UNIVERSITY OF CALIFORNIA
RIVERSIDE

Learning Ranking Functions for Video Search on the Web

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Antony M. Lam

December 2010

Dissertation Committee:

Dr. Christian R. Shelton, Chairperson
Dr. Amit K. Roy-Chowdhury
Dr. Eamonn Keogh

The Dissertation of Antony M. Lam is approved:

_____

_____

_____

Committee Chairperson

University of California, Riverside

Acknowledgements

Doing a PhD has been a challenging but very rewarding experience. In my years as a student, I have had the good fortune of having many kind and supportive people in my life. First of all, I would like to thank my advisors Dr. Christian R. Shelton and Dr. Amit K. Roy-Chowdhury for without their guidance and patience, I would not be where I am today. The list of things they both have done for me would be too long to show here but I will be forever grateful for all of the support and time put in for me.

I would also like to thank former and current Riverside Lab for Artificial Intelligence Research (R-LAIR) student members for our fun and fruitful interactions. In particular, Juan Casse, Busra Celikkaya, Dr. Yu Fan, Kevin Horan, Dr. Kin Fai Kan, William Lam, Joon Lee, Dr. Guobiao Mei, Dr. Jing Xu, and Dr. Teddy Yap, Jr. In addition, I also want to thank the Video Computing Group. I have enjoyed our interactions during group meetings. In particular, Anirban Chakraborty, Chong Ding, Uttkarsh Gaur, Ting Yeuh Jeng, Ahmed Tashrif Kamal, Min Liu, Katya Mkrtchyan, Nandita Nayak, Dr. Ricky Sethi, Hessamoddin Shafeian, Dr. Bi Song, Elliott Staudt, and Moses Tataw. Also, thanks to all the people of EBU2 room 368 and those I met through the NSF EAPSI program!

I would also like to thank the professors I had at my undergraduate university, Cal Poly Pomona for supporting me in getting into a PhD program in the first place. These include but are not limited to Dr. Jim R. McKinney, Dr. Halina Przymusinska, Dr. Amar Raheja, and Dr. Barry I. Soroka. In addition, I would like to thank my high school calculus teacher Ms. Lillian Kimura for making me believe I could achieve a goal if I set my mind to it. Without

her, I would not have even attempted to enter a PhD program.

Last but certainly not least, I want to thank my family for the support they have given me throughout all these years. I would also like to give special thanks to my grandfather, Huyen Lam, who sadly passed away shortly before the start my graduate education. I always admired his character and integrity. In addition, I thank my parents, Sam Lam and Sophia Lam for their support and encouragement of my endeavors. I also thank my brother, William Lam for always being available to listen to my concerns and offering his opinions. (In addition to being a fun guy to be around.) Without my family, my years in graduate school would have been much more difficult. I dedicate this dissertation to them.

ABSTRACT OF THE DISSERTATION

Learning Ranking Functions for Video Search on the Web

by

Antony M. Lam

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2010
Dr. Christian R. Shelton, Chairperson

Videos on the Internet have become widespread. However search engines are still mostly limited to using associated text data to find desired content. In this dissertation, we build ranking functions that can directly analyze image and video content and assign a ranking to a database with respect to user queries.

A common approach to building ranking functions is to use a machine learning algorithm to perform a priori training of class concepts and use the trained classifier as the ranking function. However, a priori training of class concepts for retrieval is daunting since users queries can be very diverse. In addition, a priori training cannot capture the subjective component of user queries. For example, if a user were searching for videos of "nice basketball shots," there would be no way to know what the user considers "nice." Relevance feedback (RF) is an interactive search framework that captures user subjectivity and supports on-the-fly learning

of target classes.

However, RF is limited in its need for large amounts of user feedback when the data being searched are complex (e.g. Internet content). Transfer learning (TL) is a machine learning formulation where existing knowledge about a related "source" classification task can be used to improve the generalization performance of a "target" task (where training data is scarce). In this thesis we explore the combination of RF and TL and present a framework which can learn more from the user with less feedback. We show extensive experiments with real-world data taken from the Internet and show improved performance over past RF frameworks.

Although our RF and TL framework is effective for a wide range of queries, we acknowledge that there are some highly specific but common queries users could make which would benefit from more dedicated design of a ranking function. For example, finding particular people using face recognition would be an important type of query on the Internet. The problem in this case is well defined and objective. While the problem is specific, it is important enough to warrant the dedicated design of a ranking function. Thus we complete our studies in this thesis through the exploration of a robust face recognition based ranking function and show strong results in a challenging face identity retrieval task.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Video databases have become abundant thanks to the Internet. However, current methods for searching such databases are limited as the predominant algorithms rely on analyzing associated text data rather than the visual content itself. Direct analysis of visual content by computers would be desirable for achieving the best search results but computer vision is still an active field of research. In addition, Internet content exhibits a high degree of variability in terms of lighting, camera movement, and image quality. Thus appropriate functions for analyzing data for specific and general content need to be developed. This dissertation explores different approaches to searching video content on the web using machine learning and computer vision techniques.

When considering visual content on the Internet, users could make queries that span an large number of concepts. Thus building specific classifiers for all possible queries would be a daunting task. Furthermore, even if a very extensive amount of a priori learning were

available, it would still not cover the full range of queries users could make. This is because users can have subjective views about what they would consider relevant for any given search session. Thus pretraining classifiers would not be able to capture user subjectivity. We call such queries, "user-specific queries."

The majority of this dissertation addresses the problem of searching based on user-specific queries and we make use of the relevance feedback (RF) framework from information retrieval [Rocchio, 1971] to solve this problem. The basic idea behind RF is to first have a system retrieve items from a database based on an initial query and then solicit user feedback on the relevance of the returned items. (The feedback provides examples of relevant and irrelevant data.) This feedback is then used to improve the quality of returned results and the feedback process may be repeated until satisfaction.

The advantage of using RF is that the system can learn class concepts on the fly. In addition, if a user's target class is subjective, RF provides a framework where the user's subjective views can be learned. For example, if the user were interested in finding videos of "slightly obese cats," it would not be clear what the user considers to be "slightly obese" without interactively asking the user for clarification. Thus RF is useful when it is either prohibitively expensive or impossible to pretrain a wide range of target class concepts.

