion. Furthermore, track points from background regions move inside the face. Thus, the number of point tracks passing through a face before occlusion and other faces after occlusion significantly declines, resulting in failed face grouping. Figure 2.6 shows an example of such problems.

We handle this problem by detecting and replacing incorrectly tracked points as well as not generating track points in background regions. When a face in the current frame is grouped into an existing face-track, we investigate all track points passing through the face or the last face of the face-track. All points whose tracks only pass through one of the two faces are removed. Because such points are likely to have been tracked inaccurately, removing them prevents us from transferring tracking errors to later frames. Then, additional points are generated to replace those that have been removed and to provide points updated with a new visual appearance of the face. We apply the approach of Shi and Tomasi [49] to the face bounding box to generate additional points. Points whose tracks pass through both the faces are kept.

2.4.4 Our Proposed Approach for Face-track Extraction

Given a video shot with all flash-frames removed, our approach starts by finding the first frame in which faces are detected. All point tracking and face grouping processes are initialized from this frame. This helps us to save computational cost and avoid tracking errors caused by transitional effects between shots. Initial track points will be generated inside all detected faces in the frame. Each face now becomes the first face of a corresponding newly formed face-track.

After initialization, we sequentially process each frame afterward until the end of the shot is reached. The pseudo-code is presented in Algorithm 1. Figure 2.7 has a step-by-step illustration of our approach.

Algorithm 1. *Our approach for face-track extraction*

Require: a video shot with identified flash-frames

Require: faces detected in frames

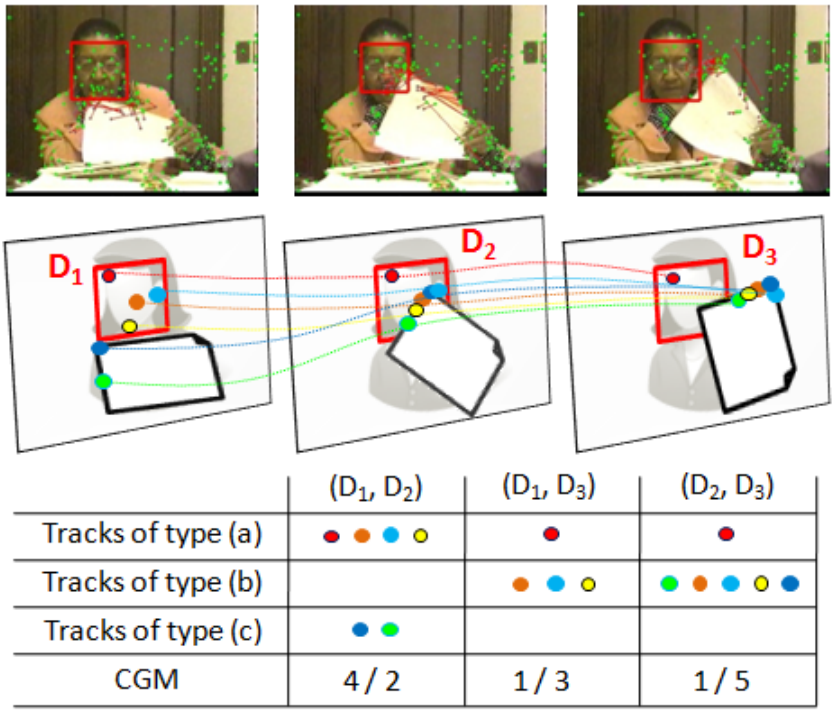


Figure 2.6: A real example with simplified illustration of tracking errors due to occlusion. Although all track points are retained in such cases, their tracks cannot help to connect face D_3 with other faces D_1 and D_2 of the same character, given the threshold of CGM for grouping two faces is 0.5.

1: Move to the first frame fr_t in which faces are detected.
2: Initialization
   - Create a new face-track with each detected face in fr_t.
   - Generate track points inside the detected faces.
3: **for** each frame fr_i from fr_t to the end of the shot **do**
4: **if** fr_i is not a flash-frame **then**
5: Use the KLT tracker to track existing points and update their positions in fr_i.
6: **for** each detected face f in fr_i **do**

7: **for** each existing face-track ft_j **do**

8: Compute CGM between the face f and the last face $f^l_{ft_j}$ of ft_j, $CGM(f, f^l_{ft_j})$.

9: **end for**

10: Find face-track ft_X so that $CGM(f, f^l_{ft_X}) \geq CGM(f, f^l_{ft_j}), \forall ft_j$ and $CGM(f, f^l_{ft_X}) \geq 0.5$.

11: **if** ft_X found **then**

12: Group f to ft_X.

13: Remove incorrectly tracked points and generate additional points for f.

14: **else**

15: Create a new face-track with f.

16: Generate new points for f.

17: **end if**

18: **end for**

19: **end if**

20: **end for**

Note that our approach does not compare all possible pairs of faces for face grouping. Such pairwise comparison rapidly becomes intractable as the number of faces in a shot increases. Instead, we group faces into face-tracks according to the temporal order of their appearance. A detected face in a frame is only compared to the last faces of existing face-tracks. By doing this, we avoid greedy pair-wise comparisons.

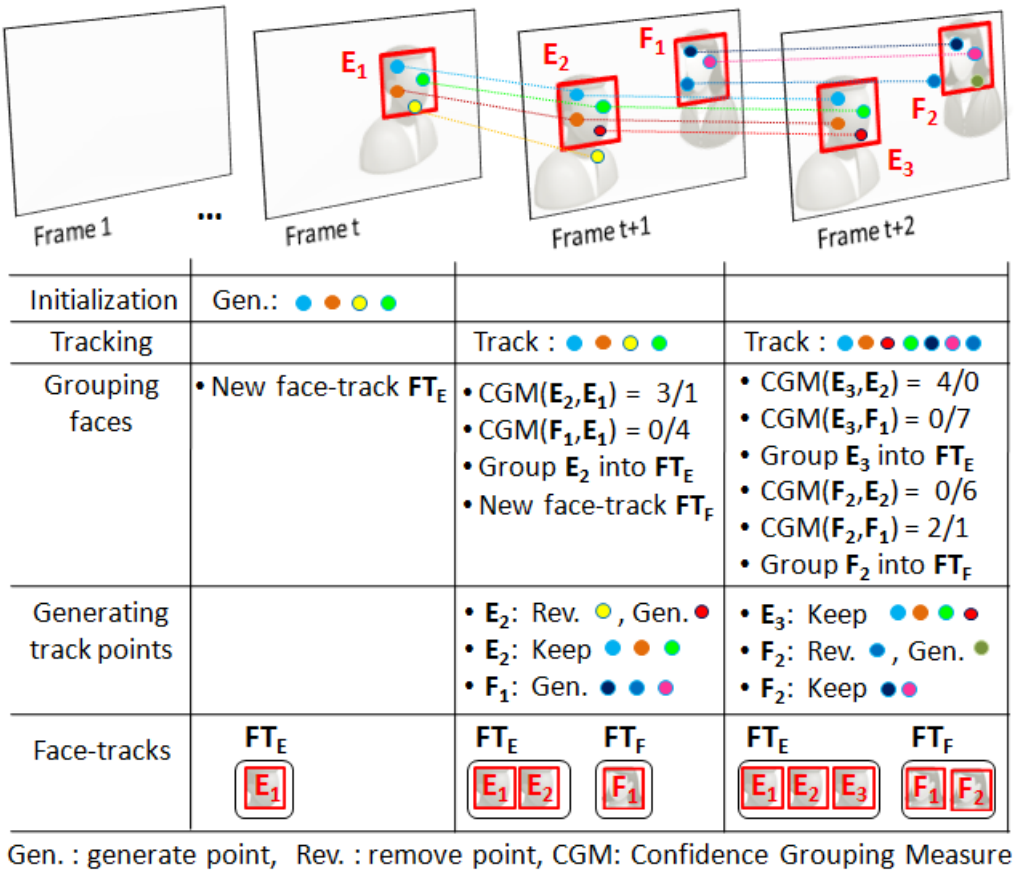


Figure 2.7: A step-by-step illustration of our approach for face-track extraction.

2.5 Matching Face-tracks

Several approaches for matching face-tracks have been proposed (as presented in Section 2). However, although these approaches have shown high accuracy in benchmark datasets, their high computational costs limit their practical applications in large-scale datasets. This motivates us to target a matching approach that provides a good balance between accuracy and computational cost. The approach should be extremely efficient while achieving a competitive performance with state-of-the-art approaches.

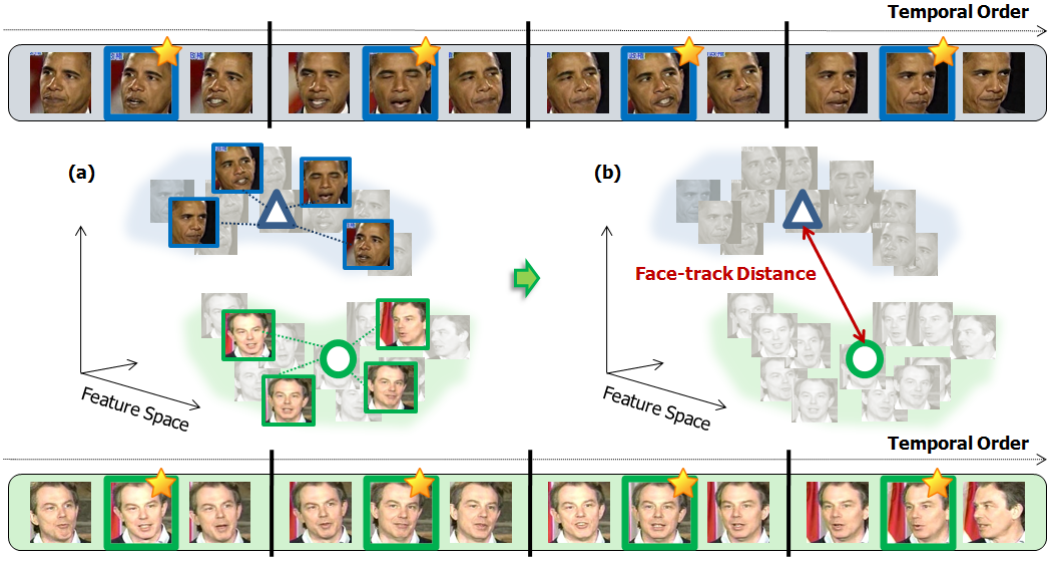


Figure 2.8: An illustration of our proposed k-Faces approach. In (a), each face–track is first divided into k equal parts ($k = 4$, in this example). The middle face (with bright-colored outline) is selected in each part to represent the part. Then, k selected faces (marked with stars) are used to compute a mean face (circle or triangle) on the feature space. The mean face now represents the whole face-track. Finally, in (b), the similarity of face-tracks is estimated based on the distance between their mean faces.

To maintain competitive accuracy, we still use the plenteous information from the multiple faces of a face-track to enrich the representation. However, instead of using all the faces in a face-track, we propose taking a subsample of the faces. In doing so, the required computational cost can be reduced while keeping the amount of information sufficient to improve accuracy. We call our approach k-Faces.

Based on an observation that faces in neighboring frames of a character are not dramatically changed, we propose to subsample faces of a face-track regarding

their temporal order of appearance. The neighboring range is controlled by a variable, k. With a given value of k, our approach starts by temporally dividing the face-track into k equal parts. The middle face for each part is selected to represent all faces within the part since we assume that faces within such a part are barely similar each other when k is sufficiently large.

Given k faces subsampled from the face-track and each face has been extracted its own facial feature, the face-track now is corresponding to a set of k points distributed in the feature space. We employ the mean point to represent the set. The distance between two sets now relies on the distance between their mean points. In other words, if the mean point is called a mean face, the similarity between two face-tracks corresponds to the distance between their mean faces. Figure 2.8 illustrates our k-Faces approach.

We have two main advantages by using a single mean face of k subsampled faces to represent a face-track. First, the computational cost on estimating similarities between face-tracks is low, since only one mean face is used for one face-track. Second, if there are noisy faces in a face-track and they are not a majority, subsampling faces helps us to reduce the number of noisy faces actually involved in processing. Our approach is therefore less sensitive to noisy faces than other approaches such as those employing all faces of the face-track to compute the mean face or those estimating the similarity between two face-tracks based on the pair-wise distances of their faces.

Let $m^A = \{m_1^A, m_2^A, .., m_N^A\}$ and $m^B = \{m_1^B, m_2^B, .., m_N^B\}$ denote the mean faces of face-tracks A and B, respectively, and N represents the number of dimensions of feature space. We use the following standard distance types to compute

the distance between m^A and m^B.

$$Cosine: \qquad distance = 1 - \frac{\sum_{i=1}^{N} m_i^A \times m_i^B}{\sqrt{\sum_{i=1}^{N}(m_i^A)^2} \times \sqrt{\sum_{i=1}^{N}(m_i^B)^2}} \tag{2.4}$$

$$Euclidean: \qquad distance = \sqrt{\sum_{i=1}^{N}(m_i^A - m_i^B)^2} \tag{2.5}$$

$$L1: \qquad distance = \sum_{i=1}^{N} |m_i^A - m_i^B| \tag{2.6}$$

The pseudo-code for our k-Faces is presented below.

Algorithm 2. *The proposed k-Faces approach*

Require: Two face-tracks A, B and a predefined value of k

1: **for** each face-track A, B **do**
2: Divide the face-track into k equal parts according to the temporal order.
3: In each divided part, select the middle face.
4: Compute the mean face (on the feature space) of the k selected faces.
5: **end for**
6: Compute the distance between the mean faces of A and B.
7: **return** the computed distance.

Clearly, the higher the value of the k set, the more faces in each face-track selected to compute the mean face and the better the approximations, which may result in improved accuracies. However, note that the computational cost can greatly increase. By using k as a predefined parameter, k-Faces provides users with flexibility in balancing the accuracy that they expect and the cost that they can afford (or the time they can spend waiting for the result). The question of selecting reasonable values for k on a dataset is presented in Subsection 6.2.4.

2.6 Experiments

In this section, we present our experiments to evaluate the proposed approaches. The experiments are divided into two parts: the first evaluates the performance of the proposed approach in face-track extraction, and the second in face-track matching.

2.6.1 Evaluation of Face-track Extraction

We tested our proposed approach for face-track extraction on 8 video sequences from different video broadcasting stations including NHK News 7, ABC News, and CNN News. All shot boundaries are provided in advance. A face detector based on the Viola-Jones approach [33] is used to detect near frontal faces in every frame of the video sequences. A conservative threshold is used to reduce the number of false positives (i.e. a non-face classified as a face). In particular, we only keep detected faces which are larger than 60×60, and the number of neighbor rectangles that makes up a candidate face must be greater than 4.

Ground-truth annotation on the face-tracks in the videos is manually prepared. Each face-track of a character appearing in a video shot is annotated by indicating all detected faces of the character. An extracted face-track is regarded as correct if it contains all faces of the face-track compared to ground-truth annotation. If the face-track has more or less faces than in the annotation, it is said incorrectly extracted. Note that if a character moves out of the frame and then moves back into it again, annotators will divide the appearance of that character into two independent face-tracks in ground-truth annotation. Table 1 summarizes the number of frames, faces, and face tracks.

We directly compare our approach with the state-of-the-art approach pro-

Table 2.1: Detailed information on the videos used in our experiment. FT-F indicates face-tracks having flash frames. FT-OM indicates face-tracks containing occlusions or that do not appear at the beginning of the shot. O stands for occlusion and, M for middle.

Videos	#frames	#faces	#FT	#FT-F	#FT-OM
NHK_0507	36,284	13,791	58	10	1
ABC_061098	20,374	11,080	51	1	3
ABC_020698	53,441	15,200	147	6	10
ABC_020898	53,610	13,968	140	14	21
ABC_022098	52,740	15,489	122	5	6
CNN_062698	19,363	6,712	56	3	12
CNN_020498	52,584	14,242	98	4	7
CNN_021898	52,448	18,399	83	0	8
Total	340,884	108,881	755	43	68

posed by Everingham et al. [44] in this experiment. Their approach generates track points in the first frame of the shot and tracks them throughout the shot based on local appearance matching. Points that cannot be tracked from one frame to the next are eliminated and replaced with new points. Their face grouping criteria is similar to that presented in Subsection 4.1. The threshold of CGM for grouping two faces is also 0.5.

As shown in Table 2, by detecting flash-frames, our approach successfully overcomes the problem of face-track fragmentation due to flash lights. Meanwhile, the approach by Everingham et al. almost completely fails to do that. In addition, the results also show that our approach is superior to that of Everingham et

Table 2.2: Performance of the evaluated approaches.

Approaches	#extracted FT-F	#extracted FT-OM	#total extracted FT
Everingham et al.	4/43	9/68	613/755 (81.19%)
Ours.	43/43	60/68	711/755 (94.17%)

Table 2.3: Processing time of the evaluated approaches.

Approaches	#frames	#Processing time (seconds)	Speed (frm/snd)
Everingham et al.	340,844	39,633.2	8.6
Ours.	340,844	89,696.8	3.8

al. in handling problems caused by partial occlusion and the appearance of a character in the middle of a shot. The only face-tracks that we could not extract exactly are those fully occluded in some frames during their occurrences. In those cases, all points in the face regions are *drifted* to the background region. After such full occlusions, there is no clue to re-grouping the face of that character. Using only a tracker is not enough to handle this problem. One can apply visual information-based clustering to group the fragmented face-track, as in [45], but this obviously require extra cost. Nevertheless, we observe that full occlusion rarely happens in news video because the characters featured in the news are recorded with care, especially the important and well-known ones. This is a special characteristics of news videos. The last column of the table shows the overall extraction performance of both approaches. These facts clearly indicate that our approach is robust and outperforms that of Everingham et al. in [44].