However in applying RF to video search, a number of issues arise. First, what kinds of feedback should a system solicit in order to quickly learn what a user has in mind? Second how can the machine capture a user-specific notion from just a few examples?

We approach the first issue by considering standard feedback solicitation methods from

Active Learning [Settles, 2010]. In particular we provide extensive experimental results comparing standard feedback strategies such as getting feedback on top ranked items, random items and the most ambiguous items in terms of relevance, as is done by Tong and Chang [2001].

For the second issue, we make use of transfer learning (TL) [Pan and Yang, 2009]. In TL one typically makes use of source data from a task where there is plentiful data and uses this source data to better learn a related target task where data is scarce. In our case, the user feedback is the scarce data of the target task. The source task data (which would only account for a small fraction of possible user queries) could be obtained from human labor through services such as Amazon's Mechanical Turk or just from the loosely labeled tags already available on popular video sharing websites. Specifically, we develop a simple method for training Support Vector Machines (SVMs) [Burges, 1998] when provided with source data from a related task.

We also address the issue of automatically determining how related tasks are by providing an experimental comparison of different methods for measuring relatedness. These include using the score clustering and score accuracy methods (to be described later), and cross validation methods. We also compare the mentioned methods against work in the literature to be described in the next chapter. Through these experiments, we show the benefits of combining RF and TL for accommodating a large variety of queries.

However, despite the benefits of our proposed framework for accommodating a large number of queries, we acknowledge the need for dedicated functions that analyze very spe-

cific but common concepts. For example, a large portion of Internet videos contain people. So being able to identify people is a very important feature of any search engine. Thus this thesis also explores the specific but common problem of identifying people through face recognition. Face recognition in the context of the web is especially challenging as imagery is taken under uncontrolled settings and there are many sources of variation that can be introduced into the appearance of any given face. One of the most important and challenging problems is recognizing people across out-of-plane rotations of the head (different poses). Specifically we build a collection of SVM classifiers trained on the joint appearances of facial parts from different poses and show that when linearly combined, these classifiers provide strong recognition between even frontal and profile views of the face. Comparisons against other approaches in the literature are made.

Overall this dissertation investigates different effective means for Internet search of visual content which are generally not currently used in industry. We give extra attention to the issue of how a computer can actively learn from and interact with human users to improve the search experience in a realistic scenario. As will be seen later, we demonstrate strong performance through experiments on a large YouTube dataset of nearly 3,600 videos with footage lasting 127.5 hours in total.

Chapter 2

Related Work in Content Based Retrieval of Images and Video

The problems of Content Based Image Retrieval (CBIR) and Content Based Video Retrieval (CBVR) are related and since much of the early work was in CBIR, we will introduce basic concepts through a discussion of CBIR.

2.1 Basic Concepts of Content Based Image Retrieval

The increasing availability of image data from various domains ranging from specific applications, such as searching medical image databases, to more general classes, such as the Internet, has made CBIR an important problem. [Datta et al., 2008]. When one builds a CBIR system, the first thing to consider is what the users would want to use the system for. A CBIR survey [Datta et al., 2008] notes that there are three dimensions of usage when it comes to

CBIR. The first is data scope (e.g. personal image collection, domain-specific, Web), the second is query modality (e.g. keywords, free-text, query image), and the third is user intent (browser, surfer, and searcher).

In our case, we are interested in searching the Internet and so our data scope is in one of the largest and most varied domains for search. In searching the Internet, common search engines such as Google regularly use crawlers to keep databases updated as the Internet is highly dynamic. While important, work on effective caching of the state of the Internet is beyond the scope of this dissertation. Instead we focus on the more fundamental problem of high variation in Internet media and assume the database to be searched is static (the media does not change).

For query modality, Datta et al. [2008] indicates that, depending on the application, users could search by keyword, free-text, image, graphics, or a composite. Search by keyword is the method most users today are familiar with. However the drawback of this approach is its reliance on text metadata (which may be absent or may insufficiently describe the media's content). Search by free-text is where a system would use natural language processing (NLP) to understand the user's request. This would require both NLP and Computer Vision to be mature enough for effective use (when metadata is either not available or not rich enough). For query by image, the user would submit an image representative of the kinds of images to be retrieved by the system. This method has the advantage that no text metadata needs to be available for search to be possible. The disadvantage is that it has to be clear what about the image makes it a relevant example. Graphic queries consist of hand-drawn examples or

computer-generated imagery to be submitted to a search engine. Some examples of systems using graphic queries mentioned by Datta et al. [2008] include sketch based retrieval of color images Chalechale et al. [2005] and querying using 3D models Assfalg et al. [2002]. There are advantages and disadvantages to any of the modalities of query and a study on different querying methods could be a subject of its own. Thus we restrict our study to querying by example since this is a common approach used in current research.

For user intent, Datta et al. [2008] introduces three types of users. The browser is a user with no end-goal. This type of user tends to jump from one query topic to the next during a search session. Thus queries would be incoherent in this case. The surfer starts a search session somewhat exploratory in the beginning and queries become more coherent over time. The searcher has a very clear idea of his target query from the beginning of the search session and tends to have short search sessions. In this dissertation, we focus on the case of a searcher.

Another concept of interest is the question of how search results should be presented to the user. Datta et al. [2008] show some categories and examples of ways different systems present results to the user. The most common is the relevance-ordered method used by Google. This is where results are sorted based on some numeric value indicating relevance. Another is the time-ordered method where images are shown in chronological order. This is useful for personal collections of images. Clustering is another means of presentation. For example, Zhang et al. [2009] use a face clustering algorithm to create clusters of same identity faces along with a visualization scheme. In addition, the interface also allows the user to

easily reassign cluster membership. The hierarchical approach shows results in the form of a tree with relationships between different database items. The work of Zhang et al. [2010] is able to discover social relation trees from photo albums. Of course, a composite of the mentioned presentation schemes could also be used. The composite method is used especially for personal collections Datta et al. [2008]. Since our emphasis is on Internet search, we adopt the Relevance-Ordered method.

The concepts discussed here are general enough that they also apply to CBVR. As noted earlier, in terms of data scope, we are interested in searching Internet content. While videos may not have been as prevalent on the Internet in the past, the popularity of video sharing websites has lead to an abundance of videos. For query modality, since videos are made up of a sequence of images, many of the methods for query could be applied to video search. There could be more room for innovation in query modality CBVR since videos contain more data than images but in our case, we adopt the most general of query methods–query by example (which is applicable to any kind of information retrieval). Similarly, the concepts of user intent also apply to any kind of retrieval system for the Internet. A major difference between CVBR and CBIR with respect to the concepts mentioned in this section is how retrieval results should be presented to the user. One approach would be to display retrieval results like the popular video sharing website, YouTube. On YouTube, video search results consist of a single still frame for each of the retrieved videos to give users an idea of what has be retrieved. Of course, this method of display has the drawback that it cannot adequately summarize all video content since only one video frame is shown. Other proposed methods of that could be

used for video summary to humans include Rapid Serial Visualization Presentation (RSVP) [Spence, 2002] where video frames could be presented to the user rapidly but in a pattern where humans would typically be able to get a good idea of what is in the video. Another proposed method for presentation is to use Manual Paging with Variable Page Size (MPVP) [Hauptmann et al., 2006], where video shots are displayed to users in in nxm grid and users can quickly label each shot on its relevance. Another simple approach we propose is to simply show the user more thumbnail frames than YouTube does for each retrieved video and if that is insufficient, allow the user to seek their own frames from the video in question. In this work, our focus is on the learning component of our retrieval framework. Thus we will assume that the user uses our proposed mode of presentation and is able to fully judge the relevance of videos.

2.2 Feature Extraction and Learning for Content Based Image and Video Retrieval

In CBIR, there are two basic issues called the sensory and semantic gaps [Smeulders et al., 2000]. The sensory gap is the gap between the state of an object in the real-world and what can be represented in the recording of the scene. The semantic gap is the gap between what information can be extracted from the visual data and how the user would interpret the data in any given situation.

Methods for addressing the sensory gap in the literature focus on extracting invariant

characteristics from images. For example, early work used color histograms for image indexing Swain and Ballard [1991]. Other work made use of feature extraction [Flickner et al., 1995], color constancy [Finalyson, 1996], Gabor Filters for local shape extraction [Manjunath and Ma, 1996], and viewpoint and occlusion invariant local features [Schmid and Mohr, 1997], to name a few. Some more recent work in invariant feature extraction include the popular Scale-Invariant Feature Transform (SIFT) [Lowe, 2004], Speeded-Up Robust Features (SURF) [Bay et al., 2008], and Space-Time Interest Points (STIP) for video [Laptev, 2005]. These local feature extraction algorithms are similar in that they all involve first finding "interesting regions" in images or video. These interesting regions are typically based on regions where high amounts of change can be detected such as corners. After detecting the interest points, descriptors are built on those regions which summarize the gradients in the region (flow are also computed in the case of STIP since motion is present in video). In contrast to local descriptors, Oliva and Torralba [2001] present and example of a global descriptor where the authors capture the "gist" of a scene by developing models to capture the naturalness, openness, roughness, expansion, and ruggedness in images.

Of course, once features are extracted, the next question is how we should compare features for retrieval purposes. The answer to this question depends on what the user's needs are and so this question addresses the semantic gap. A straightforward approach to comparing features if they are fixed length vectors such as histograms or the gist descriptor Oliva and Torralba [2001] is to use Euclidean distance (or other well known metric) to match images to known relevant examples. For retrieval of similar images, this is sufficient. For more difficult

retrieval tasks involving higher-level semantics, learning frameworks can be used. These include probabilistic frameworks [Vasconcelos and Lippman, 2000, Jin and Hauptmann, 2002] or the use of supervised learning techniques [Tong and Chang, 2001, Zhang et al., 2002, Tieu and Viola, 2004]. As the goal of closing the semantic gap is to interpret the given data as a user would in any given situation, involving user interaction in the learning process is a promising direction of research as this would allow the system to ask the user what he deems relevant in the given situation. As noted by Datta et al. [2008], relevance feedback (RF) is a major advance in user interaction technology for image retrieval.

Note that in addition to feature extraction techniques, other work in have made use of methods such as shot segmentation and key frame extraction for CBVR [Geetha and Narayanan, 2008]. A survey on spatial-temporal information for CBVR [Ren et al., 2008] also points out work involving tracking of object trajectories, structure from motion (inferring the 3D structure of an object from its motion), and sequence-to-sequence matching. An example of sequence-to-sequence matching, is the work of Adjeroh et al. [1999] where videos were converted into strings which could be compared via an edit distance. Recent work in view videos as strings of actions can be found in Kitani et al. [2008a] and Kitani et al. [2008b], where the probability of a video string can be determined using a stochastic context free grammar to model activities. While all these approaches have their advantages and disadvantages, application of most of these approaches to our data scope of the Internet remains challenging. Thus we restrict our attention to use interest point feature extraction techniques since they have been found to be most robust on highly varied data such as Internet content.

2.3 Relevance Feedback Search of Images

The application of RF to image retrieval provides a number of advantages. In image retrieval, a given image can have more subjective meanings than with text documents. This make traditional learning of class concepts (which are learned independent of the user) undesirable since the subjective component of the user is not captured [Zhou and Huang, 2003]. RF provides a framework for systems to interactively refine queries and learn a user's subjective notions of relevance for any given search session.

RF has been around since at least the 1970s [Rocchio, 1971]. However it was not until the late 1990s [Rui et al., 1998] that its application to image retrieval was introduced in the literature. In that work, user feedback (in the form of graded ratings on relevance for retrieved items) is used to adjust weights in a weighted feature similarity measure for retrieval. While results were effective, it was found that in practice, the amount of feedback from users would be small. For example, if a user is providing relevance labels for a set of images, one should not expect the user to provide a large number of labels. Thus much of the later work focused on how to effectively learn from only a small amount of data. For example, Wu et al. [2000] proposed a discriminant-EM algorithm that uses unlabeled data from the database for feature selection. Other work made note that positive examples from user feedback would often be more consistently located in feature space and proposed a one-class SVM framework for learning relevant regions in feature space [Chen et al., 2002]. Meanwhile other work viewed RF as an active learning (AL) process [Settles, 2010] and used strategies for deciding which

examples to ask the user for feedback on that would maximize learning [Tong and Chang, 2001, Goh et al., 2004, He et al., 2004]. While the different approaches to RF for CBIR can be effective, we adopt the AL approach as our base method in RF because it is the only approach that actively seeks to learn more from the user during search sessions.