In terms of speed, our approach is approximately 2 times slower than that of Everingham et al. However, our complexity is somehow linear to the total

number of faces because we consequently enlarge face-tracks according to the temporal order by checking new faces with only the last appeared face of each face-track. Meanwhile, Everingham et al. compared all pairs of faces in the shot. Their complexity is polynomial to the total number of faces. If the number of faces increases, the gap in speed between our approach that by Everingham et al. will narrow rapidly.

In this experiment, we show that our proposed techniques and solutions to the problems are robust and efficient enough for extracting face-tracks in real-world news videos by successfully extracting 94% of all face-tracks. Based on our observations, other complex techniques can be applied to handle the problems. However, the trade-off between obtaining the 6% remaining face-tracks and incurring an overly high computational cost should be considered with care.

2.6.2 Evaluation of Face-track Matching

2.6.2.1 Datasets

Due to the limitations of existing public datasets, we prepared new datasets for the experiments. Face-tracks are extracted from videos of the datasets by using our proposed approach to face-track extraction (see section 4.2). The identity of the character associated with each extracted face-track is given by annotators. Because our approach extracts face-tracks in each video shot, we used a robust shot boundary detector to obtain shot boundaries for videos. The whole process, including shot boundary detection and face-track extraction, is fully automated.

TRECVID dataset. We used TRECVID news videos from 2004 to 2006. This dataset contains 370 hours of videos in different languages, such as English, Chinese, and Arabic. The total number of frames that we processed was approx-

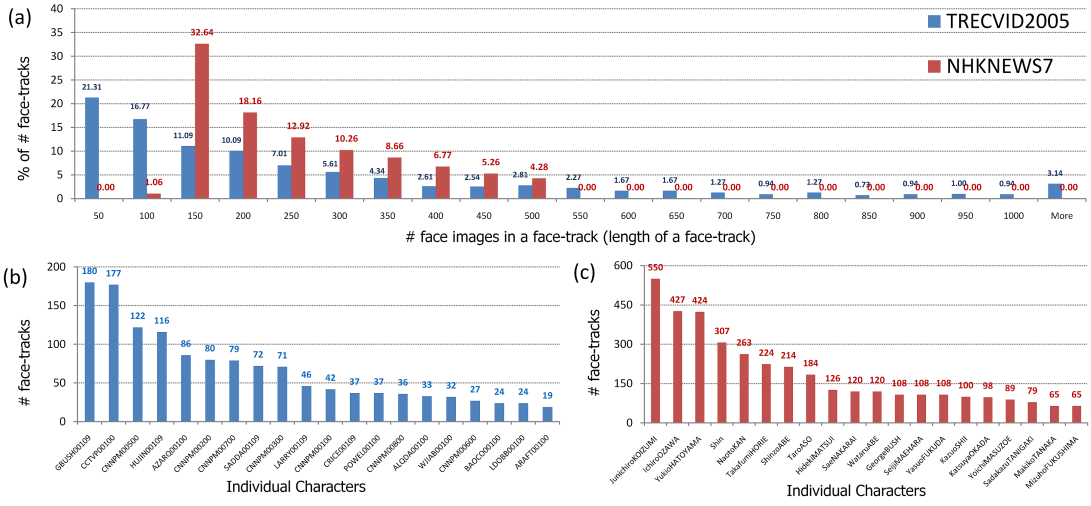


Figure 2.9: Statistical information on our datasets. (a) shows the distribution of face-tracks over their lengths; (b) and (c) present the number of face-tracks for the top 20 individual characters in each dataset.

imately 35 million. Among those, 20 million faces were grouped into 157,524 face-tracks. We filtered out short face-tracks that had less than 10 faces, which resulted in 35,836 face-tracks. Finally, we annotated 1,497 face-tracks containing 405,887 faces of 41 well-known individual characters.

NHKNews7 dataset. This dataset consists of observations from the NHK News 7 program over 11 years. After the annotation process, 1,259,320 faces of 111 individuals are provided. The total number of face-tracks is 5,567. Each character has from 4 to 550 face-tracks. In this dataset, we discard face-tracks with fewer than 100 faces and more than 500 faces. Compared to the TRECVID dataset, the NHKNews7 dataset is much more challenging.

Table 2.4 shows a comparison between our datasets and some public benchmark datasets. Based on the results, it is obvious that our datasets are superior over the other datasets, such as MoBo and Honda/UCSD, on all statistical terms,

including number of videos, number of characters, and average face-track length. Buffy dataset [61], a yet another popular dataset, is also smaller than ours. Although they have more face-tracks for each character, their face-tracks are rather small. The number of face-tracks having less than 10 faces is 374, approximately 47% of the dataset. Compared to the YouTube Faces dataset, we provide much more face-tracks (or video shots) per character. Thus, our datasets are more relevant for stably evaluating a face retrieval system.

Figure 2.9 presents statistical information on our datasets. The datasets can be downloaded at the following link [1]. However, due to copyright issues, the face images in face-tracks can not be published. Instead, we provide a feature vector, used in [44], for each face image. The feature vector of a face is extracted by computing the descriptors of the local appearance of the face around each of the located facial features. Before extracting the descriptors, the face is geometrically normalized to reduce the effect of pose variation. An affine transformation is estimated, which transforms the located facial feature points to a canonical set of feature positions. Then, the appearance descriptors around each facial feature are computed. The final feature representation of the face is formed by concatenating all the descriptors of its facial features. Regarding to our experiments in [62] and this work with the TRECVID dataset, this feature is better than local binary pattern (LBP) feature for face-track matching.

2.6.2.2 Evaluated approaches

We compared k-Faces with several approaches, including those based on pair-wise distances, MSM [42], and CMSM [43].

Given two face-tracks having multiple face images represented as feature vec-

[1] http://satoh-lab.ex.nii.ac.jp/users/ndthanh/NIIFacetrackDatasets

Table 2.4: Statistical comparison between our datasets and other benchmark datasets.

Datasets	#Face-tracks (Ftk.)	#Characters (Chr.)	#Faces per Ftk.	#Ftk. per Chr.
MoBo	96	24	300	4.00
Honda/UCSD	75	20	300-500	3.75
Buffy[61]	802	11	1-392	**72.9**
YoutubeFaces	**3,425**	1,595	48-6,075	2.15
TRECVID	**1,497**	41	10-3,781	**36.51**
NHKNews7	**5,567**	111	100-500	**50.15**

tors, pair-wise based approaches compute the distances between each possible pair of feature vectors in two face-tracks. The the maximum distance, the minimum distance, or the mean distance of the computed pair-wise distances is then used as the similarity measurement between two face-tracks. We refer to the approaches as *pair.max*, *pair.min*, and *pair.mean* , respectively (see Figure 2.10 for the illustration). The *pair.min* (sometimes called min-min) is one of the state-of-the-art approaches widely used in other studies [44, 47, 60, 63].

In addition, we also evaluate an yet another approach that is a hybrid of *pair.min* and our proposed approach. The approach starts by dividing each face-track into k equal parts according to the temporal order and selecting the middle face of each part. Given k selected faces for each face-track, the *pair.min* approach using only these selected faces is then applied to estimate the similarity between two face-tracks. We call this approach *k-pair.min*. With k is not larger than the number of faces in the face-track, by using only k faces in each face-track, this approach is more efficient than the original *pair.min*. When k is getting

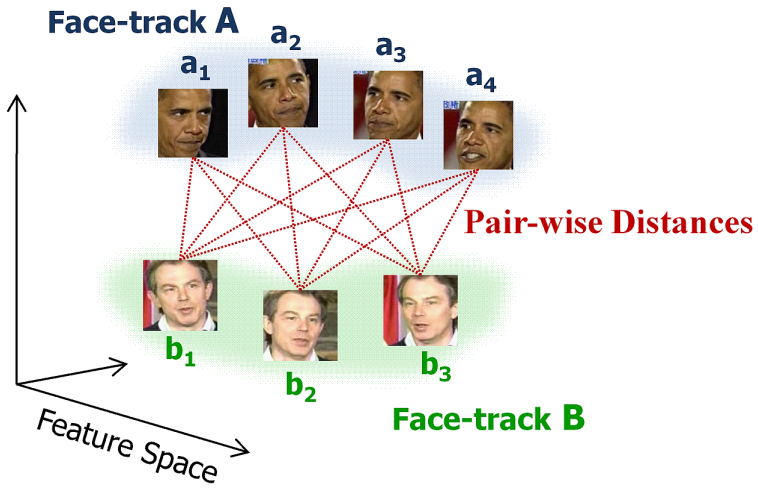


Figure 2.10: Pair-wise based approaches. Based on the possible pair-wise distances of faces in face-tracks A and B, we have: $pair.min(A,B) = \min dist(a_i, b_j)$, $pair.max(A,B) = \max dist(a_i, b_j)$, and $pair.mean = M^{-1}N^{-1}\sum_i \sum_j dist(a_i, b_j)$, where $i = \overline{1,M}$, $j = \overline{1,N}$. Knowing that a_i and b_j are feature vectors, the function $dist(a_i, b_j)$ is used to compute their distance on the feature space. Different types of distances are used for $dist$ such as L1 distance, Euclidean distance, and Cosine distance. In this illustration, $M = |A| = 4$ and $N = |B| = 3$.

larger, the performance of this approach definitely approximates the performance of *pair.min*. However, because *k-pair.min* still requires k^2 comparisons between the selected faces of two face-tracks, it is theoretically k^2 more expensive than our k-Faces approach. And, *k-pair.min* is also more sensitive to noisy faces than our k-Faces due to the fact that it relies on pair-wise distances between faces of the face-track.

Regarding [64], if the pair-wise based approaches are representative of non-parametric sample based approaches, MSM and CMSM are representative of approaches based on a parametric model. MSM, introduced by Yamaguchi et

al. [42], represents an image set by a linear subspace spanned by the principal components of the images. The similarity between the sets is computed using the angle between subspaces. CMSM is an extension of MSM, in which subspaces of the sets are projected onto a constraint subspace. In doing so, the subspaces are expected to be more separable. All of these approaches have been shown their robustness in benchmark datasets, such as MoBo [58], HondaUCSD [59], and YouTube Faces [60]. Therefore, it is appealing to compare our k-Faces with them for a comprehensive evaluation.

Besides evaluating k-Faces with different values of k and different types of distance (e.g. *Euclidean*, $L1$, and *cosine*), we try another criterion for selecting k representative faces in a face-track. Instead of temporally dividing the face-track and choosing the middle face of each part, another criterion that is based on clustering can be applied in selecting these representative faces. In this new way, all the faces in a face-track will be clustered to k groups by using a clustering algorithm. The centroid of each group is selected. Then, the mean of k centroids is used as the representative face for the face-track. In this experiment, we use the standard K-Means for clustering. We refer to the former k-Faces as k-Faces.$Temporal$ and to the latter k-Faces as k-Faces.$KMeans$.

We evaluate the performance of a face-track matching approach by computing the average precision of the rank list that it returned. In particular, in each dataset, a face-track is alternatively picked out as a query face-track while the remaining face-tracks are used as the retrieved database. Given a query, the average precision of the returned ranked list is computed. Finally, the mean of all average precision (MAP) values for all queries is reported as the overall evaluation metric for the approach with the given database.

Let r denote a rank in the returned face-track list, $Pre(r)$ the precision at rank

r of the list, $\mathcal{N}_l$ the length of the list, $\mathcal{N}_{hit}$ the total number of face-tracks matched with the query face-track q, and $IsMatched(k)$ a binary function returning 1 if the face-track at rank r is matched with q (based on ground-truth annotations) and, zero otherwise. Then, the MAP of the evaluated approach can be computed as follows:

$$AP(q) = \frac{\sum_{r=1}^{\mathcal{N}_l}(Pre(r) \times IsMatched(r))}{\mathcal{N}_{hit}} \tag{2.7}$$

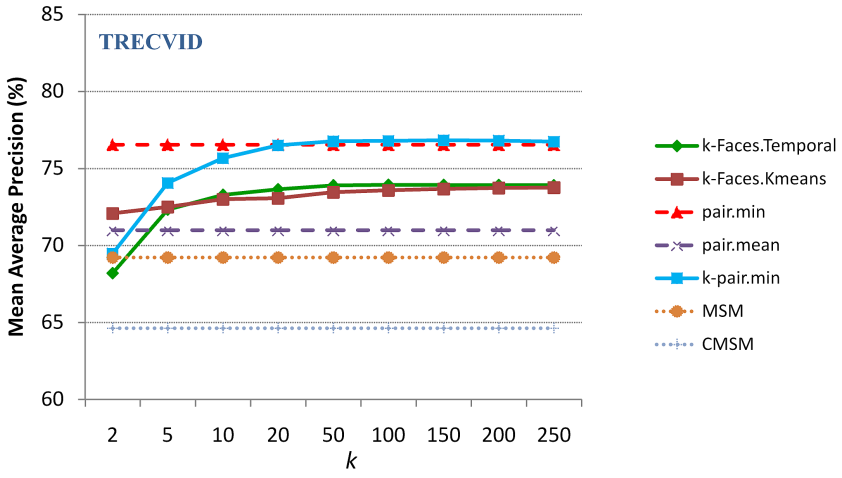
$$MAP = \frac{\sum_{q} AP(q)}{\text{number of queries}} \tag{2.8}$$

MAP is a standard metric for evaluating retrieval and matching systems. Besides the MAP, we record the processing times of the approaches in each dataset to compare their efficiency.

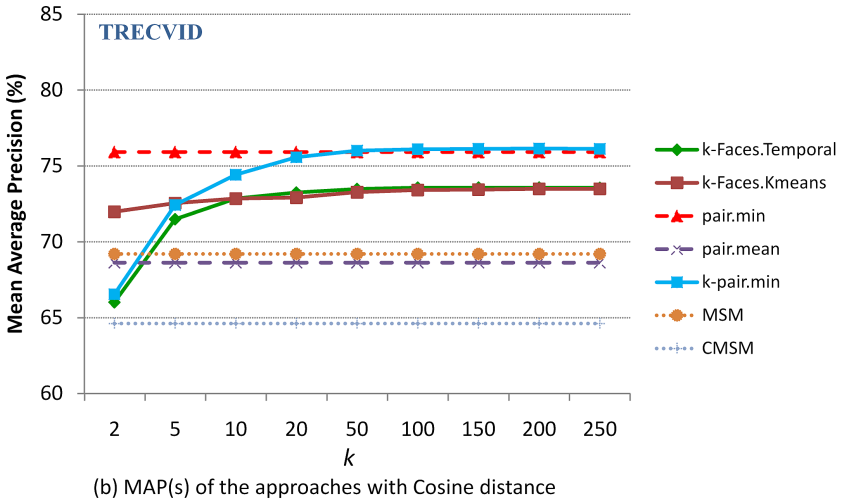
2.6.2.3 Results

Figure 2.11 and Figure 2.12 present the mean average precision (MAP) of the evaluated approaches in our two datasets, TRECVID and NHKNews7. With MSM and CMSM, their best performances over various sets of parameters are selected.

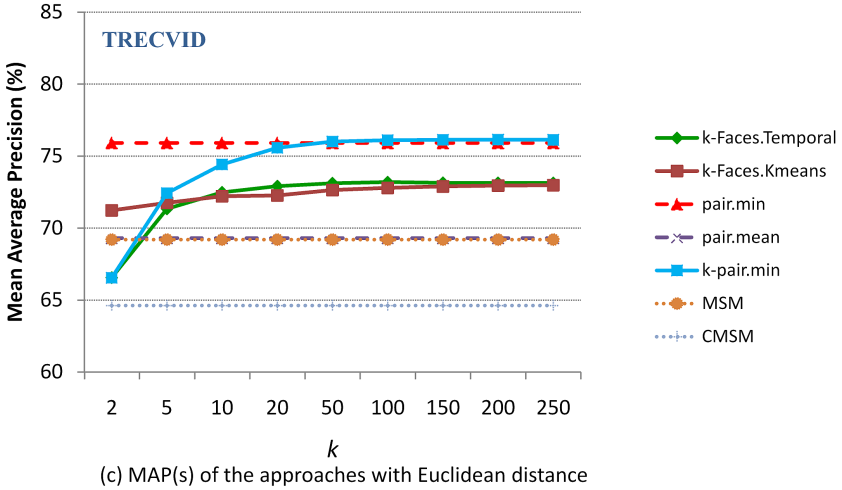
In general, all the MAPs vary from 64.61% to 76.82% in the TRECVID dataset. Meanwhile, in the NHKNews7 dataset, the best MAP is 60.99%, and the worst is 40.89%. The difference in the MAPs between the two datasets can be explained by following reasons. First, the number of characters in NHKNews7 is larger than that in TRECVID, 111 characters in NHKNews7 compared to 41 characters in TRECVID. This clearly increases the probability of mis-matching face-tracks. Second, the videos in NHKNews7 were recorded over a long time (i.e. 11 years). Thus, besides facial variations in each face-track caused by the environ-



(a) MAP(s) of the approaches with L1 distance



(b) MAP(s) of the approaches with Cosine distance



(c) MAP(s) of the approaches with Euclidean distance

Figure 2.11: MAP(s) of the approaches in the TRECVID dataset.