2.4 Relevance Feedback Search of Video

In early work such as that of Liu et al. [1999], the authors use various factors computed from videos such as whether the corresponding shots of two videos appear in the same temporal order and the discrepancy in speed between corresponding shots. These factors are then combined in a weighted score where the weights could be set by the user.

Chen et al. [2008] present a system that first uses shot detection to segment out clips in videos and average color and motion histograms over all frames in a given clip are computed. A weighted similarity score for the color and motion features is then used to evaluate a given video clip. Relevance feedback from the user based on the top ranked videos is then used to adaptively tune the weights between the color and motion scores.

Hauptmann et al. [2006] employ five different types of retrieval components are considered: text, color, texture, edge-based retrieval, and person-x retrieval. These retrieval components are then used in a weighted combination to describe other concepts. (In their work, they assume all query concepts can be expressed as a mixture of their retrieval components.) In addition, they pretrain the weighted combinations for 14 semantic concepts such as “cars.”

Text queries are then used to start the search process. (Thus text metadata is assumed to be present.) For different queries, they can then determine which weightings would be most effective. On adding RF, these weights could be estimated to be similar to the most pertinent concepts. For example, if the user wanted to find "cars on a road", the pretrained weightings for the semantic concepts of cars and roads could be used to estimate a weighting effective for retrieval of both concepts. A drawback of this approach is that they assume the weighted combination of their features can capture all concepts. Another drawback is that their method of adapting weights to incorporate multiple semantic concepts relies on text annotations in data being searched.

Luan et al. [2008] present a method that adaptively chooses an appropriate feedback strategy to use in each round of feedback. These strategies utilize different features and scoring methods. Their method requires expert users that have a good understanding of the system to first perform various searches on a database while manually choosing the best strategy for any given feedback iteration. The usage statistics of the expert users are then used to train the adaptive strategy recommendation system.

These systems showed good ranking performance in their results. However, some can be quite complex given all the different components involved. For example, Hauptmann et al. [2006] used speech recognition to extract text as a feature from their videos. The adaptive strategy recommendation of Luan et al. [2008] required expert human users to train the system. If one were to apply these systems on larger scale problems, tuning the many parameters involved could be a daunting task. We therefore address the problem of search

using the basic approach of Tong and Chang [2001] where standard support vector machines (SVMs) [Burges, 1998] in an active learning framework were found to be effective for image retrieval.

2.5 Conclusion

As noted earlier, CBIR and CBVR share many of the same basic problems. The main difference between the two is that videos contain more information which can either help improve retrieval or make the learning problem more complex when there is limited training data. In this dissertation, we aim to improve upon the work of Tong and Chang [2001] which was found to be effective for CBIR and apply our improved framework to the retrieval of the more information rich videos.

Chapter 3

Experiments in Relevance Feedback

In this chapter, we conduct a series of experiments on our dataset of YouTube videos. These experiments test the effectiveness of different feature representations and also the general trends in different feedback solicitation methods.

3.1 Relevance Feedback Search Framework

We begin by describing the goals and details of the RF system we employ in the experiments to follow. The basic goal of our system is to be able to take in a handful of relevant (and irrelevant) examples of videos from a user and then use these examples to find other relevant videos from an unlabeled database (where metadata is assumed to be unavailable). If the retrieval results are not satisfactory to the user, he may elect to provide more feedback to the system so that retrieval results may be improved. This process of user feedback may be repeated until the user is satisfied with the retrieval results. The basic flow of the system is

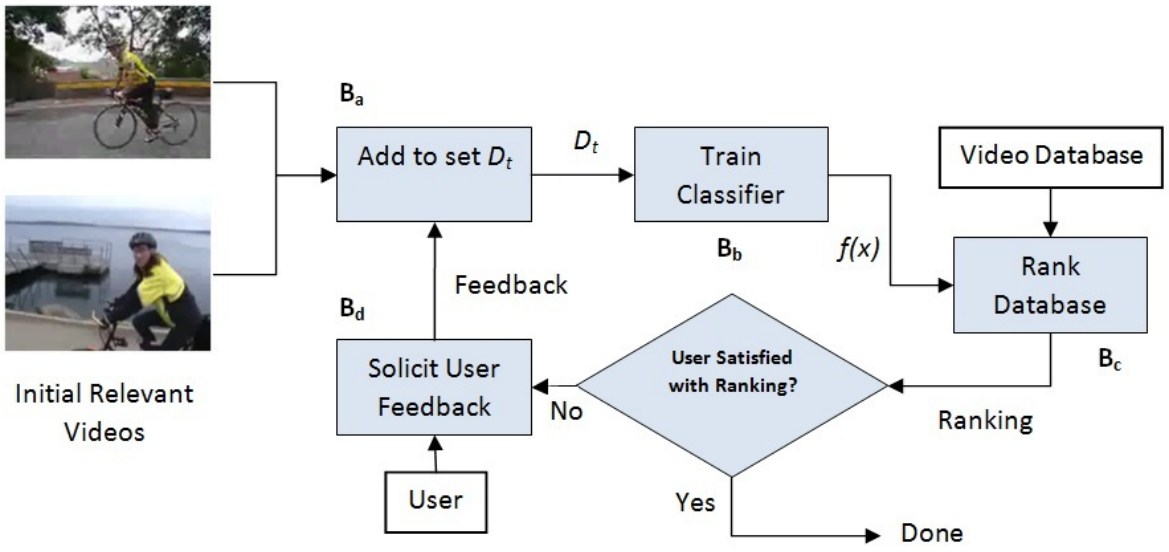

Figure 3.1: Flowchart of system. Set D_t is initially empty before execution. After the first execution of block $\mathbf{B_a}$, set D_t should consist only of the initial videos.

as follows:

(Please refer to Figure 3.1 for an illustration of the process.)