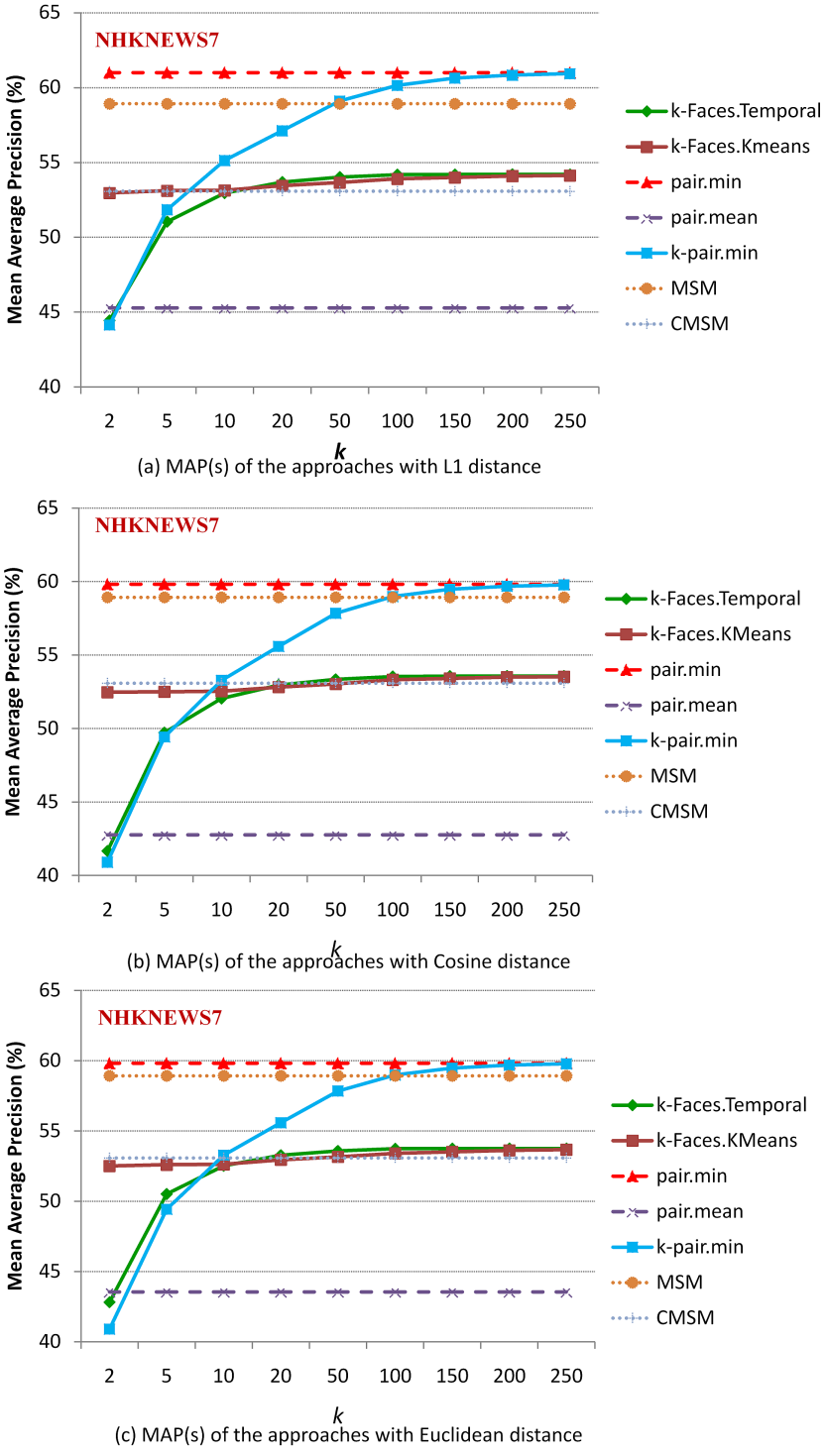


Figure 2.12: MAP(s) of the approaches in the NHKNews7 dataset.

Figure 2.13: Face-tracks of President George W. Bush recorded in 2001 (top) and 2009 (bottom).

mental conditions at the time of recording (e.g. illumination, pose, viewpoint), the face-tracks of the character themseves also reflect the biological variations of the character over time: for instance, a character may look older after several years (see Figure 2.13 for example). For these reasons, matching faces in NHKNews7 becomes more challenging, which resulted in decreased MAP(s) for all the evaluated appoaches.

A clear and consistent observation from both datasets is that *pair.min* (i.e. min-min) is always among the two best approaches. It achieved 76.54% MAP and 60.99% MAP in the two datasets, respectively. Among the distance types, $L1$ is the optimal for use with *pair.min*. A reasonable replacement is the *Euclidean* distance. However, there is still a minor accuracy gap between *pair.min* using $L1$ and *pair.min* using *Euclidean* distance. In addition, computing the *Euclidean* distance between two feature vectors is more expensive than computing their $L1$ distance.

The results also show that *pair.min* is better than *pair.mean*. This is because *pair.mean* uses the mean of all pair-wise distances between two face-tracks as the

similarity score. By computing the mean, *pair.mean* reduces the effect of noisy pairs. At the same time, it eliminates the influence of pairs containing identical faces, which can help to instantly determine that the faces are belong to the same character. Thus, the discriminative power of the the computed similarity score is reduced, compared that computed by *pair.min*. This causes the difference in MAPs between *pair.min* and *pair.mean*. More generally, this explains why such a gap between *pair.min* and *pair.mean* is larger in NHKNews7 than in TRECVID. Because the average length of face-tracks on NHKNews7 is longer (i.e. each face-track contains more faces of a character), there is a greater chance that two face-tracks of the same character contain identical faces. Among the approaches based on pair-wise distances, *pair.max* achieved the worst performance. Its best MAPs with $L1$ distance in the TRECVID and the NHKNews7 datasets are 28.16% and 14.99%, respectively. We do not include *pair.max* in the Figure 2.11 and Figure 2.12.

Regarding our k-Faces, its MAP increases when k increases. Between k-Faces.$Temporal$ and k-Faces. $KMeans$, the impact of k on the MAP of k-Faces.$KMeans$ is less significant. Because k-Faces.$KMeans$ always uses all the faces in a face-track for clustering and selecting centroids for representative faces, the final mean face is less sensitive to k. In contrast, k plays an important role in k-Faces.$Temporal$. The higher the k set, the more representative faces of each face-track selected. Thus, the final mean face of each face-track becomes more reliable and accurate. The advantages of k-Faces.$KMeans$ is that it can achieve high accuracy even when k is very small. However, its disadvantage is the high computational cost of clustering faces on a high-dimensional feature space (i.e. 1,937 dimensions). When k is large enough, there is no substantial difference in MAP between k-Faces.$KMeans$ and k-Faces.$Temporal$.

Table 2.5: Mean Average Precision and processing times (in seconds) of the evaluated approaches. Note that the preprocessing process is only performed once for a given dataset. And, k of the approaches in this table is equal to 20.

Approaches	**TRECVID**			**NHKNews7**		
	MAP(%)	**Preprocessing Time**	**Matching Time**	**MAP(%)**	**Preprocessing Time**	**Matching Time**
$pair.min$+$L1$	76.54	0.00	2544.73	60.99	0.00	6678.00
$k-pair.min$+$L1$	76.50	0.00	418.24	57.11	0.00	972.36
k-Faces.$Temporal$+$L1$	73.65	26.54	1.63	53.68	41.4	3.23
k-Faces.$KMeans$+$L1$	73.07	6380.50	1.20	53.45	14949.12	3.45
MSM	69.20	4454.10	347.39	58.92	4896.72	667.15
CMSM	64.62	4991.02	95.36	53.08	5841.80	155.40

In both datasets, when k increases from 2 to 20, the MAPs of k-Faces approaches grow rapidly. However, the MAPs become stable from $k = 20$ upward. Because further increasing k does not help improve accuracy but increases the computational cost, we select $k = 20$ for investigating the trade-off between the accuracy and computational cost of k-Faces approaches in comparison to others.

As we can see from Figure 2.11 and Figure 2.12, the hybrid approach *k-pair.min* provides more competitive performance with the *pair.min* approach than our proposed k-Faces. The performance of *k-pair.min* with most values

of k is between those of *pair.min* and k-Faces. When k is sufficiently large (20 in TRECVID, and 150 in NHKNews7), *k-pair.min* approximates *pair.min*. In some cases, *k-pair.min* is even better than *pair.min*, e.g., when k is larger than 50 in the TRECVID dataset. However, the gap is very small varying from 0.2% to 0.25%. This is because there are noisy faces (e.g. blurred faces) in a few face-tracks, and *k-pair.min* is less sensitive to noise than *pair.min* due to its subsampling process. Although *k-pair.min* is more accurate than our k-Faces, we need to be reminded that it theoretically requires k^2 times more comparisons than ours. Their practical efficiency is summarized in Table 2.5.

Table 2.5 shows MAP and processing time of each approach. Processing time is divided into two parts, preprocessing and matching. The preprocessing time refers to the time required to preprocess face-tracks in a given dataset before matching. In k-Faces approaches, the preprocessing of face-tracks includes selecting representative faces and computing their mean face. In MSM and CMSM, preprocessing includes computing subspaces for face-tracks. The matching time is averaged over one query run. The time unit used is seconds.

As shown in Table 2.5, k-Faces.$KMeans$ and k-Faces.$Temporal$ achieve almost equal accuracy and consume the same amount of time for one query in both datasets. However, k-Faces.$Temporal$ is hundreds of times (240 times in TRECVID and 360 times in NHKNews7) faster than k-Faces.$KMeans$ in the preprocessing phase. This suggests that in terms of both accuracy and efficiency, selecting representative faces based on temporal sampling is better than that based on clustering.

Compared to state-of-the-art approaches, our k-Faces.$Temporal$ is thousands of times faster than *pair.min*, and hundreds of times faster than MSM, CMSM and $k-$*pair.min* in both datasets. In terms of accuracy, k-Faces takes third place,

with 73.65% in the TRECVID dataset, after *pair.min* and *k-pair.min*. The difference in MAP between our approach and *pair.min* is only 2.89%. Meanwhile, k-Faces.$Temporal$ is significantly better than MSM and CMSM, which respectively achieved 69.20% and 64.62% accuracy. In NHKNews7 dataset, k-Faces.$Temporal$ is better than CMSM, but worse than *pair.min*, *k-pair.min*, and MSM. One may question why MSM performed poorly in the TRECVID dataset, but was superior to k-Faces.$Temporal$ in the NHKNews7. The reason for this is the fact that the face-tracks in the NHKNews7 dataset are larger than those in the TRECVID dataset. Therefore, more sample faces in each face-track can be used to obtain a reliable subspace.

The results obtained from this experiment generally indicate that our proposed approach is extremely efficient while achieving performance comparable with that of state-of-the-art approaches.

2.6.2.4 Discussion

This subsection discusses two main arguments on using k-Faces approaches :

- Why the k-Faces.$Temporal$ has comparable accuracy to k-Faces.$Kmeans$ and how to select a reasonable value for k given a dataset.
- How accurate the performance of k-Faces approaches (k-Faces.$Temporal$ and k-Faces.$KMeans$) are compared to other approaches on benchmark and public video datasets.

The key idea behind both k-Faces.$Temporal$ and k-Faces.$Kmeans$ to achieve reasonable accuracy while maintaining efficient retrieval speed is to only use a subset of faces among all available faces in each face-track. To do that, k-Faces.$Temporal$ is based on the assumption that the faces of a character appear-

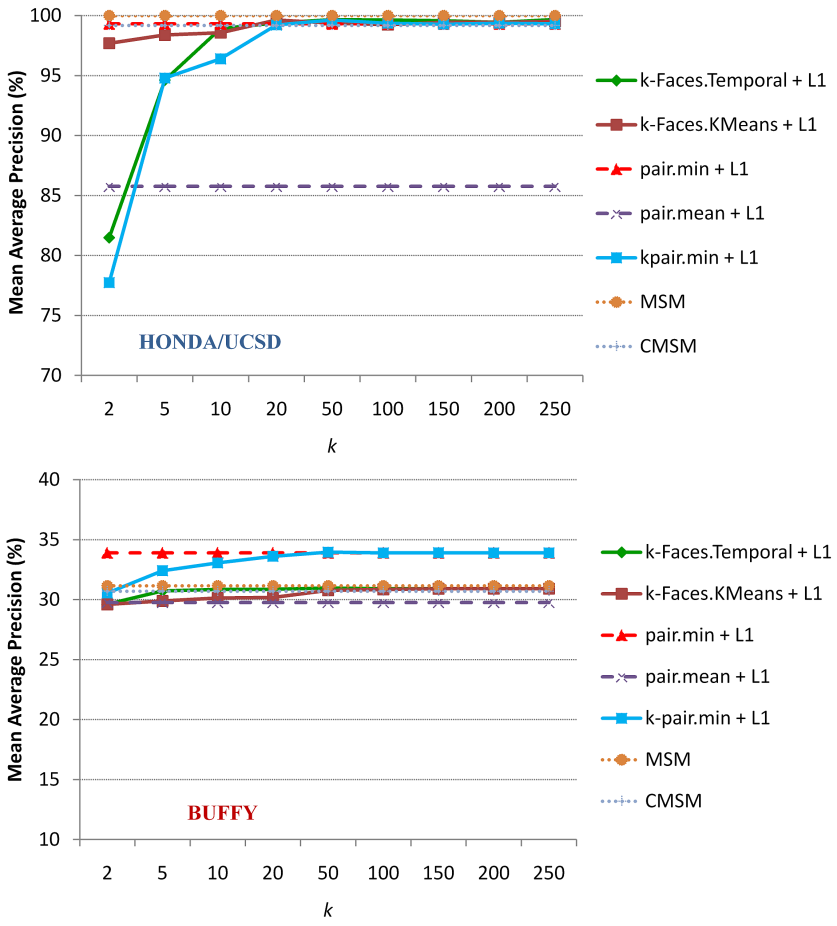


Figure 2.14: MAP(s) of the evaluated approaches in Honda/UCSD (left) and Buffy (right) datasets.

ing in neighboring frames are visually similar. Thus, one of them can be used as being representative. Meanwhile, k-Faces.$KMeans$ relies on clustering to find clusters of similar faces directly in the feature space. Faces within a cluster are considered to be neighboring in the space. Hence, the center of each cluster can be used to represent the cluster.

In both approaches, the variable k controls the range of neighboring. The larger the value of k that is selected, the smaller the range is. In other words,

the number of faces in a part temporally divided by k-Faces.$Temporal$ and the number of faces in a cluster in k-Faces.$KMeans$ are smaller. This means the representative face for each part (or cluster) selected by both approaches becomes less different with other faces within the part (or cluster). In addition, more faces of a face-track are involved in computing the single mean face (i.e. the mean of k selected faces, or the mean of k centroids) as k is increasing to represent the whole face-track. As the two sets of actual faces used by both approaches gradually overlap, the means of the sets become more similar. All of these reasons explain why k-Faces.$Temporal$ has comparable accuracy to k-Faces.$Kmeans$, given sufficiently large k.

To evaluate the *representativeness* of the mean faces computed by both approaches, we compute the mean distance from each of the mean faces to all other faces of a face-track. Smaller distance indicates better *representativeness* since the mean face is more similar to the other faces within the face-track. The mean distance is used instead of the sum of distances since the numbers of faces in face-tracks are very different. We report the average of such mean distances over all face-tracks (i.e. average *representativeness*) with different values of k in Figure 2.15.

When k is sufficiently large (e.g. by being larger or equal to 20) in Figure 2.15, the mean faces computed by both k-Faces.$Temporal$ and k-Faces.$KMeans$ have equal *representativeness*. This observation is also consistent with what we learned from previous experiments in which we evaluate the accuracy of the approaches with different values of k. In general, the larger k is, the better the *representativeness* of the mean faces and the higher the accuracy.

By computing the *representativeness* of the mean faces with different values of k in a given dataset, we obtain insights into selecting a reasonable k to balance

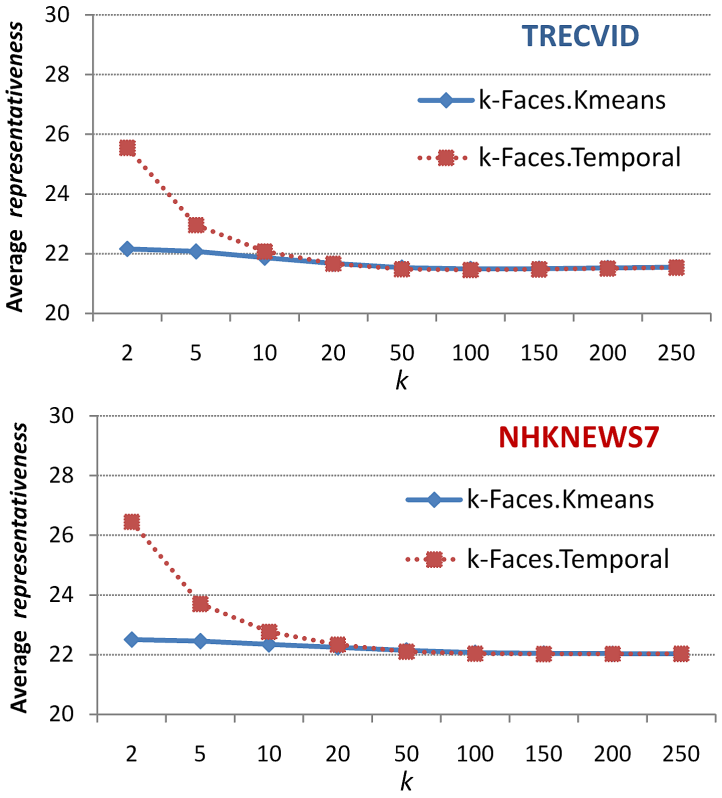


Figure 2.15: Evaluation on *representativeness* of mean faces computed by k-Faces.$Temporal$ and k-Faces.$KMeans$ over TRECVID (top) and NHKNews7 (bottom) datasets.

computational costs and accuracy. At a value of k that keeps increasing k does not help to significantly improve the *representativeness* of the mean faces, that value of k should be selected (e.g. $k = 20$ as in our datasets). This is because if the mean faces do not change, performance also may not change.

In order to investigate the performances of k-Faces approaches compared to other approaches on public and well-known video datasets, we carry out other experiments using two datasets, Honda/UCSD [59] and Buffy [61]. The Buffy dataset already provides face-tracks (i.e. sets of faces belonging to the same

characters). Meanwhile, with Honda/UCSD, we download its videos and apply our face-track extraction approach to obtain face-tracks. All experimental settings are kept the same as those in our previous experiments. The performance of all evaluated approaches are presented in Figure 2.14.

The experimental results shown in Figure 2.14 once again demonstrate that the performance of our propsosed k-Faces.$Temporal$ approach is comparable to that of other state-of-the-art approaches.

2.7 Summary

In this work, we investigate face retrieval in large-scale news video datasets. Our contributions are threefold.

- We present a face-track extraction approach that incorporates techniques to handle problems due to flash lights, partial occlusions, and scattered appearances of characters in real-world news videos.
- We present an approach for face-track matching that significantly reduces the computational cost while achieving competitive performance compared with state-of-the-art approaches.
- We prepare datasets, evaluate state-of-the-art face retrieval approaches, and make public two real-world face-track datasets of such scales that have never been considered in the literature.

CHAPTER 3

Query Recommendation for Video Retrieval

3.1 Introduction

Thanks to the advances of modern technology, a large amount of digital videos and images can be easily created and stored. Although videos and images are somewhat different types of media, videos can be convert to images by regarding video frames as individual images or using key-frame selection approaches to select key-frames from the videos. By doing that, approaches applied to image domain can be also applied to video domain. Image search or retrieval approaches are such kind of approaches. In the remaining of this work, we treat video retrieval as image search and video frames as images.

Image search has gained interest in recent years because of its importance and wide range of applications. In a typical scenario, users supply a query item, which is usually a region cropped from an image. The search system then returns a list of relevant images retrieved from a database. The images are expected to contain the query item. Extensive studies have been conducted with an eye to improving the performance of this sort of search [65–74]. However, regardless of the powerfulness of state-of-the-art search techniques, there are still cases in which users are disappointed with their search results. The reason is that relevant items are not in the database. Under such circumstances, whatever the search technique is, results are obviously irrelevant and unexpected. A normal user

Figure 3.1: Having no idea about the database, how do users know whether their intentional search item will return relevant results , without any trial search ? If not, which query should be used instead to search or to explore the database? Recommend-Me targets to answers these questions. In this example, it recommends a stop sign (green box), which occurs in 4 images of the database, rather than other candidate items (yellow boxes), which can not be found in the database.

without prior knowledge about a database has no choice but to search it by trial-and-error. We decided to tackle this problem to help users in searching and exploring images in unknown databases. Our proposal is a novel recommendation system, named Recommend-Me.

The envisioned scheme can be described as follows (see Figure 3.1 for an example). Given an unknown database and an input query image, Recommend-Me

automatically presents its recommendations to the user. One recommendation is one item, bounded by a rectangular region, in the query image. Each recommended item is assigned a number to show in how many images of the database it occurs. Items with larger assigned numbers will be more recommended. By providing such recommendations, Recommend-Me helps users to:

- avoid unexpected search experience with poor queries that are subjectively (and sometimes randomly) selected,
- rapidly refine the input query image before any actual search, if the recommendations show that current search intention can not return relevant results,
- explore the database for knowledge discovery using the recommendations as hints.

Recommend-Me is a pure visual recommendation system. No extra information or knowledge is required for an input besides an input query image and a database. Compared to text-based recommendation systems, Recommend-Me has advantages especially in cases that users' intention is difficulty to describe by texts.

To automatically generate recommendations, we need to address several issues. First, there tends to be a huge pool of candidate regions in the input query image. Basically, any rectangular region in the image can be considered as a candidate item. Examining all of them would incur an enormous computational cost. Second, even if a candidate item is known, enumerating its occurrences in the database is a not trivial task because it is subject to many variations in viewpoint, scale, rotation, and occlusion. Furthermore, the cost of scanning all

regions of the images of the database will inevitably be prohibitive for practical purposes.

To this end, we propose a scalable approach to tackle the aforementioned problems. The proposed approach is based on the observation that *(i) at the image level, most of the images in the database do not have any relevant region to regions of the input image. and (ii) at the region level, the number of regions of interest, such as object regions, is very small, i.e 5-10.*

Since we only need to find a small number of region pairs (formed by one region in the input query image and one region in an image in the database) with top similarity scores for recommendations, it is not efficient to have the same treatment for all possible region pairs. Instead, we use scalable and efficient techniques to quickly filter out 'easy' region pairs, i.e. region pairs having low similarity score. For 'difficult' region pairs, i.e. region pairs having high potential of recommendations, we use more complex processing to ensure high accuracy.

Specifically, firstly we use inverted index to quickly filter out a large number of images in the database that are not relevant to any region of the input image. Each image is represented by a bag of visual words. Each visual word corresponds to 'salient' regions in the input image and is represented by such a descriptor as SIFT [65] that is invariant to affine transformations. The inverted index is constructed using this representation and allows to quickly return a list of top-k relevant images. Then we use a selective search approach to sample regions bounding object-like items in the remaining images as a preprocessing step. As a result, it significantly reduces a large number of region pairs with low computational cost. Finally, for the remaining region pairs, we propose a branch-and-bound framework that allows to reduce the number of similarity evaluations to identify the region pairs of highest score.

Our contribution is two-fold:

- A novel system that recommends good queried regions given an input image to improve search quality and user experience. The proposed approach to build such system can be used in practical image retrieval systems and show scalable and feasible to large datasets (100K and 1M). To the best of our knowledge, it is never studied before for visual search, since most of the systems are search systems, not recommendation systems.

- The branch-and-bound framework is able to find the global optimum of a quality function over all possible region pairs, thus returning the same region pairs of highest scores that an exhaustive search approach would. However, it requires much fewer similarity evaluations. For this purpose, we introduce a novel representation based on hierarchical structures describing sets of region pairs and a corresponding function bounding the similarity scores of pairs over such sets.

We review related work in section 3.2. The framework overview are presented in section 3.3. Section 3.5 introduces experiments on two large image datasets of 100K and 1M images respectively. Finally, section 3.6 concludes this work.

3.2 Related Work

On the topic of discovering common items, Recommend-Me is related to recent studies on mining common items in image databases such as [75, 76]. However, in contrast to these studies, Recommend-Me targets items which are shared by both an image database and the user's particular interest limited in the input initial image. Meanwhile, [75, 76] only aim at finding common items within the

database. One can employ these techniques to solve our problem by first identifying common items among the images of the database, then looking them up in the initial query image again to make recommendations. However, doing that incurs extra cost for mining unnecessary items that appear in the database, but not in the initial query image. Furthermore, mining common objects in image databases is an yet another challenging problem that has not been solved.

One of the most related studies to ours for query suggestion is that of Zha et al. in [77]. They introduced a system, called Visual Query Suggestion (VQS), that simultaneously provides both keyword and image suggestions to users. There are clear differences between Recommend-Me and VQS. VQS requires an initial text query for formulating the suggestion, and its suggestions are both keywords and images. On the other hand, Recommend-Me takes an image as input, and its outputs are regions in the image. Recommend-Me is a query suggestion system based on pure visual information. Above all, although both Recommend-Me and VQS aim at helping users search for images, their targeted problems are different. VQS proposes to help users to overcome their tendency to formulate ambiguous queries by precisely expressing search intents, assuming the relevant items are always available. Meanwhile, Recommend-Me helps users to select queries based on the existence of relevant items in the retrieved database. To the best of our knowledge, Recommend-Me is the first attempt at this sort of targeted suggestion scheme.

From a technical point of view, our solution is motivated by recent works on object localization and sub-image retrieval based on branch-and-bound optimization [78–80]. However, ours is differentiated from other studies in that we represent sets of region pairs, instead of only sets of regions. ESS and ESR [78, 79] use coordinate intervals for their presentation. In contrast, we utilize hierarchical

structures in order to do that, since our regions are discrete. Although coordinate intervals as in ESS (or ESR) can be extended to represent sets of region pairs, such a criterion in the context of the branch-and-bound algorithm may suffer from the curse-of-dimensionality problem since the number of dimensions required at least doubles. Finally, ESS, ESR and Recommend-Me differ in that they have different approaches to constructing a bounding quality function and to computing bounding values over the sets.

3.3 Framework Overview

The framework of Recommend-Me consists of four main steps. Figure 3.2 summarizes the pipeline.

Step 1: Select candidate images. In large scale image search systems, most of the images in the database are not relevant with the input image. Therefore, the similarity of region pairs formed by a region in the input image and a region in a image in this irrelevant set is apparently low. Including region pairs derived from these images in similarity evaluation is expensive but unnecessary. We use inverted index (cf. Section 3.3.1), an efficient indexing technique well-known in text retrieval, to quickly filter out these images and only keep a list of small number of candidate images.

Step 2: Select candidate regions in each image. Although the number of candidate images to be evaluated is small (e.g. 1,000 images), the number of region pairs is still huge. For example, one 320×240 image usually has 100K candidate regions, the number of possible region pairs when matching with 1,000 candidate images of the database is $1,000 \times (100,000)^2 = 10^{13} = 10{,}000$B. Therefore, using all possible rectangular regions in images as candidate items is overly

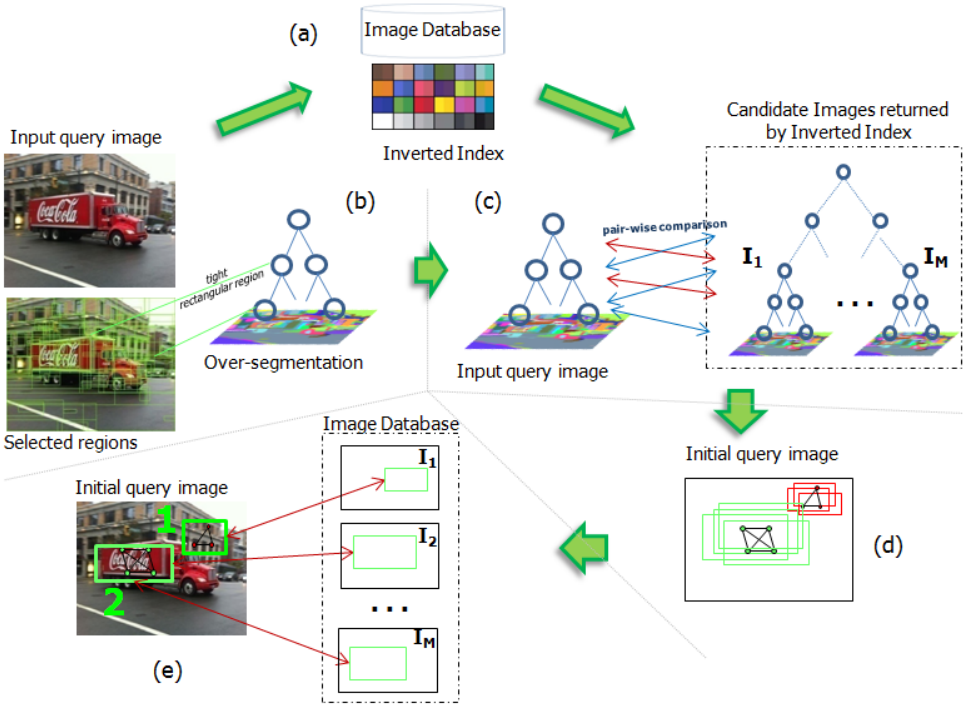


Figure 3.2: Framework pipeline. (a) Step 1: a list of candidate images are returned by using inverted index. (b) Step 2: only rectangular regions (green boxes) that tightly bound segments throughout the hierarchy are selected. One node of the hierarchy represents one rectangular region. (c) Step 3: among all possible region pairs (blue arrows) between regions of the initial query image and regions of all images in the database, only the top TP pairs with the highest similarity scores (red arrows) are returned. (d) Step 4: overlapping rectangular regions in the initial query image are grouped using maximal clique analysis. (e) Step 5: for each group, all the of images such that one image contains at least one match of one member region of the group are counted. These numbers (green numbers) are used to rank the groups. One group represents one recommendation (green boxes). Best viewed in color.

expensive in the subsequent processing. More importantly, users are often attracted by object-like items. Thus, we employ a visual attention based approach (cf. Section 3.3.2) to select a subset of such object-like regions for both the input image and the images in the candidate image sets.

Step 3: Find top region pairs with the highest similarity scores. There will be a pool of region pairs. However, only region pairs with sufficiently high similarity scores are meaningful for identifying occurrences of candidate regions. In this step, we propose a branch-and-bound framework (cf. Section 3.4) to find the top TP (the expected number of returned region pairs) of such pairs in the pool.

Step 4: Group overlapping regions. Given TP region pairs returned in Step 3 and assuming each region pair in the TP pairs is formed by a candidate region and its corresponding match, we can enumerate the number of occurrences of the items. However, there are likely several regions that overlap each other due to the merging done in Step 2. These regions would be perceived as the same region by users. Thus, we propose to use maximal clique analysis algorithm (c.f Section 3.3.3) to group such regions so that the recommendations will be consistent. One clique is one group of regions.

Step 5: Formulate recommendations. Finally, for each group of regions, we count the number of images containing at least one match of one member region of the group. The number indicates how frequent the item, represented by the group, occurs in the database. Using those numbers, we rank all groups and show them as recommendations to users. A representative of each group is a rectangular region located by averaging the coordinates of all member regions of the group.

3.3.1 Select Candidate Images Using Inverted Index

The inverted index is a popular technique in large scale text retrieval. In object retrieval, the technique described in [68] that has mimicked simple text-retrieval systems using the analogy of 'visual words' is widely used. Firstly, a keypoint detector is used to find 'salient' regions in the input image. Next, these regions are represented by such a high dimensional descriptor as SIFT[1] [65] that is robust to local affine distortion. These descriptors are then clustered into a vocabulary of visual words, and each salient region is mapped to the visual word closest to it under this clustering. Finally, an image is represented as a bag of visual words, and an inverted index is constructed similar as in text-retrieval systems.

When the dataset size is increased, the bottle neck is in steps such as clustering and mapping regions to visual words. In [69, 81], two scalable clustering methods are proposed and widely used. Therefore, we just simply follow these techniques for this step.

Specifically, we use Hessian Laplace keypoint detector to find affine-invariant regions and SIFT descriptor is computed for each region. Hierarchical k-means (HIK) [81] is used for clustering in order to form a codebook. This codebook has 1 million visual words to guarantee good performance according to [69]. We use an open source software available at http://www.vlfeat.org to implement this step.

3.3.2 Select Candidate Regions in Each Image

There are many methods that have been proposed to rapidly detect visually salient regions in an image using visual attention based approach. We propose to

[1]Scale-invariant Feature Transform

use one of the state of the art methods proposed by Van de Sande et al. [82] to find object-like regions in an input image. The method starts by over-segmenting an image into disjoint regions. Then, it performs a greedy algorithm that iteratively merges the two most similar regions together until the whole image becomes a single region.

We used two color channels, RGB and Hue, since the regions generated on those channels can cover 99.72% of the area of the annotated item regions in our dataset. A virtual root node was created to compose two color-dependent binary trees into one unique binary tree for each image. In addition, the rectangular regions which were smaller than 40 x 40 pixels were discarded.

All regions throughout the hierarchy are considered to be candidate items. Each item is represented by its rectangular bounding region. The code is available at http://koen.me/research/selectivesearch/.

Using this method, the number of candidate regions in each image is reduced 200 times from 100K to 500 regions. Furthermore, using the output hierarchy, we propose a new representation for set of regions in the branch-and-bound framework.

3.3.3 Maximal Clique Analysis Algorithm

Given the set of regions in the initial query image, we built a graph in which two regions were connected if they largely overlapped each other (we use the approach of PASCAL VOC[2] with a tighter threshold, 0.8). The Bron-Kerbosch algorithm was then applied to find all maximal cliques in the graph. One clique was one group of regions and one recommendation representative.

[2]http://pascallin.ecs.soton.ac.uk/challenges/VOC/

3.4 Branch-and-Bound Framework for Finding Top Region Pairs with the Highest Similarity Scores

In this section, we describe our approach for efficiently finding the top $\mathcal{TP}$ similar region pairs in the pool of all possible region pairs.

Given two sets of regions $\mathcal{R}_A$ and $\mathcal{R}_B$, the set of all possible region pairs can be then represented as $\mathcal{R}^* = \{(r_1, r_2) : r_1 \in \mathcal{R}_A, r_2 \in \mathcal{R}_B\}$. By using the similarity function $f : \mathcal{R}^* \rightarrow \mathbb{R}$ to evaluate the similarity of two regions in a pair of $\mathcal{R}^*$, we have to solve the following optimization problem in order to find the region pair p with the highest similarity score.

$$p = \arg\max_{p' \in \mathcal{R}^*} f(p') \tag{3.1}$$

Because $\mathcal{R}^*$ has on the order of $\mathcal{O}(|\mathcal{R}_A| \times |\mathcal{R}_B|)$ elements, it is expensive to perform this maximization exhaustively. We hence propose to use a branch-and-bound algorithm [83] to solve the problem. Once p is found, we can obtain the other top region pairs by continuing the search process with the remaining search space, in which the found top pairs have been eliminated.

A general branch-and-bound algorithm works by hierarchically dividing the parameter space into disjoint parts; this is called the branching step. In the bounding step, each part is assigned an upper bound for which the quality function could take on any of the members of the part. Those parts of the parameter space with higher upper bound values are examined first. Thus, many portions of the parameter space can be eliminated if their upper bound values imply that they cannot contain the maximum. In our problem, the parameter space is the set of all region pairs $\mathcal{R}^*$, and the quality function is the similarity function f.

It is worthy to point out that it is not trivial to re-use the branch-and-bound

framework proposed by Lampert et al. [78, 79]. Firstly, the search space in our framework is region pairs instead of single region as in [79]. If using up four parameters $[Left, Top, Width, Height]$ to represent a set of regions in one image, it is needed eight parameters to represent a set of region pairs of two images. As pointed in [79], using such many parameters will lower efficiency of the branch-and-bound framework.

Secondly, the method described in [79] is used for finding matches between one region in the input image and all possible regions in images of the database. Simply repeating that method for each candidate region in the input image and then re-rank output matches is not efficient because a set of candidate regions in the input image can be used in branching step.

In our branch-and-bound framework, we use more effective representation for a set of region pairs by using the hierarchy of candidate regions returned by the method described in Section 3.3.2. For this new representation, a new quality function for bounding step is proposed.