1. User submits a few initial query examples of relevant (and irrelevant) examples.

2. D_t is initialized to the be the set of query examples from the user. (Block $\mathbf{B_a}$.)

3. A classifier f (e.g. SVM) is trained based on the initial examples. (Block $\mathbf{B_b}$.)

4. The classifier f is used to rank the database. (Block $\mathbf{B_c}$.)

5. The top N ranked items from the database are then shown to the user.

6. If user is not satisfied with results, solicit feedback. Otherwise terminate. (Block $\mathbf{B_d}$.)

7. Add user feedback to set D_t (block $\mathbf{B_a}$) and continue from step 4 (block $\mathbf{B_b}$).

In our implementation, we use the SVM Kernel Machines Toolbox [Canu et al., 2005] to train an SVM classifier that is used as our ranking function. Specifically, the trained SVM $f(x)$ is used to score a video on relevance according to $score(\mathbf{x_i}) = \mathbf{w} \cdot \mathbf{x_i} + b$ where $\mathbf{w}$ is the normal to the SVM hyperplane, $\mathbf{x_i}$ is a video from the database (represented as a vector), and b is the bias term. Videos with higher scores are considered more relevant.

As for the feedback solicitation method, there are three that we investigate. The first one is for the system to solicit feedback on what it finds the most ambiguously relevant. This is equivalent to the approach taken by $\text{SVM}_{\text{Active}}$ [Tong and Chang, 2001] and was found to be highly effective for RF search in image retrieval. We call it uncertainty sampling. Another approach is to have the user give feedback on the top N retrieved items from the database. If the user labels a top ranked item as irrelevant, the system could potentially learn a lot. However, if a top ranked item is labeled as relevant, it may only serve to confirm what the classifier already knows. The last approach we consider is to solicit feedback from the user on random examples from the database. This approach has the advantage that feedback would be diversified (it would not be biased toward any particular examples).

3.2 Features Used

The next question is, how do we train SVMs on the videos? Learning from the videos requires converting videos into fixed-length vectors which commonly involves computing features. In our experiments, we investigate the use of Space-Time Interest Points (STIP) [Laptev

et al., 2008] and gist descriptors [Oliva and Torralba, 2001] for extracting information from videos. Our decision to test these two feature types is because they are complementary. STIP features are able to capture events at local points in space and time but do not capture global characteristics of the scene. Gist descriptors on the other hand do not capture local characteristics but can succinctly summarize the global characteristics of a scene.

We note that there are other feature types in the literature that could also be used. However, our goal in this work is in how RF search can be improved and not on how to get the absolute best ranking performance. Thus we only restricted our use of features to STIP and gist. Other work in the literature on image and video analysis such as [Hu et al., 2009, Liu et al., 2009a, Ryoo and Aggarwal, 2009] are complementary to our work and could be integrated into our framework.

STIP features are descriptors representing histograms of oriented spatial gradients (HoG) and/or histograms of optical flow (HoF) for "interesting" local regions in space and time. As described by Laptev [2005], this is done by considering where in a video large variations occur along both the spatial and temporal directions.

Of course, with the STIP features extracted, there are many ways to use these features in a learning system. Similar to Laptev et al. [2008], we extract STIP features from our videos and the build histograms of the STIP features based on a visual codebook. As fixed length vectors, these histograms can then be used in standard machine learning algorithms. In our experiments, we extracted[1] STIP features that used both HoG and HoF. We then con-

[1]Using the code from http://www.irisa.fr/vista/Equipe/People/Laptev/download/stip-1.0-winlinux.zip.

sidered the confidence ratings provided by the STIP extraction algorithm and removed all STIP features below 90th percentile in confidence relative to other features from the same video. So only the most confidently rated STIP features were used in all subsequent parts of our system. We found that using this simple filtering out of features effectively removed most features that were detected only due to camera shake. We then sampled 768,373 STIP features from a subset of the videos evenly representing each of the target categories in our database and used K-means to find 1,000 centers. These 1,000 centers were then used as our codebook for building histograms of STIP features. When building STIP histograms from each video, we treated the entire video as a single space-time volume and counted the frequencies of different feature types. Thus each video is represented as a 1,000 dimensional vector of frequency counts.

In addition to STIP features, we also investigated the use of gist descriptors [Oliva and Torralba, 2001]. The gist descriptor is a low dimensional representation of scenery that the authors call the Spatial Envelope. The gist descriptor encodes perceptual information about the naturalness, openness, roughness, expansion, and ruggedness of a given scene in a still image. Although gist descriptors only describe scenes in still images, we decided to test its effectiveness for our task because many videos from the same categories share the same types of scenery. In order to use gist descriptors in videos, we extracted still frames 10%, 50%, and 90% into each of the videos. These still frames were each resized to 128x128 pixels and a single gist descriptor was built[2] for each still frame. As a result, each video has three gist

[2]http://people.csail.mit.edu/torralba/code/spatialenvelope/gist.zip

descriptors associated with it, and the three gist descriptors are concatenated together into a single vector (of size 2,880). The effectiveness of STIP and gist either used independently or combined will be investigated in the experiments.

3.3 YouTube Dataset

Before presenting the experimental results, we will first describe the dataset used in our results. A portion of our dataset consists of the videos from the YouTube Action Dataset (YTA Dataset) [Liu et al., 2009a]. The YTA Dataset consists of about 1,600 videos[3] from 11 categories of actions/activities: basketball shooting, biking, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and dog walking.

While the YTA Dataset is challenging, we felt that the number of classes available were insufficient for our experiments. Thus we also added in our own collection of videos (downloaded from YouTube) to the YTA Dataset. In total, we added 1,983 videos with 12 categories. These categories are 100 meter dash, basketball, breast stroke, F1 racing, fighter plane, golf, grilling steak, hibachi, mixed martial arts, motorcycle Grand Prix, making sushi, and being at Yellowstone National Park. The difference between the basketball and golf activities we added in and those of the YTA Dataset is that the added videos are of sports in general (e.g. college basketball footage). The basketball and golf videos from the YTA

[3]Note that only 1,596 videos were used in our experiments because one was too short and we could not extract STIP features from it for the experiments.