Assuming we can organize regions in $\mathcal{R}_A$ and $\mathcal{R}_B$ into two hierarchical structures $\mathcal{T}_A$ and $\mathcal{T}_B$ respectively, so that:

(a) all regions are leaf nodes of the structures and non-leaf nodes are *virtual* nodes,

(b) if each node is represented by a histogram $\mathcal{H}$ with N_b bins, the value in each bin of a child node is constrained to be equal or smaller than the value in the same bin of its parent node.

Given such structures, we show in what follows how the branch-and-bound algorithm can be used to solve our problem.

Let n^A and n^B denote two nodes on $\mathcal{T}_A$ and $\mathcal{T}_B$. And let $\mathcal{S}_l(n^A)$ denote the set

containing all leaf nodes explored from n^A. If n^A is a leaf node, $\mathcal{S}_l(n^A) = \{n^A\}$. Otherwise, given n_i^A with $i = 1, .., c_A$ being direct child nodes of n^A, $\mathcal{S}_l(n^A)$ can be recursively defined as follows:

$$\mathcal{S}_l(n^A) = \{\mathcal{S}_l(n_i^A) : \forall i \in \{1, .., c_A\}\} \tag{3.2}$$

In a similar way, we have:

$$\mathcal{S}_l(n^B) = \{\mathcal{S}_l(n_j^B) : \forall j \in \{1, .., c_B\}\} \tag{3.3}$$

Letting $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ indicate the set of node pairs formed by pairing nodes in $\mathcal{S}_l(n^A)$ with nodes in $\mathcal{S}_l(n^B)$, we get:

$$\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B)) = \{\mathcal{S}_l(n^A) \times \mathcal{S}_l(n^B)\} \tag{3.4}$$

Thus, if n^A and n^B are roots of $\mathcal{T}_A$ and $\mathcal{T}_B$ respectively, $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ will be exactly the entire search space $\mathcal{R}^*$.

Branching Step. Dividing up the search space (i.e. set of region pairs) covered by $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ can be done straightforwardly by utilizing the hierarchical structures $\mathcal{T}_A$, $\mathcal{T}_B$ at certain nodes n^A, n^B. Regarding 3.2, 3.3 and 3.4, $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ can be divided into disjoint parts as follows:

$$\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B)) = \{\mathcal{S}_l(n_i^A) \times \mathcal{S}_l(n^B) : \forall i \in \{1, .., c_A\}\} \tag{3.5}$$

Or,

$$\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B)) = \{\mathcal{S}_l(n^A) \times \mathcal{S}_l(n_j^B) : \forall j \in \{1, .., c_B\}\} \tag{3.6}$$

The way to divide can be based on the sizes of $\mathcal{S}_l(n^A)$ and $\mathcal{S}_l(n^B)$. We divide the larger first. The branching step is illustrated in Figure 3.3.

Bounding Step. An essential requirement for the branch-and-bound algorithm is the quality bounding function f^* used to determine whether a part of

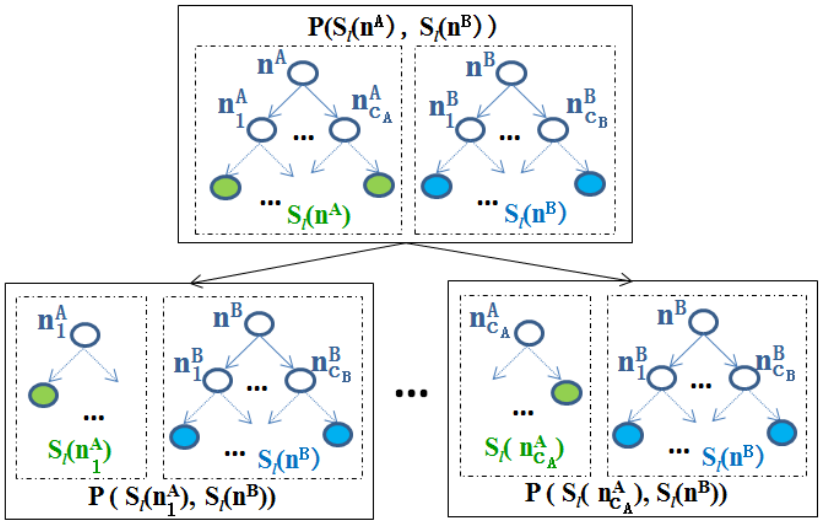


Figure 3.3: Branching Step. The parameter space covered by $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ is divided into disjoint parts $\mathcal{P}(\mathcal{S}_l(n^A_1), \mathcal{S}_l(n^B))$,..,$\mathcal{P}(\mathcal{S}_l(n^A_{c_A}), \mathcal{S}_l(n^B))$, regarding 3.2.

the search space should be examined. In particular, f^* bounds the upper values of f over a set of node pairs (i.e. region pairs).

Let us assume that we are now evaluating the upper bound of f over all region pairs in $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$. Among the several distance types for estimating the similarity of two regions, we use the Normalized Histogram Intersection (NHI) distance since it is well-balanced between computational efficiency and robustness [79]. We then rely on NHI to define f^* bounding the values of f, with:

$$f(s^A, s^B) = \sum_k \min\left(\frac{\mathcal{H}^{s^A}_k}{\| \mathcal{H}^{s^A} \|_1}, \frac{\mathcal{H}^{s^B}_k}{\| \mathcal{H}^{s^B} \|_1}\right), \quad \forall s^A \in \mathcal{S}_l(n^A), \forall s^B \in \mathcal{S}_l(n^B) \tag{3.7}$$

Referring to constraint (b) in constructing $\mathcal{T}_A$ and $\mathcal{T}_B$, we have:

$$\mathcal{H}^{n^A}_k \geq \mathcal{H}^{s^A}_k, \forall s^A \in \mathcal{S}_l(n^A), k = 1, .., N_b \tag{3.8}$$

$$\mathcal{H}_k^{n^B} \geq \mathcal{H}_k^{s^B}, \forall s^B \in \mathcal{S}_l(n^B), k = 1, .., N_b \tag{3.9}$$

As a result, the bounding value f^* over $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ can be derived as:

$$f(s^A, s^B) \leq f^*(\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))) = \sum_k \min\left(\frac{\mathcal{H}_k^{n^A}}{\min\limits_{\dot{s} \in \mathcal{S}_l(n^A)} \parallel \mathcal{H}^{\dot{s}} \parallel_1}, \frac{\mathcal{H}_k^{n^B}}{\min\limits_{\ddot{s} \in \mathcal{S}_l(n^B)} \parallel \mathcal{H}^{\ddot{s}} \parallel_1}\right), \forall s^A \in \mathcal{S}_l(n^A), \forall s^B \in \mathcal{S}_l(n^B) \tag{3.10}$$

We can efficiently evaluate f^* for the set of region pairs $\mathcal{P}(\mathcal{S}_l(n^A), \mathcal{S}_l(n^B))$ because f^* relies only on histogram representation of single rectangular regions n^A and n^B. Moreover, the normalization terms, which indicate the minimum number of visual words inside any member region of $\mathcal{S}(n^A)$, $\mathcal{S}(n^B)$, are computed once by using the integral image technique.

Inspired by [78, 79], we form the algorithm to work in a best-first manner. The algorithm iteratively examines the set of region pairs having the highest bounding value f^*. The algorithm stops if the set contain only one pair of region. Otherwise, the set is then divided into disjoint subsets for further search. Pseudo-code for the algorithm using a priority queue to store sets of region pairs, is given as follows.

Algorithm 1. Find the top region pair

Require: f^*, $\mathcal{T}_A$, $\mathcal{T}_B$

1: Empty the priority queue.

2: Push $\mathcal{P}(\mathcal{S}_l(root_A), \mathcal{S}_l(root_B))$,

the total search space, into the queue.

3: **repeat**

4: - Pop out the top state of the queue.

5: - Divide the state into disjoint parts. {Branching}

6: - Compute f^* for each part. {Bounding}

7: - Push the parts into the queue as new states.

8: **until** The top state is a pair of leaf nodes
(i.e. a pair of single regions)

To obtain more than one region pair, we simply continue the loop in Algorithm 1 until the expected number of region pairs $\mathcal{TP}$ is reached.

So far, our approach is based on an assumption that the sets of regions are already organized into hierarchical structures which satisfy constraints (a) and (b). In the remaining of this section, we show how to organize such sets, given the initial query image and the image database.

3.4.1 Organizing Regions into Hierarchical Structures

There are two type of region set. One is a set containing regions of one image. The other is a set containing regions of multiple images (i.e. database). With the first type of set, by applying the selective search approach introduced in [82] for item selection, regions in each image are already organized into a binary tree. Because such binary tree were constructed by bottom-up merging of regions, a parent region on the trees spatially covers its child regions in the image space (see Figure 3.4). As a result, constraint (b) is satisfied.

However, because we want to use all regions corresponding to all nodes throughout the tree as candidate item regions, constraint (a) will be violated if we keep using the tree for the branch-and-bound algorithm. In other words, all current non-leaf nodes of the tree will be treated as *virtual nodes* and will not be used as candidate item regions. Our solution to this problem is straightforward. We generate and attach a new leaf node to each non-leaf node of the current tree. The generated node is exactly the same as the non-leaf node it is attached to, which now becomes a virtual node. By doing that, we keep the spatial covering

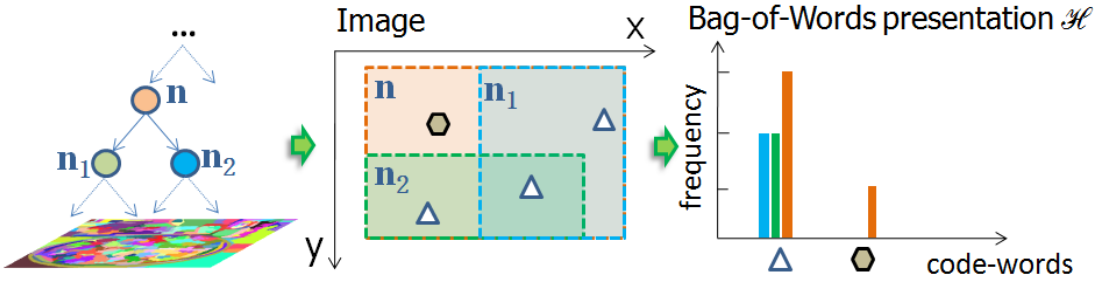


Figure 3.4: Bounding Step. A parent region (orange) covers its child regions (green and blue) on the image space. Thus, a code-world detected inside the child regions is also inside the parent region.

property of the original binary tree for the new hierarchical structure. Moreover, all non-leaf nodes will be taken into account as candidate item regions via their attachments. The new hierarchical structure therefore satisfies both constraints.

With a set containing regions from multiple images, we perform a two-stage organization procedure. In the first stage, regions in each image are organized into a hierarchical structure, as presented above. Given multiple hierarchical structures returned by the first stage, we use their root nodes as the initial elements to construct an yet another hierarchical structure over them by divisive clustering. We start with the full set of the elements. Then, we perform splits recursively as one moves down the hierarchy. In each splitting step, the split set is divided into k parts by using k-means clustering (in experiments, we use $k = 2$). Once the hierarchical structure is completed, we compute a histogram representation for all of its non-leaf nodes. The value at each histogram bin of a non-leaf node is the maximum of all values in the same bin of its child nodes. This is to ensure constraint (b) is satisfied. Finally, by unifying the results of both stages, we have a unique hierarchical structure over the set of regions of multiple images, which satisfies both constraints.

So, given the initial query image and a database, we now can construct two hierarchical structures. One is for the regions of the initial query image. The other is for the regions of all images in the database. Both structures then become the input for our approach to find the top region pairs with the highest similarity scores for making recommendations. Note that because the hierarchy of the regions of images in the database is independent of the query, we construct it only one time.

3.5 Experiments

3.5.1 Datasets

We perform experiments on two public and large datasets. The first dataset is OxfordBuildings-100K[3] that is widely used in evaluations of large scale object retrieval systems such as [69, 78–80]. This dataset consists of 5062 images collected from Flickr by searching for 11 different Oxford landmarks. In addition, there are 100,071 distractor images collected from Flickr by searching for popular Flickr tags. Among 5062 available object (i.e., building) images, we keep only *good* and *ok* images in which more than 25% of the building is visible.

The second dataset is MQA-1M[4]. This dataset consists of 438 images containing object instances of 52 types with various types such as fashion, art, pet, vehicle, and food, and was crawled from Yahoo! Answers, Flickr, and Google image search. In addition, there are over 1 million distractor images collected from Flickr. However, most of the instance images of MQA-1M containt only the object without complex background. We therefore collect a new set of instance images in more complicated settings for our experiments. Particularly, we crawl

[3]http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/
[4]http://vireo.cs.cityu.edu.hk/mqa/

Items	#images	Items	#images	Items	#images
airship	12	asimo	20	burgerking	27
clock	30	cocacola	36	exit_sign	15
fire_extinguisher	10	hp	18	monalisa	21
seven_eleven	31	starbuck	32	stop_sign	28
superman_logo	40	telephonebox	30	uniqlo	25

Table 3.1: The number of images for each type of item in our dataset.

a set of 375 images of 15 items. The number of images for each instance varies from 10 to 40 images (see Table 3.1 for details). All images are crawled from the internet using Google Image Search.

In each dataset, we randomly pick 3-5 images of each target object instance as the input query image. The remaining images of the instance are combined with distractor images and grouthtruth images to formulate the retrieved database. Location of an object instance (i.e, its bounding box) is manually annotated.

3.5.2 Performance Evaluation

A recommendation is a good one if it exactly locates an object instance which exists in both the initial query image and the database. We call such recommendations *hit recommendations*, and a good recommendation system should accurately provide them to users. More importantly, users always expect that hit recommendations are ranked higher than false recommendations (if there are a number of them) on the list of the recommendations by the system. Based on these insights, we evaluated the accuracy of Recommend-Me system using mean average precision (MAP) of the recommendation list.

MAP is widely used as standard criteria for evaluation of retrieval systems. It gives high score for relevant (or hit) items ranked at the top of the result list. In our system, for each input query image, the system returns a ranked list of top 20 recommended regions. These regions are compared against the ground truth (one region per image) to estimate average precision. We used an approach from the Pascal VOC challenge to clarify whether a recommendation is a hit recommendation or not. In particular, the intersection area between a hit recommendation and an item should be larger than half of their union area.

To evaluate the efficiency of Recommend-Me in finding $\mathcal{TP}$ region pairs with the highest similarity scores, we counted the number of evaluations for the quality bounding function in the branch-and-bound algorithm. This number was then divided by the number of all possible region pairs formed by regions in the initial query image and regions in images of the database. The later number can be understood as the number of region pair evaluations of a greedy search approach. That fraction was taken to be the efficiency improvement of Recommend-Me. Note that regions in images were pre-selected as presented in our framework above.

In this experiments, we also compare our approach with a standard approach using inverted index based on regions. Given each candidate region in the input query image, the approach return a list of top voted regions in images of the database. A region in an image of the database is voted by enumerating the number of shared visual codeword with the candidate region. All the lists of all candidate regions are then merged to find the top $\mathcal{TP}$ region pairs.

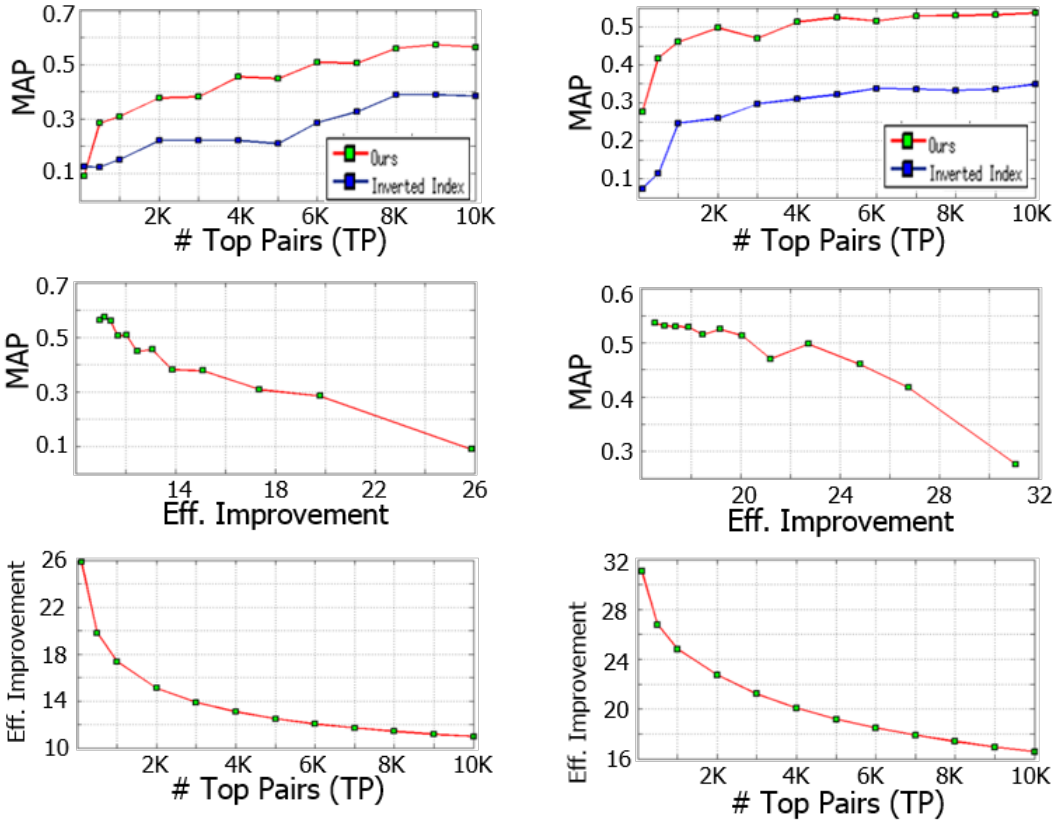


Figure 3.5: Evaluation results on two datasets (Oxford: left column, MQA-1M: right column). Top figures on both columns demonstrates that our proposed approach is significant better than the standard inverted index approach based on indexing regions in term of accuracy. Other bottom figures show the trade-off accuracy vs. efficiency of our approach at different values of $\mathcal{TP}$.

3.5.3 Results

Figure 3.5 shows the results of our evaluation with different values of $\mathcal{TP}$. Clearly, one can realize that the performance of Recommend-Me is influenced by $\mathcal{TP}$. By increasing $\mathcal{TP}$, we can obtain more region pairs with sufficiently high similarity scores. This gives more chances to get region pairs of the instance, thus improving

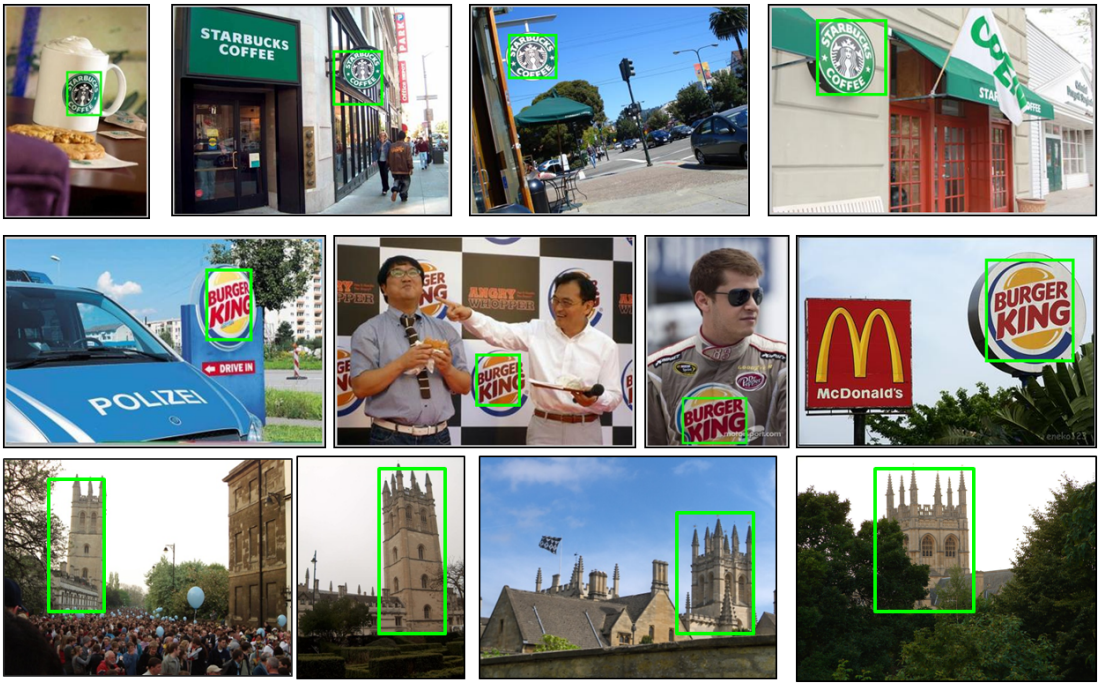


Figure 3.6: One image is one initial query image. In these examples, Recommend-Me successfully identified the items.

recommendation accuracy. However, the trade-off is efficiency. This is because the branch-and-bound algorithm has to visit more parts of the search space in order to find extra local optimals. On Oxford dataset, when $\mathcal{TP}$ increases from 100 to 10000, the recommendation accuracy also increases from 8.97% to 56.36%; meanwhile, the efficiency improvement drops from 26x to 11x (i.e., 26-11 times more efficient than the greedy approach).

Besides the decline of efficiency, it worth noting that keep increasing $\mathcal{TP}$ may not always give better accuracy. This is due to the limited number of occurrences of the instance in the database. At a value of $\mathcal{TP}$ that all occurrences of the instances are found, increasing $\mathcal{TP}$ means adding more noise to the recommendation list.

Figure 3.7: Examples of region pairs found by our approach. A pair is form by a region in the input query image and a region in an image of the database. The first image (from left to right) is the input query image, the others are images of the MQA-1M database.

In all of our evaluations, Recommend-Me performed more than tens of times faster than the greedy search with all $\mathcal{TP}$. This advantage will be important for practical applications. The results also show that our approach is significant better than the standard approach based on inverted index of regions. We achieve the highest accuracy at 57.28% of MAP with $\mathcal{TP} = 9000$ on Oxford datatset and 53.68% of MAP at $\mathcal{TP} = 10000$ on MQA-1M dataset. About running time, the inverted index based approach is faster than ours. However, they alwyas need a huge a mount of memory to store their index table. Therefore, if the number of candidate regions in images are increased, such approach becomes less feasible.

We learn that there are two types of false recommendations in top places of the list. The first type consisted of background regions (e.g. trees, buildings, roads), which are easily found in many images. Such regions can be considered as *stop-words* and should be removed. The second type was items lacking manual annotations such as windows, cars, and humans. Thus, recommendations about those items are not counted as hit recommendations. However, if users are interested in using them as hints to explore the database, they may still be very much helpful. Figure 3.6 presents examples of hit recommendations returned by

Recommend-Me in some query images. Meanwhile, Figure 3.7 show examples of region pairs found.

3.6 Summary

We described a system, named Recommend-Me, for making visual query suggestions. Given an input query image and a database, Recommend-Me gives recommendations that indicate which and how frequent regions in the input query image appear in the database. Such recommendations help users to select the search query, to rapidly refine the initial query image or to explore the database. An efficient solution to make Recommend-Me practical was also presented based on a two-stage approach using different treatments for different region pairs. To the best of our knowledge, Recommend-Me is the first attempt toward its targeted suggestion scheme.

CHAPTER 4

MIL-based Object Categorization for Content Analysis

4.1 Introduction

Video retrieval based-on concepts such as predefined object categories require object categorization approach which is to detect presences of the object categories in video at frame-level. Object categorization is a challenging problem especially when a label is provided for a training image only instead of the object region. Low categorization accuracy may result because the object region and background region within one training image share the same object label. To eliminate labeling ambiguity, object categorization and localization should be simultaneously performed.

To do that, we focus on object categorization using Multiple Instance Learning (MIL), which is a generalization of standard supervised learning. Unlike standard supervised learning in which the training instances are definitely labeled, in the MIL setting, labels are only available for groups of instances called bags. A bag is positive if it contains at least one positive instance. Meanwhile, all instances in negative bags must be negative. Given training bags and instances that satisfy MIL labeling constraints, MIL approaches can learn to classify unlabeled bags as well as unlabeled instances in the bags. Thus, if we regard each image as a bag

and sub-windows in images as instances, we can perform object categorization and localization simultaneously using MIL.

Several MIL approaches have been proposed [84–90]. Empirical studies [85, 87, 90] demonstrate that generative MIL approaches perform worse than discriminative MIL approaches on benchmark datasets, because of their strict assumption on compact clusters of positive instances in the feature space. Thus, it is more appealing to tackle object categorization by using discriminative MIL approaches. In a brief overview, discriminative MIL approaches can be found in [87–90]. Andrews et al. [88] introduce a framework in which MIL is considered in different maximum margin formulations. A similar formulation of [88] can be found in [91]. DD-SVM presented in [87] trains an SVM for bags in a new feature space constructed from a mapping model defined by the local extremums of the Diverse Density function on instances of positive bags. In contrast, MILES [89] uses all instances in all training bags to construct the mapping model without applying any instance selection method explicitly. IS-MIL [90] then propose an instance selection method to tackle large-scale MIL problems. Because [87, 89, 90] heavily rely on bag-instance mapping process which is out of scope, we address our work to the framework proposed in [88].

In this work, we first extend the framework in [88] using spatial relations between sub-windows. Although spatial relation information have shown their important role in computer vision tasks [92–95], there is a few of MIL works utilizing such information. Zha et al. [96] introduced a MIL approach which captures the spatial configuration of the region labels. However, their work target to multi-label MIL problem and spatial relations between segmented regions. Instead of that, we investigate single-label MIL problem and overlapping relations between sub-windows. In the framework [88], learning a discriminative MI clas-

sifier is formulated as a non-convex problem and requires an iterative solution. In each round, positive training sub-windows (i.e. instances) for the next round should be selected with certain criteria. With original criteria, selecting only one positive sub-window per positive bag may limit the search space for the global optimum; meanwhile, selecting all temporal positive sub-windows may add noise into learning. We propose to select a subset of sub-windows per positive bag to avoid those limitations. Spatial relations between sub-windows are used as clues for selection. We directly enforce sub-windows spatial relations into learning by selecting sub-windows of the subset based on their overlapping degree with the most discriminative sub-window.

Second, we propose to combine the proposed MIL-based approach with a global scene classifier to improve categorization accuracy. We are motivated by the fact that combining global scene classification with object detection has helped in improving the categorization accuracy [97]. However, training an object detector requires a large amount of manual annotation. The object detector may also fail when the object is occluded. Meanwhile, the presence of the object is not only indicated by the entire object region but any of its parts or its correlations with other regions in the image. To overcome these limitations, we propose using discriminative region localization instead of object detection in the combination. By learning to identify the most discriminative regions (i.e. instances) representing presence of objects in images (i.e. bags), our proposed MIL-based approach can be regarded as a discriminative region localization approach and used in the combination.

Experimental results demonstrate the effectiveness of our approach.

4.2 Related Works

Multiple Instance Learning was first introduced by Dietterich et al. [84] to deal with the problem of predicting drug molecule activity. In this work, a class of methods was proposed for learning an axis-parallel hyper-rectangle (APR) in instance feature space to capture the target concept. The obtained APR is supposed to contain at least one instance from each positive bag and exclude all instances from negative bag. Motivated by original idea of Dietterich et al., Maron and Lozano Perez proposed Diverse Density (DD) as a general framework for MIL [85]. Instead of using hyper-rectangle to describe the concept, they use a concept point in feature space. The optimal concept point is defined as the one maximize a measure called diversity density computed based on the number of positive bags have instances closed to the point and distances from negative instances to that point. Expectation-Maximization Diverse Density (EM-DD) proposed by Zhang et al. [86] is a variant of DD which combines EM approach with DD algorithm. Application of these generative methods is limited because they rely on an assumption that all true positive instances form a compact cluster in feature space. Besides generative methods, other attempts to utilize discriminative learning to MIL has been proposed such as mi/MI-SVM [88], DD-SVM [85] and MILES [89]. DD-SVM mostly focuses on bag classification and tries to overcome the limitation of DD approach by learning multiple target distribution for positive instances. In contrast, MILES use all instance in the training set to construct the feature map without applying any instance selection method. This prevent MILES from solving large-scale data set because of its high dimensional bag-level feature vector which corresponding to the total number of instances. Other recent MIL methods focus on object detection [98] and object modeling [99].

In the aspect of computer vision, our work related to weakly supervised object localization and recognition methods. Todorovic and Ahuja [100] proposed a framework to learn a tree-like representation of common object in a set of images by combining multi-scale image segmentation with sub-tree mining algorithm. Russel et al. [101] uses an unsupervised method to discover objects in image based on multiple segmentation and pLSA. The weakness of their method is the costly segmentation processes with numerous different parameters to guarantee their assumption that exist at least one region perfectly cover the object. Fergus at al. [102] build probabilistic models that not only learn the appearance but also the position of parts of object. Image classification is performed in a Bayesian fashion using the learned models. Among recently proposed methods, our work mostly related to the one proposed by Nguyen et al. [91]. They introduce a learning framework that simultaneously localizes the most discriminative sub-windows in images and learn a discriminative classifier to distinguish them. The formulated optimization problem is similar to MI-SVM. However, instead of using SVM iterative training loop to solve the non-convex objective, they use coordinate descent approach.

4.3 Support Vector Machine for Multiple Instance Learning

In statistical pattern recognition, given a set of labeled training instances coupled with manual labels $(x_i, y_i) \in \mathcal{R}^d \times \mathcal{Y}$, the problem is how to obtain a classification function going from instances to labels $f : \mathcal{R}^d \rightarrow \mathcal{Y}$. In the binary case, $\mathcal{Y} = \{-1, 1\}$ indicates positive or negative labels associated with instances. MIL generalizes this problem by relaxing the assumption on instance labeling. Labels are given for bags, which are groups of instances. A bag is assigned a positive

label if and only if at least one instance of the bag is positive. Meanwhile, a bag is negative if all instances of the bag are negative. Formally, given a set of input instances $x_1, \ldots, x_n$ grouped into non-overlapping bags $B_1, \ldots, B_m$, with $B_I = \{x_i : i \in I\}$ and index sets $I \subseteq \{1, \ldots, n\}$. Each bag B_I is then given a label Y_I. Labels of bags are constrained to express the relation between bag and instances in the bag as follows: if $Y_I = 1$ then at least one instance $x_i \in B_I$ has label $y_i = 1$, otherwise, if $Y_I = -1$ then all instances $x_i \in B_I$ are negative: $y_i = -1$. A set of linear constraints can be used to formulate the relation between bag labels Y_I and instance labels y_i:

$$\sum_{i \in I} \frac{y_i + 1}{2} \geq 1, \forall I : Y_I = 1 \quad \text{and} \quad y_i = -1, \forall I : Y_I = -1, \tag{4.1}$$

or compactly represented as: $Y_I = \max_{i \in I} y_i$.

Learning the discriminative classifiers entails finding a function $f : \mathcal{X} \to \mathcal{R}$ for a multiple-instance dataset with the constraint $Y_I = \operatorname{sgn} \max_{i \in I} f(x_i)$.

4.4 The Former Approaches of SVM-based Multiple Instance Learning

Andrews et al. [88] proposed two learning approaches based on SVM with different margin notions. The first approach, called mi-SVM, aims at maximizing the instance margin. Meanwhile, the second approach, called MI-SVM, tries to maximize the bag margin. Both mi-SVM and MI-SVM can be formed as mixed integer quadratic programs and need heuristic algorithms to be solved. The algorithms have an outer loop and an inner loop. The outer loop sets the values for the integer variables. Meanwhile, the inner loop trains a standard SVM. The outer loop stops if none of the integer variables changes in consecutive rounds.

The mixed integer formulation of mi-SVM based on the generalized soft-

margin SVM can be presented as:

$$\begin{aligned} \min_{\{y_i\}} \min_{\{w,b,\xi\}} \quad & \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ \text{subject to} \quad & \forall i:\ y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i\,,\ \xi_i \geq 0\,, \\ & y_i \in \{-1, 1\}\ \text{, and (1) hold.} \end{aligned} \tag{4.2}$$

In (2), labels y_i of instances x_i not belonging to any negative bag are treated as unknown integer variables. The target here is to find a linear discriminative *MI-separating* that satisfies the constraint wherein at least one positive instance from each positive bag lies in the positive half-space, while all instances belonging to all negative bags are in the negative half-space.

In MI-SVM, Andrews et al. introduce an alternative approach to the MIL problem. The notion of a margin is extended from individual instances to bags. The margin of a positive bag is defined as the margin of *"the most positive"* instance of the bag. Meanwhile, the margin of a negative bag is defined by the margin of *"the least negative"* instance of the bag. Let $x_{mm(I)}$ be the instance of bag B_I and has maximum margin to the hyper-plane. Then, MI-SVM can be formulated as follows:

$$\begin{aligned} \min_{\{y_i\}} \min_{\{w,b,\xi\}} \quad & \frac{1}{2}\|w\|^2 + C\sum_I \xi_I \\ \text{subject to} \quad & \forall I:\ Y_I = -1 \wedge -\langle w, x_i\rangle - b \geq 1 - \xi_I\,,\ \forall i \in I\,, \\ & \text{or} \qquad Y_I = 1 \wedge \langle w, x_{mm(I)}\rangle + b \geq 1 - \xi_I\text{, and } \xi_I \geq 0 \end{aligned} \tag{4.3}$$

4.5 Support Vector Machine with Spatial Relation for Multiple Instance Learning

MI-SVM and mi-SVM can be applied to object categorization by regarding each image as a bag and sub-windows in images as instances. However, their formu-

lations and heuristic solutions do not involve spatial relations of sub-windows despite such information being extremely meaningful. Surrounding sub-windows always contain highly related information with respect to visual perception. If a sub-window in image is classified as a positive instance, it is supposed to be associated with the object label given to the class. In that sense, its neighboring sub-windows should be positive also. For example, if a sub-window tightly covers an object, its slightly surrounding sub-windows also contain that object.

Moreover, in terms of learning, the original approaches require a heuristic iterative solution to obtain the final discriminative classifier. In each learning round, candidate positive instances must be selected for the next round. Thus, positive instance selection criterion is the key step in the learning process. With mi-SVM, selecting all positive instances in the current round may add noisy instances to learning. Meanwhile, selecting only the most positive instance which has largest margin in the current round, as in MI-SVM, may limit the search space for the global optimum. To avoid such limitations, we propose to select a subset of instances as candidate positive instances for the next learning round. Spatial relations between instances (i.e. sub-windows) can be used as clues for selection. Therefore, we extend the framework proposed by Andrews et al. to take the spatial relation between sub-windows into account. Positive candidate selection criteria of the approaches are illustrated in Figure 4.1 .