Dataset are of single actions from the sports. In our experiments, we consider our general basketball and golf videos to be distinct from the basketball and golf videos in the YTA Dataset. (That is we assume that a user searching for one would not be interested in the other.) In all, our combined dataset consists of 3,579 videos with 23 classes and video run times ranging from 1 second to 24 minutes[4]. The average frame rate of the videos is 28.7 frames/sec. Our proposed dataset is ideal for testing an interactive system which could be used in a real application. This is because it uses actual videos downloaded from YouTube which results in the videos exhibiting a large amount of intraclass variation. In addition, we also included similar categories of videos such as "general basketball" vs. just "basketball shooting" to test the subjective distinctions that could be made by people. Sample frames and counts of videos per class can be seen in Figures 3.2 and 3.3.

[4] A humorous coincidence is that the longest video was of golf. Although other videos such as a jet fighter video also ran for close to 24 minutes.

Basketball Shooting (138)

Biking (145)

Diving (156)

Golf Swing (142)

Horse Riding (197)

Soccer Juggling (156)

Swing(137)

Tennis Swing (167)

Trampoline Jumping (119)

Volleyball Spiking (116)

Dog Walking (123)

Figure 3.2: Sample frame grabs from the YouTube Action Dataset. (Number of videos per class are shown in parenthesizes.)

100 Meter Dash (222)

Basketball (219)

Breast Stroke (375)

F1 Racing (115)

Fighter Plane (207)

Golf (184)

Grilling Steak (99)

Hibachi (101)

Mixed Martial Arts (113)

Motorcycle Grand Prix (100)

Making Sushi (102)

Yellowstone (146)

Figure 3.3: Sample frame grabs from our collection of YouTube Videos. (Number of videos per class are shown in parenthesizes. These videos are added to the YouTube Action Dataset for a combined dataset used in our experiments.)

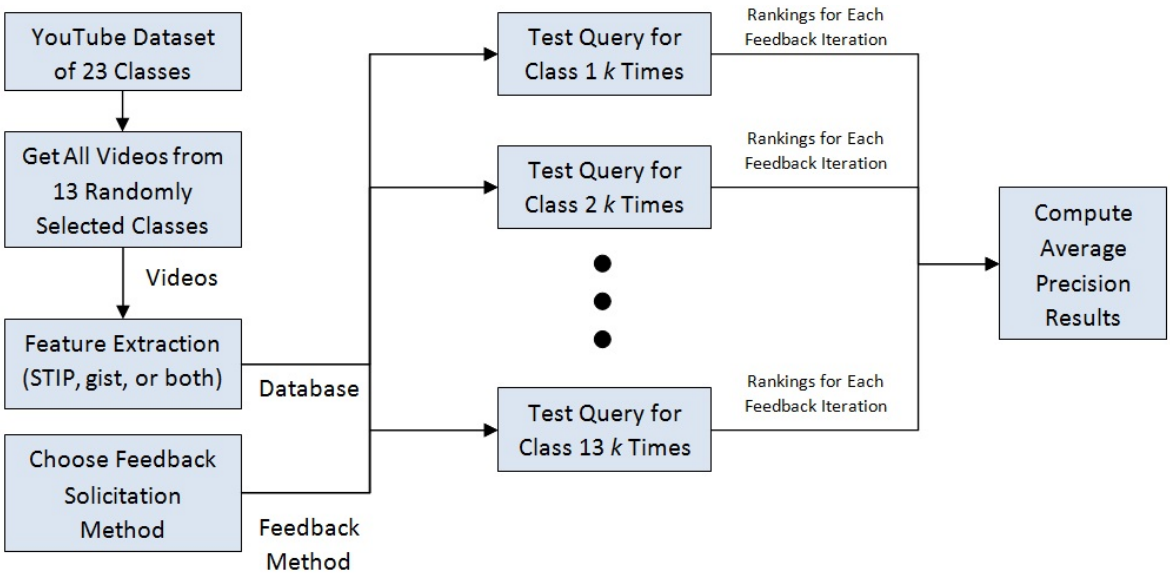

Figure 3.4: Flowchart of experimental setup where k = 10. The blocks labeled "Test Query for Class..." are equivalent to Figure 3.1. This flowchart illustrates one experiment and we repeat such an experiments for 10 choices of 13 class databases, different feature types, and different feedback solicitation methods.

3.4 Experimental Setup

In each experiment, we chose to either use STIP features, gist descriptors, or a combination of the two feature types (by concatenating STIP and gist vectors together). In a later chapter, we show experiments where a subset of the dataset is considered to be source data and the other subset, the videos to be searched. To make the results in this chapter comparable with later results using transfer learning, we conducted our retrieval tests on several subsets of the database. Specifically, we randomly select 13 of the 23 categories and refer to that subset as a database. We constructed a total of 10 such databases (each consisting of 13 categories). (Please see Table 3.1 for a list of the test databases used.) For each database, we performed 10 test search sessions for each category (using a simulated user that would label video relevance based on ground truth labels). For each test search session, the user

would submit initial five random relevant and five random irrelevant examples to the system. We would then run the search session for three iterations of user feedback where the system would select 10 examples based on one of the three feedback solicitation strategies in Section 3.1. For each of the rankings shown to the user, we measured the ranking performance and compared the results using different features and feedback solicitation strategies.

To measure ranking performance, we used average precision (AP) defined as

$$AveragePrecision = \frac{1}{num} \sum_{r=1}^{N} (P(r) \times \mathrm{rel}(r)) \tag{3.1}$$

where N = 50 in our experiments, num is the number of relevant items in the entire database, $P(r)$ is the precision at rank r, and $rel(r)$ is the indicator function for whether the r^{th} item in the ranking is relevant. In this metric, higher values are better. Note that since the least number of relevant examples in any category for our database is 99, we should not expect AP values higher than 0.51 in any test. This is because we only consider the ranking quality of the top 50 items. If all the top 50 items were relevant, the AP would be 50/99 which is less than 0.51. (Please refer to Figure 3.4 for an illustration of the experimental setup.)

In order to compare AP between different experiments, we could compute the mean of the APs from each experiment and compare them. However, the types of video categories we query for have a high level of variation in terms of difficulty. Thus the AP values would have high variance and comparisons using means and standard deviations in AP would be problematic. Also, our goal in this work is to determine how to get the best *improvement*

in AP and not how to get the best *absolute* AP, so it also makes more sense to observe how much improvement can be observed between competing approaches for each test query individually. In other words, if one were to submit the same initial query to two competing algorithms searching the same database, which one would perform better and by how much? To observe trends in performance differences, we decided to compute the percent change in AP between corresponding runs of the same queries across different algorithms (e.g. choices of feature type, solicitation method). For example, to compute the percent change from $test1$ to $test2$, we would compute $(AP_{test2} - AP_{test1})/AP_{test1}$ where AP_{test1} is the AP for test1 and likewise for test2. After computing the percent changes, to get a good idea of how these percentage changes are distributed over multiple test queries we use a quartile plot. The quartile plot shows the 25th, 50th, and 75th percentile changes in AP performance and gives a good idea of whether the performance change is generally skewed towards positive or negative values. For example, if we were computing percent changes in AP from algorithm1 to algorithm2 and the 25th percentile mark in the quartile results were at 10%, this would mean that in 75% of test queries, algorithm2 had at least 10% improvement over algorithm1. *Please note that in our quartile plots, the first iteration only consists of the initial query videos submitted by the user. Solicited user feedback is not incorporated until the second iteration.* In our tests of different algorithms, we compare algorithms using quartile plots over feedback iterations. (Note in cases where AP = 0, we changed these values to 0.001 so a meaningful measure of percent change could be computed. Setting 0 to 0.001 is reasonable since there is a negliable difference observed between rankings with AP of 0 or 0.001.)

DB1	DB2	DB3	DB4	DB5
100m Dash	B. Shooting	Basketball	100m Dash	100m Dash
Basketball	Biking	Dog Walking	B. Shooting	Breast Stroke
Breast Stroke	Dog Walking	Golf	Dog Walking	Diving
Diving	Fighter Plane	Golf Swing	F1 Racing	Dog Walking
Dog Walking	Golf	Grilling Steak	Fighter Plane	Fighter Plane
Golf	Golf Swing	Hibachi	Hibachi	Grilling Steak
Golf Swing	Grilling Steak	Making Sushi	Horse Riding	Horse Riding
Hibachi	Motorcycle	Martial Arts	Martial Arts	Making Sushi
Making Sushi	Soccer Juggling	Soccer Juggling	Soccer Juggling	Martial Arts
Motorcycle	Tennis Swing	Tennis Swing	Swing	Motorcycle
Swing	Trampoline	Trampoline	Tennis Swing	Soccer Juggling
Volleyball	Volleyball	Volleyball	Trampoline	Tennis Swing
Yellowstone	Yellowstone	Yellowstone	Volleyball	Volleyball

DB6	DB7	DB8	DB9	DB10
100m Dash	100m Dash	100m Dash	100m Dash	Basketball
Basketball	Basketball	Basketball	Basketball	B. Shooting
B. Shooting	B. Shooting	B. Shooting	B. Shooting	Dog Walking
Biking	Biking	Diving	Biking	Golf
Breast Stroke	Fighter Plane	Dog Walking	Breast Stroke	Golf Swing
Diving	Golf Swing	Golf Swing	Diving	Grilling Steak
F1 Racing	Grilling Steak	Grilling Steak	Fighter Plane	Hibachi
Grilling Steak	Hibachi	Horse Riding	Horse Riding	Horse Riding
Horse Riding	Swing	Martial Arts	Making Sushi	Making Sushi
Making Sushi	Tennis Swing	Soccer Juggling	Motorcycle	Motorcycle
Martial Arts	Trampoline	Tennis Swing	Tennis Swing	Swing
Trampoline	Volleyball	Trampoline	Trampoline	Tennis Swing
Volleyball	Yellowstone	Volleyball	Volleyball	Volleyball

Table 3.1: Different 13 Category Databases Tested On (Each column corresponds to a database and the rows lists the categories present in each database. "Basketball Shooting" was abbreviated to B. Shooting due to space constraints.)

3.5 Experiments in Choice of Feature Extraction

In this section, we explore which features are most effective for RF search of YouTube Videos. For our experiments, we tested using only STIP features, only gist descriptors, and a combination of both. As noted earlier, STIP and gist were combined for each video by concatenating its histogram of STIP features with its three gist descriptors (taken from three

still frames). As the STIP histograms and gist descriptors have very different ranges of numerical values, performing normalization of the values is helpful in combining them. This normalization was done by computing the standard deviation s_d for each dimension d of the vectors (based on all database items) and dividing each dimension d by s_d. So in addition to testing each feature independently and combined, we also tested how the system would perform if they were normalized. For the choice of the user feedback solicitation method, we chose to fix it to uncertainty sampling here. This is because uncertainty sampling (same as $\text{SVM}_{\text{Active}}$) has been established in the literature as effective for RF search. After establishing which features are most effective, experiments in user feedback solicitation methods will be shown for the best choice of feature types.

In our experiments we first tested the effectiveness of STIP features versus gist descriptors (without normalization) and found that gist descriptors were more effective than STIP features for RF search. The quartile plot of this test can be found in Figure 3.5(a). It can be seen from the 50th percentile line, that out of half of all test queries, use of gist resulted in at least a 225% improvement in AP over all user feedback iterations. The 75th percentile indicates even greater improvements. The most degradation in AP seen from the 25th percentile line is that the first feedback iteration's 25th percentile percent change had a 50% loss in AP. The large percent improvements are likely observed because many of the video categories are well correlelated with certain settings. For example, tennis swings usually take place on tennis courts. The gist descriptor was specifically built to describe many known characteristics of scenery so it is not surprising that without much training, gist is able to classify

the scenery observed in videos reasonably well. STIP features on the other hand only try to capture "interesting" local events in space-time but does not have higher level characteristics built into it.