In our extension, the notion of a bag margin is used as in the MI-SVM formulation. This means the margin of a positive bag is defined as the margin of *"the most positive"* instance of the bag. However, we directly enforce the spatial relations between *"the most positive"* instance with its spatially surrounding instances by adding constraints to the optimization formulation. Here, let $x_{mm(I)}$ be the instance of bag B_I has maximum margin with respect to the hyper-plane,

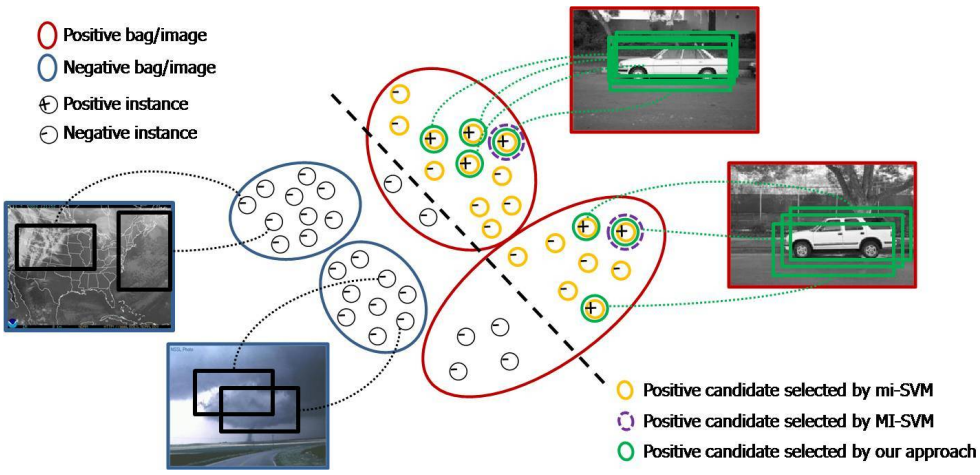


Figure 4.1: Illustration of positive candidate selection for the next learning round by different approaches. mi-SVM selects all temporal positive instances (*orange*). MI-SVM selects only the most positive instance per positive bag (*dash-purple*). Meanwhile, our approach selects a subset of spatially related instances (*green*) per positive bag based on their overlap degree with the most positive instance of the bag.

and $\mathcal{SR}(x_{mm(I)}, T)$ denotes the set of $x_{mm(I)}$ and instances that surround $x_{mm(I)}$ with respect to the overlap parameter T. An instance belongs to $\mathcal{SR}(x_{mm(I)}, T)$ if its overlap degree with $x_{mm(I)}$ is greater or equal to T, where $0 < T \leq 1$. The overlap degree between two instances (i.e. sub-windows) is the fraction of their overlap area over their union area. To this end, our formulation can be expressed as follows:

$$\begin{aligned} \min_{\{y_i\}} \min_{\{w,b,\xi\}} \quad & \frac{1}{2}\|w\|^2 + C\sum_I \xi_I \\ \text{subject to} \quad & \forall I : \; Y_I = -1 \wedge -\langle w, x_i\rangle - b \geq 1 - \xi_I \,, \forall i \in I \,, \\ & \text{or} \quad Y_I = 1 \wedge \langle w, x^*\rangle + b \geq 1 - \xi_I \,, \\ & \quad \forall x^* \in \mathcal{SR}(x_{mm(I)}, T) \,, 0 < T \leq 1, \text{ and } \xi_I \geq 0 \end{aligned} \tag{4.4}$$

This formulation can be cast as a mixed integer program in which integer variables are the selectors of $x_{mm(I)}$ and instances in $\mathcal{SR}(x_{mm(I)}, T)$. This problem is hard to solve for the global optimum. However, we exploit the fact that if integer variables are given, the problem reduces to a quadratic programming (QP) that can be solved. Based on that insight, our solution is as follows.

Pseudo code for heuristic algorithm

Initialize: for every positive bag B_I

  Compute $x_I = \sum_{i \in I} x_i / |I|$.

  $SR_I = x_I$.

REPEAT

  - Compute QP solution w, b for dataset with positive samples $\{SR_I : Y_I = 1\}$ and negative samples $\{x_i : Y_I = -1\}$.

  - Compute outputs $f_i = \langle w, x_i \rangle + b$ for all x_i in positive bags.

  - FOR (every positive bag B_I)

    Set $x_I = x_{mm(I)}$, $mm(I) = \arg\max_{i \in I} f_i$

    $SR_I = FindSurround(x_I, T)$

  - END

WHILE ($\{mm(I)\}$ have changed)

OUTPUT (w, b)

In our pseudo code, $FindSurround(x_I, T)$ is the function to find instances (i.e. sub-windows) surrounding x_I and have an overlap degree with x_I greater than or equal to T. The greater T is chosen, the fewer instances (i.e. sub-windows)

surrounding x_I are selected. Thus, T can be considered as a trade-off parameter for expanding the search space as well. T is a predefined number and is fixed throughout learning iteration. The optimal T is obtained automatically by cross validating on the training set. Additionally, negative candidates of all learning rounds are instances of the negative bags.

By optimizing formula (3.4), we obtain the SVM classifier for sub-window (i.e. instances) classification. Given an unlabeled image (i.e. bag), the classifier can be used to classify the image by finding the sub-window that maximizes the score of the sub-window classifier. If this score is positive, the image is said to be positive, which means it contains the object of interest. In addition, the sub-window yielding the maximum score is the most representative region in the image for the presence of the object. This explains why the approach can be regarded as a discriminative region localization approach. The approach is weakly supervised since training images are labeled at image-level only.

4.6 Combining with a Global Scene Classifier

A global scene classifier approach is that extracting features from the whole region of each image and then training a classifier based on these features. Approaches following this direction [103, 104] are helpful for capturing and learning the global scene configuration in which the objects appear. However, a low classification accuracy may result when the global scene configurations are ambiguous. To overcome such problem, one can use an alternative approach independent of the global scene classifier to detect the presence of the object and then combine them together. Following that methodology, using an object detector as the alternative approach in combination with a global scene classifier has shown an improvement in the classification performance [97].

However, because the object detector strictly aims at looking for the object region as a whole, it may fail when the object is occluded. Meanwhile, the presence of an object in an image is not only indicated by the entire object but also by any parts of the object or by the correlations of the object with other regions in the image. In addition, the main purpose of using an alternative approach is realizing the presence of the object, but is not exactly locating its position. Moreover, training an object detector requires a large amount of object-level manual annotation. Thus, in order to avoid the limitations of object detection, our weakly supervised discriminative region localization rather than object detection should be used in combination with global scene classification.

We are given two independent classifiers. The global scene classifier is efficient for capturing the global scene configuration in which the objects appear. Meanwhile, the discriminative sub-window classifier is good at locating regions in the images that best represent the presence of the object of interest. In order to use the classifiers to complement each other, which may help to boost the classification performance, we combine their results for final classification decisions. Without any loss of generality, we assume the global scene classifier is a discriminative SVM classifier. The problem of binary class classification is investigated.

Lets denote $P(O|G_I)$ as the probability that image I contains the object of interest given the score G_I returned by the global scene classifier. In addition, $P(O|D_I)$ is the probability that image I contains the object of interest given the score D_I returned by the discriminative sub-windows classifier. We obtain the probabilities $P(O|G_I)$ and $P(O|D_I)$ using SVM classifiers with probability estimation output. Note that in the case of our discriminative sub-windows classifier, the score of an image is equal to the score of the most representative

sub-windows of the image.

There are several ensemble learning techniques that can be used to combine the classifiers. One of the most straightforward approach is linear combination. Because of its simplicity and generality, we use linear combination as a base-line for the combining approach. In another attempt to get higher performance, we employ stacking (sometimes called stacked generalization) technique to combine the returned outputs of the classifiers. The global scene classifier and the discriminative sub-window classifier are used as the base-level classifiers in the stacking framework. An ensemble classifier is then learned on decisional samples formed by composing outputs of the base-level classifiers.

In particular, we use a k-NN classifier as the ensemble classifier. The training set for learning are obtained by firstly applying the base-level classifier to the images of a validation set. With each image I in the validation set, $P(O|G_I)$ and $P(O|D_I)$ are returned as the outputs of the base-level classifiers. We then form a 2-dimensional decisional sample as $(P(O|G_I), P(O|D_I))$. If I is a positive image (rept. negative image), the sample is assigned a positive label (rept. negative label). In order to generalize k-NN for different datasets in which the effectiveness of the combined classifiers to final classifying performance are unknown, we use weighted distance instead of normal Euclidean distance for finding the nearest neighbors. Given two decisional samples $A(a_1, a_2)$ and $B(b_1, b_2)$, the weighted distance is presented as $d_{(A,B)} = (\gamma(a_1 - b_1)^2 + (1 - \gamma)(a_2 - b_2)^2)^{\frac{1}{2}}, \gamma \in [0, 1]$. The γ can be set equal to 0.5 if there is no prior knowledge about the combined classifiers.

4.7 Experiments

4.7.1 Dataset

We perform experiments on Caltech benchmark datasets.

- **Caltech 4** contains images of 4 object categories: airplanes (1,075 images), cars_brad (1,155 images), faces (451 images), motorbikes (827 images), and a set of 900 clutter background images.
- **Caltech 101** consists of images in 101 object categories and a set of clutter background images [105]. Each object category contains about 40 to 800 images.

Ground-truth annotations indicating object's locations in images are available for all object categories (but cars_brad category in Caltech 4). These are challenging datasets because of their large variations in object appearance and background. Some example images are shown in Figure 4.2.

We evaluate the performance of the approaches on binary categorization tasks which are distinguishing images of each object category from background images. On the Caltech 101 dataset, with each binary classification task, a set of 15 positive images taken from one object category and 15 negative images from the background category are given for training; 30 other images from both categories are used for testing. The correlative numbers of positive images, negative images and testing images on Caltech 4 dataset are 100, 100 and 200 respectively. All images are randomly selected.

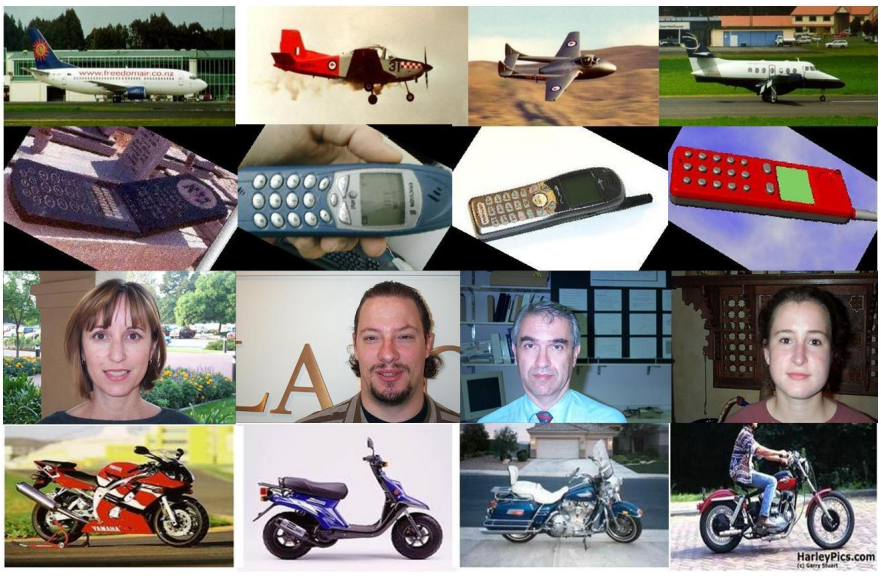


Figure 4.2: Example images taken from Caltech 101. From top to bottom are images of airplanes, cellphones, faces and motorbikes respectively.

4.7.2 Bag and Instance Representation

In order to apply the MIL approaches, we treat bags as images and instances of a bag as sub-windows in the image. We employ the standard Bag-of-Word (BoW) approach for feature representation. First, on each image, we sample a set of points using a grid. The sampling grid has an 8-pixel distance between adjacent points. Then, we use the SIFT descriptor to extract SIFT feature at each point. The SIFT descriptor frame has a 16-pixel width. All descriptors are then quantized using a visual codebook with 100 visual words obtained by applying K-Means to 100,000 training descriptors. Finally, the sub-windows of the image are represented by using a histogram of visual words appearing inside the sub-window region.

4.7.3 Evaluated Approaches

We conduct the experiment to clarify two main points. The first one is evaluating the role of taking spatial information into account for the proposed MIL-based object categorization approach (i.e. our weakly supervised discriminative region localization approach). In order to do that, we compare the proposed approach to the original SVM-based MIL approaches - mi-SVM and MI-SVM and two other standard approaches called GSC and MA. GSC denotes the traditional global scene classifier in which SVM is used to classify images represented by a histogram of visual words on the whole image region. Meanwhile, MA is an approach that uses tight object rectangles given manually as positive examples and a set of randomly selected windows from negative images - ten windows per negative image - as negative examples for training (MA stands for Manual Annotation). The second point is evaluating the effectiveness of combining the global scene classifier GSC and our proposed approach compare to using them individually and the state-of-the-art discriminative region localization approach presented by Nguyen et al. [91]. We denote the combinations based on linear combination and k-NN as LiCom, kNNCom. The approach by Nguyen et al. [91], LiCom, and kNNCom are only evaluated on the Caltech 101 dataset.

The measure for comparison is the accuracy ratio with respect to image classification performance. To obtain the best performance of the approaches for fairness, all parameters are optimized. Kernel parameters for SVM and overlap threshold T of our approach are automatically obtained by using the grid-search approach together with 5-fold cross validation.

Table 4.1: Average classification accuracy of the evaluated approaches on Caltech 4. Note that the performance of MA is computed on 3 categories (airplanes, faces and motorbikes) only due to the lack of ground-truth object box of the category cars_brad.

Approaches	Average Classification Accuracy(%)
MA	90.73
GSC	94.46
mi-SVM	72.54
MI-SVM	95.74
Ours	**96.28**

Table 4.2: Average classification accuracy of the evaluated approaches on Caltech 101.

Approaches	Average Classification Rate(%)
MA	78.32
GSC	83.53
mi-SVM	60.49
MI-SVM	84.25
Ours	**87.26**

Table 4.3: Average classification accuracy of the evaluated approaches on 10 categories of Caltech 101.

	MA	GH	mi-SVM	MI-SVM	Ours
Butterfly	76.7	76.7	53.3	86.7	**93.3**
Camera	70.0	80.0	53.3	73.3	**86.7**
Ceiling_fan	70.0	**80.0**	53.3	66.7	**80.0**
Cellphone	80.0	**90.0**	63.3	83.3	**90.0**
Laptop	80.0	76.7	66.7	76.7	**86.7**
Motorbikes	73.3	**93.3**	63.3	80.0	90.0
Platypus	83.3	90.0	53.3	86.7	**100.0**
Pyramid	**90.0**	**90.0**	63.3	76.7	**90.0**
Tick	76.7	83.3	56.7	80.0	**90.0**
Watch	**80.0**	**80.0**	53.3	73.3	**80.0**

Table 4.4: Average classification accuracy of all evaluated approaches on 101 categories of Caltech 101.

Approaches	Average Classification Accuracy(%)
GSC	83.53
Nguyen et al. [91]	84.55
Ours	87.26
LiCom	89.76
kNNCom	**91.58**

Table 4.5: Average classification accuracy of the evaluated approaches on 10 categories.

	GSC	Nguyen et al. [91]	Ours	LiCom	kNNCom
Bass	83.3	80.0	83.3	**93.3**	90.0
Binocular	78.3	75.0	78.3	85.0	**86.7**
Carside	85.0	91.7	95.0	**96.7**	**96.7**
Crayfish	80.0	78.3	80.0	81.6	**86.7**
Stopsign	80.0	71.7	88.3	**91.7**	**91.7**
Dragonfly	91.7	90.0	88.3	93.3	**98.3**
Ibis	88.3	91.7	90.0	**93.3**	90.0
Ketch	75.0	71.7	76.7	81.3	**83.7**
Menorah	70.0	68.3	70.0	73.3	**80.0**
Umbrella	66.7	73.3	75.0	76.7	**81.7**

4.7.4 Experimental Results

Table 4.1, Table 4.2, and Table 4.3 list the classification performances of the approaches on Caltech 4 and Caltech 101. Our proposed approach is superior to the others in most object classes. This means the most discriminative instances found by our approach are more meaningful than the one selected by MI-SVM and is also more discriminative than the object regions classified by MA. Moreover, these results prove that our arguments on the effectiveness of using the spatial relation and the limitations of the instance selection criteria of mi/MI-SVM are valid. Because of adding all possible positive instances, mi-SVM also adds more noise to learning and its performance consequently suffers. MI-SVM has a better accuracy than mi-SVM, but it is still worse than ours because of its limited search space.

On the other hand, Tables 4.4 and 4.5 show that our proposed approach (i.e.) also outperforms the approach introduced by Nguyen et al. [91]. By utilizing the spatial context information, our proposed classifier was able to discard the noisy representative sub-windows that do not satisfy the label constraints between the spatial related windows. Thus, this results in an improved performance with an increment of 2.71%. Although the formulations of the approach by Nguyen et al. [91] and MI-SVM are fairly similar each other, their perfomances are slightly different since MI-SVM works with sampled sub-windows (i.e. instances) only. Meanwhile, the approach by Nguyen et al. [91] utilizes an branch-and-bound based approach to find the most disriminative sub-windows in the space of all possible sub-windows in each image.

Both combinations based on k-NN and linear combination improved the classification performance. The k-NN based combination achieves the best performance with an average precision of 91.58%. We observed that this combination

provides a superior level of accuracy compared to the best of the individual classifiers for 78 object classes with differences varying from 1.6% to 10.0%. We also evaluated the k-NN based combination with fixed neutral $\gamma = 0.5$. It performed slightly worse than the linear combination but still substantially better than individual classifiers.

4.8 Summary

We proposed an extension of the SVM-based Multiple Instance Learning framework for object categorization by integrating spatial relations between instances into the learning process. Experimental results on the benchmark dataset show that our approach outperforms state-of-the-art SVM-based MIL approaches as well as standard categorization approaches. By regarding the proposed approach as a discriminative region localization and combining with a global scene classifier, the accuracy is significantly improved. Compared to other approaches consuming the same amount of annotation cost, it achieves better balance between cost-effectiveness and accuracy.

CHAPTER 5

Conclusion

In this final chapter, we summarize the original contributions and findings of our works presented in this dissertation. And, we present potential directions for future works.

5.1 Summary of Research

5.1.1 Face Retrieval in Large-scale Video Datasets

Robust face-track extraction. We propose a point tracker based face-track extraction approach, which is very efficient compared to approaches using an affine covariance region tracker or face clustering. The basic idea is that if two faces detected in different frames share a large amount of similar point tracks (i.e. trajectories of tracked points) passing through both of them, they are likely to be faces of the same character. To make point tracks reliable and sufficient in number for grouping faces of multiple characters throughout a shot, we introduce techniques to handle problems due to flash lights, partial occlusions, and scattered appearances of characters. All of these problems have not been carefully considered in former face-track extraction approaches, especially within the domain of news videos. By combining these techniques, our approach achieves a significant improvement to accuracy compared to a state-of-the-art approach.

Efficient face-track matching. We introduce an approach that significantly reduces the computational cost for face-track matching while maintaining a competitive performance with state-of-the-art approaches. Based on the observation that face-tracks obtained by tracking provide highly similar faces in consecutive frames, we argue that it is redundant to use all the faces in a face-track for learning the variation of faces. Thus, a set of faces is sampled from the original face-track for matching. The size of the set is much smaller than that of the original face-track. The mean face of the sampled faces in the set is then computed. The similarity between two face-tracks is based on the distance between their mean faces.

Large-scale face-track datasets from real-world videos. We investigate video retrieval with datasets whose scales have never been considered in the literature. Our first dataset is from 370 hours of TRECVID news videos and it contains 405,887 detected faces belonging to 41 individuals. The second dataset includes 1.2 million faces of 111 individuals observed in the NHK News 7 program over 11 years. The total number of available face-tracks is 5,567. The number of occurrences of each individual character varies from 4 to 550. We make both datasets publically accessible by research community.

5.1.2 Query Recommendation for Video Retrieval

We introduce a novel approach, named Recommend-Me, for visual query recommendation. Given an input image and a retrieved database, Recommend-Me gives recommendations that indicate which and how frequent object instances in the input image appear in the database. Such recommendations help users to select the search query, to rapidly refine the initial query image or to explore the database. An efficient solution to make Recommend-Me scalable on datasets with

million of images was also presented. Our solution comprises of two main stages. First, images unrelated to the input image are filtered by applying inverted index. Second, quantifying occurrences of multiple candidate object instances is formulated as an optimization problem and solved by Branch-and-Bound (BB) algorithm. To the best of our knowledge, Recommend-Me is the first attempt toward its targeted suggestion scheme.

5.1.3 MIL-based Object Categorization for Content Analysis

We propose an object categorization approach based on Multiple Instance Learning (MIL) to detect presences of predefined object categories in videos at frame-level. We improve standard SVM-based Multiple Instance Learning approaches for object categorization by integrating spatial relations between instances into learning process and combining with a global scene classifier. Experimental results on benchmark datasets show that our proposed approaches outperforms original MIL approaches as well as standard categorization approaches. Moreover, compared to other approaches consuming the same amount of annotation cost, it achieves better balance between cost-effectiveness and accuracy.

5.2 Future Directions

We briefly discuss here our future research based on the dissertation.

- Training data plays an important role in learning prediction model or classifier for video content analysis. Up to now, clear training manually annotated by human (e.g., labels for positive/negative images) is always required. However, with the growth of the Internet, a huge amount of information is already there. How to use such free source of data for training is

absolutely of great interest. For example, sample images for training classifiers can be crawled from image search engines (e.g., Google/Bing Image Search) using name of the concepts. They are noisy, but we can expect that there are more that one positive sample returned. In that sense, MIL-based approaches can be used to train the classifier. However, the current setting of MIL-based approaches is too strict since they assume only one positive sample/instance in a bag. Relaxing this assumption, possibly, helps us to improve their performance.

- We plan to extend the proposed recommendation approach so that it can support multiple input images. This is definitely helpful for users who have no clear search intention. On the other hand, to make recommendations meaningful in real-life applications, the approach should respond in real-time (or near real-time). In order to do that, the algorithm for finding pairs of similar regions should be enhanced with comprehensive acceleration techniques. A real-time recommendation system has a wide range of promising applications. One of such applications is augmented reality oriented application. For instance, an augmented dictionary program of which all objects or visual patterns appear on the screen of a users' recording device (e.g., camera, mobile phone, Google glass) can be explained by a pre-prepared visual dictionary (known as the retrieved database in our formulated problem).

References

[1] Cooper, M., Liu, T., Rieffel, E.: Video segmentation via temporal pattern classification. IEEE Trans. Multimedia (2007)

[2] Camara-Chavez, G., Precioso, F., Cord, M., Phillip-Foliguet, S., de A. Araujo, A.: Shot boundary detection by a hierarchical supervised approach. Proc. Int. Conf. Syst., Signals Image Process. (2007)

[3] Hoi, C.H., Wong, L.S., Lyu, A.: Chinese university of hong kong at trecvid 2006: Shot boundary detection and video search. Proc. TREC Video Retrieval Eval. (2006)

[4] Choi, Ko, K.C., Cheon, Y.M., Kim, G.Y., H-Il, Shin, S.Y., Rhee, Y.W.: Video shot boundary detection algorithm. ACM Trans. Multimedia Comput., Commun. Appl. (2006)

[5] Cernekova, Z., Pitas, I., Nikou, C.: Information theory-based shot cut/fade detection and video summarization. IEEE Trans. Circuits Syst. Video Technol. (2006)

[6] Yuan, J., Zhang, B., Lin, F.: Graph partition model for robust temporal data segmentation. 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) (2005)

[7] Zhang, H.J., J. Wu, D.Z., Smoliar, S.W.: An integrated system for content-based video retrieval and browsing. Pattern Recognit. (1997)

[8] Zhang, X.D., Liu, T.Y., Lo, K.T., Feng, J.: Dynamic selection and effective compression of key frames for video abstraction. Pattern Recognit. (2003)

[9] Porter, S.V., Mirmehdi, M., Thomas, B.T.: A shortest path representation for video summarization. Proc. Int. Conf. Image Anal. Process. (2003)

[10] Hu, W., Xie, N., Zeng, X., Maybank, S.: A survey on visual content-based video indexing and retrieval. IEEE Transactions on Systems, Man, and Cybernetics (2011)

[11] Rasheed, Z., Shah, M.: Scene detection in hollywood movies and tv shows. Proc. IEEE Conf. Comput. Vis. Pattern Recog. (2003)

[12] Rasheed, Z., Shah, M.: Detection and representation of scenes in videos. IEEE Trans. Multimedia (2005)

[13] Zhai, Y., Shah, M.: Video scene segmentation using markov chain monte carlo. IEEE Trans.Multimedia (2006)

[14] Tan, Y.P., Lu, H.: Model-based clustering and analysis of video scenes. Proc. IEEE Int. Conf. Image Process. (2002)

[15] Gu, Z.W., Mei, T., Hua, X.S., Wu, X.Q., Li, S.P.: Ems: Energy minimization based video scene segmentation. Proc. IEEE Int. Conf. Multimedia Expo. (2007)

[16] Goela, N., Wilson, K., Niu, F., Divakaran, A., Otsuka, I.: An svm framework for genre-independent scene change detection. Proc. IEEE Int. Conf. Multimedia Expo. (2007)

[17] Bashir, F.I., Khokhar, A.A., Schonfeld, D.: Real-time motion trajectory-based indexing and retrieval of video sequences. IEEE Trans. Multimedia (2007)

[18] Chen, W., Chang, S.F.: Motion trajectory matching of video objects. Proc. SPIE vol. 3972: Storage and Retrieval for Media Databases (2000)

[19] Su, C.W., Liao, H.Y.M., Tyan, H.R., Lin, C.W., Chen, D.Y., Fan, K.C.: Motion flow-based video retrieval. IEEE Trans. Multimedia (2007)

[20] Hsieh, J.W., Yu, S.L., Chen, Y.S.: Motion-based video retrieval by trajectory matching. IEEE Trans. Circuits Syst. Video Technol. (2006)

[21] Quack, T., Ferrari, V., Gool, L.V.: Video mining with frequent item set configurations. Proc. Int. Conf. Image Video Retrieval (2006)

[22] andN.Canagarajah, A.A.: Aunified framework for object retrieval and mining. IEEE Trans. Circuits Syst. Video Technol. (2009)

[23] Sivic, J., Zisserman, A.: Video data mining using configurations of viewpoint invariant regions. Proc. IEEE Conf. Comput. Vis. Pattern Recog. (2004)

[24] Fischer, S., Lienhart, R., Effelsberg, W.: Automatic recognition of film genres. Proc. ACM Int. Conf. Multimedia (1995)

[25] Rasheed, Z., Sheikh, Y., Shah, M.: On the use of computable features for film classification. IEEE Trans.Circuits Syst.Video Technol. (2005)

[26] Roach, M.J., Mason, J.S.D., Pawlewski, M.: Motion-based classification of cartoons. Proc. Int. Symp. Intell. Multimedia (2001)

[27] Yu, X., Xu, C., Leong, H.W., Tian, Q., Tang, Q., Wan, K.: Trajectory-based ball detection and trackingwith applications to semantic analysis of broadcast soccer video. Proc. ACM Int. Conf. Multimedia (2003)

[28] Chang, P., Han, M., Gong, Y.: Extract highlights from baseball game video with hidden markov models. Proc. IEEE Int. Conf. Image Process. (2002)

[29] Xu, G., Ma, Y.F., Zhang, H.J., Yang, S.Q.: An hmm-based framework for video semantic analysis. IEEE Trans. Circuits Syst. Video Technol. (2005)

[30] Xu, C.S., J.Wang, J., Lu, H.Q., Zhang, Y.F.: A novel framework for semantic annotation and personalized retrieval of sports video. IEEE Trans. Multimedia (2008)

[31] Snoek, C.G.M., M.Worring, van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. Proc. ACM Int. Conf. Multimedia (2006)

[32] Hu, W.M., Xie, D., Fu, Z.Y., Zeng, W.R., Maybank, S.: Semantic based surveillance video retrieval. IEEE Trans. Image Process. (2007)

[33] Viola, P., Jones, M.: Rapid real-time face detection. International Journal of Computer Vision (IJCV) (2004)

[34] Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. Transaction on Pattern Analysis and Machine Intelligence (PAMI) (2012)

[35] Turk, M., A.Pentland: Face recognition using eigenfaces. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (1991)

[36] Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (1994)

[37] Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Transaction on Pattern Analysis and Machine Intelligence (PAMI) (1997)

[38] Shakhnarovich, G., Fisher, J., Darrell, T.: Face recognition from long-term observations. European Conference on Computer Vision (ECCV) (2002)

[39] Cevikalp, H., Triggs, B.: Face recognition based on image sets. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)

[40] Wang, R., Shan, S., Chen, X., Gao, W.: Manifold-manifold distance with application to face recgonition based on image set. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)

[41] Fan, W., Yeung, D.Y.: Locally linear models on face appearance minifolds with application to dual subspace based classification. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2002)

[42] Yamaguchi, O., Fukui, K., Maeda, K.: Face recognition using temporal image sequence. Face and Gesture (FG) (1998)

[43] Fukui, K., Yamaguchi, O.: Face recognition using multi-viewpoint patterns for robot vision. International Symposium of Robotics Research (2003)

[44] Everingham, M., Sivic, J., Zisserman, A.: Taking the bite out of automated naming of characters in tv video. Image and Vision Computing (2009)

[45] Ramanan, D., Baker, S., , Kakade, S.: Leveraging archival video for building face datasets. International Conference on Computer Vision (ICCV) (2007)

[46] Sivic, J., Everingham, M., Zisserman, A.: Person spotting: video shot retrieval for face sets. International Conference on Image and Video Retrieval (CIVR) (2005)

[47] Merler, M., Kender, J.: Selecting the best faces to index presentation videos. ACM International Conference on Multimedia (ACMM) (2011)

[48] Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. European Conference on Computer Vision (ECCV) (2004)

[49] Shi, J., Tomasi, C.: Good features to track. IEEE Conference on Computer Vision and Pattern Recognition (1994)

[50] Hadid, A., Pietikainen, M.: From still image to video-based face recognition: An experimental analysis. Face and Gesture (FG) (2004)

[51] Lee, K., Ho, J., Yang, M., Kriegman, D.: Video-based face recognition using probabilistic appearance manifolds. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2003)

[52] Liu, X., Chen, T.: Video-based face recognition using adaptive hidden markov models. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2003)

[53] S. Zhou, V.K., Chellappa, R.: Probabilistic recognition of human faces from video. Computer Vision and Image Understanding (CVIU) (2003)

[54] Li, Y., Gong, S., Liddell, H.: Video-based online face recognition using identity surfaces. Face and Gesture (FG) (2001)

[55] Li, Y., Gong, S., Liddell, H.: Constructing facial identity surfaces for recognition. International Journal of Computer Vision (IJCV) (2003)

[56] Edwards, G., Taylor, C., Cootes, T.: Improving identification performance by integrating evidence from sequences. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (1999)

[57] Wolf, L., Shashua, A.: Learning over sets using kernel principal angles. Journal of Machine Learning Research (IJML) (2003)

[58] Gross, R., Shi, J.: The cmu motion of body (mobo) database. Carnegie Mellon University (2001)

[59] Lee, K., Ho, J., Yang, M., Kriegman, D.: Visual tracking and recognition using probabilistic appearance manifolds. Computer Vision and Image Understanding (CVIU) (2005)

[60] Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)

[61] Everingham, M., Sivic, J., Zisserman, A.: "hello,my name is... buffy" - automatic naming of charecters in tv video. British Machine Vision Conference (2006)

[62] Nguyen, T., abd D.-D. Le, T.N., Satoh, S., Le, B., Duong, D.: An efficient method for face retrieval from large video datasets. Conference on Image and Video Retrieval (CIVR) (2010)

[63] Satoh, S.: Comparative evaluation of face sequence matching for content-based video access. Face and Gesture (FG) (2000)

[64] Shan, C.: Face recognition and retrieval in video. Video Search and Mining (2010)

[65] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) (2004)

[66] Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. European Conference on Computer Vision (ECCV) (2006)

[67] Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision (IJCV) (2004)

[68] Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. International Conference on Computer Vision (ICCV) (2003)

[69] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007)

[70] Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. European Conference on Computer Vision (ECCV) (2008)

[71] Jegou, H., Harzallah, H., Schmid, C.: A contextual disimilarity measure for accurate and efficient image search. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007)

[72] Chum, O., Mikulik, A., Perdoch, M., Matas, J.: Total recal ii: Query expansion revisited. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)

[73] Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

[74] Perronnin, F., Liu, Y., Sanchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vector. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)

[75] Zhao, G., Yuan, J.: Mining and cropping common objects from images. ACM International Conference on Multimedia (ACMM) (2010) 975–978

[76] Yuan, J., Wu, Y.: Spatial random partition for common visual pattern discovery. International Conference on Computer Vision (ICCV) (2007) 1–8

[77] Zha, Z.J., Yang, L., Mei, T., wang, M., Wang, Z., Chua, T.S., Hua, X.S.: Visual query suggestion: Toeards capturing user intent in internet image search. TOMCCAP (2010)

[78] Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) (2009)

[79] Lampert, C.H.: Detecting objects in large image collections and videos by efficient subimage retrieval. International Conference on Computer Vision (ICCV) (2009)

[80] An, S., Peursum, P., Liu, W., Venkatesh, S.: Efficient algorithms for subwindow search in object detection and localization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

[81] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006)

[82] van de Sande K. E. A., R., U.J.R., T., G., M., S.A.W.: Segmentation as selective search for object recognition. International Conference on Computer Vision (ICCV) (2011)

[83] Lawler, E.L., Wood, D.E.: Branch-and-bound methods: A survey. Operation Research (1978) 191–194

[84] Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence (1997)

[85] Maronand, O., Lozano-Perez, T.: A framework for multiple instance learning. Advances in Neural Information Processing Systems (NIPS) (1998)

[86] Zhang, Q., Goldman, S.: Em-dd: An improved multiple instance learning technique. Advances in Neural Information Processing Systems (NIPS) (2002)

[87] Chen, Y., Wang, J.Z.: Image categorization by learning and reasoning with regions. Journal of Machine Learning Research (2004)

[88] Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. Advances in Neural Information Processing Systems (NIPS) (2003)

[89] Chen, Y., Bi, J., Wang, J.: Miles: Multiple-instance learning via embedded instance selection. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) (2006)

[90] Fu, Z., Robles-Kelly, A.: An instance selection approach to multiple instance learning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

[91] M.H., N., L., T., la Torre F., D., C., R.: Weakly supervised discriminative localization and classification: a joint learning process. International Conference on Computer Vision (ICCV) (2009)

[92] Galleguillos, C., Belongie, S.: Context based object categorization: A critical survey. Computer Vision and Image Understanding (CVIU) (2010)

[93] Marques, O., Barenholtz, E., Charvillat, V.: Context modeling in computer vision: Techniques, implications, and applications. Journal of Multimedia Tools and Applications (2010)

[94] Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A., Hebert, M.: An empirical study of context in object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

[95] Wolf, L., Bileschi, S.: A critical view of context. International Journal of Computer Vision (2006)

[96] Zha, Z.J., Hua, X.S., Mei, T., Wang, J., Qi, G.J., Wang, Z.: Joint multi-label multi-instance learning for image classification. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)

[97] Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. International Conference on Computer Vision (ICCV) (2009)

[98] Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. Advances in Neural Information Processing Systems **18** (2006) 1417–1424

[99] Dollr, P., Babenko, B., Belongie, S., Perona, P., Tu., Z.: Multiple component learning for object detection. European Conference on Computer Vision (2008) 211–224

[100] Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. IEEE Computer Vision and Pattern Recognition **1** (2006) 927–934

[101] Russell, B.C., Efros, A.A., Sivic, J., Freeman, W.T., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. IEEE Computer Vision and Pattern Recognition (2006) 1605–1614

[102] Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. IEEE Computer Vision and Pattern Recognition (2003) 264–271

[103] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006)

[104] Jianchao, Y., Kai, Y., Yihong, G., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

[105] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. Workshop on Generative-Model Based Vision, IEEE Conference on Computer Vision and Pattern Recognition (2004)