Despite gist outperforming STIP, it still might be beneficial to combine them. This is because gist only looks at static frames and does not capture any temporal information. As described earlier, we combined gist and STIP by concatenating their descriptors together. However it can be seen in Figure 3.5(b) that using gist alone was still better. Despite the poor performance of combining STIP and gist, this still does not disprove the benefits combining them. As mentioned earlier, the STIP histograms and gist descriptors are very different ranges of numerical values. Thus normalization of the values could be beneficial to combining them. In Figure 3.6(a), we see that overall the combination of STIP and gist with normalization outperforms gist alone. Although the 25th percentile line indicates some degradation in performance when STIP is added, the 75th percentile line indicates a much greater level of improvement. For completeness, we also tested the normalized combination of STIP and gist against normalized gist. Results can be seen in Figure 3.6(b). It can be seen that even after normalizing gist, the normalized STIP and gist combination is better. Figures 3.6(c) and 3.6(d) show the results of comparing the normalized STIP and gist combination to STIP and normalized STIP. It is clear that STIP alone performs worse in both cases. Thus the normalized combination of STIP and gist has been shown experimentally to be the best combination of features for ranking performance.

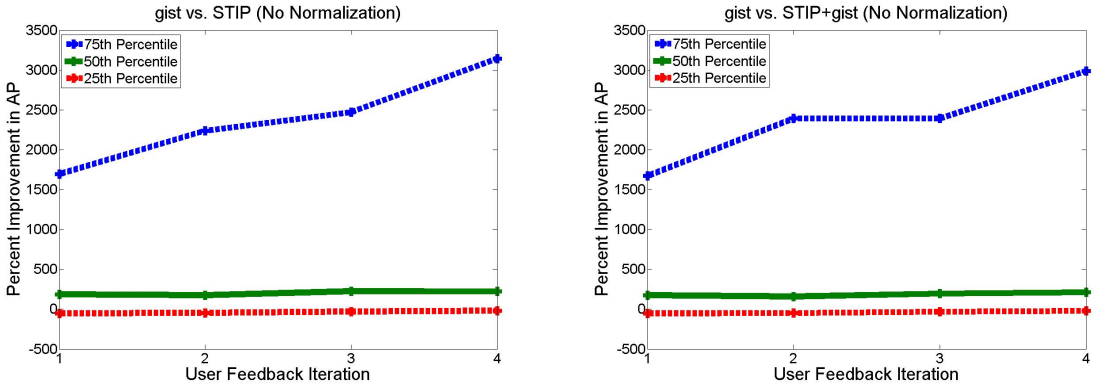

Figure 3.5: Quartile Plots of Percent Improvements using gist

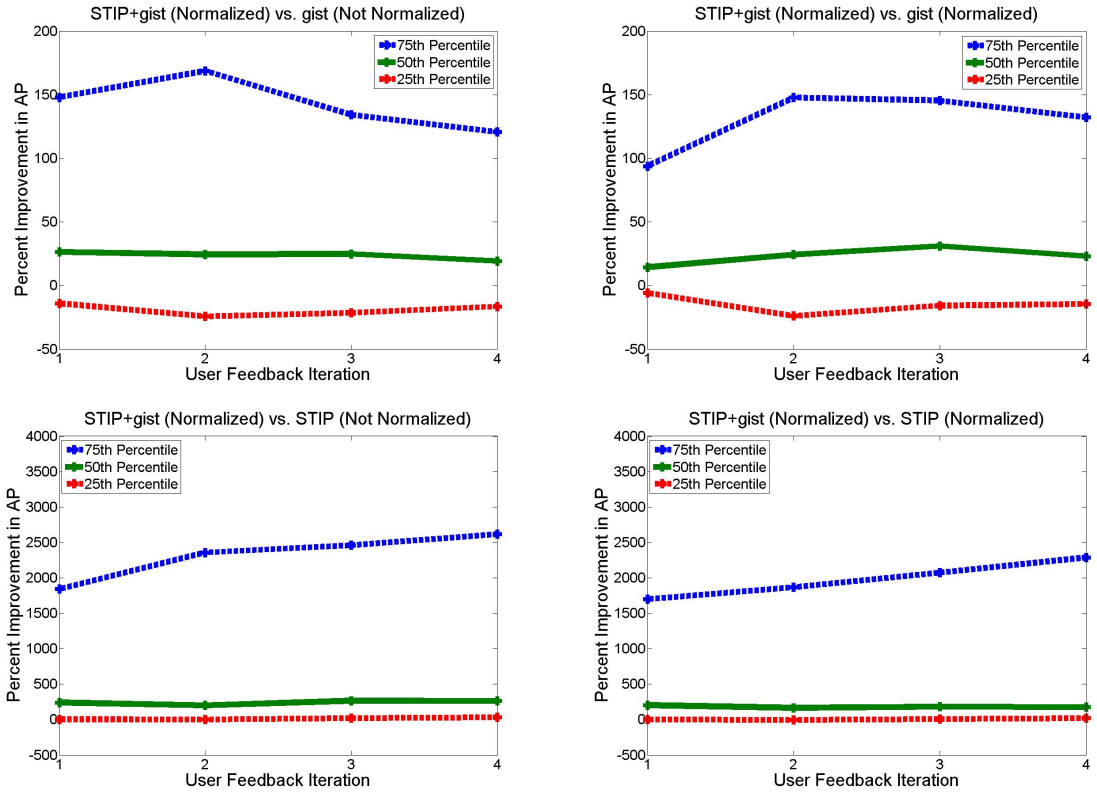

Figure 3.6: Quartile Plots of Percent Improvements using STIP+gist

3.6 Experiments in Choice of User Feedback Solicitation Method

The previous section performed experiments in different choices of features under the assumption that uncertainty sampling of user feedback solicitation was the best one. While Tong and Chang [2001] showed uncertainty sampling to be highly effective, we conduct a series of tests to verify this. In this section, we compare three user feedback solicitation methods we call the "top ranked," "uncertainty sampling," and "random" methods. The top ranked method is where the system takes user feedback on the top k videos. Uncertainty sampling solicits feedback on the k videos nearest the SVM hyperplane (the most ambiguously relevant). The random method simply randomly selects k videos from the database. In all tests, we now fix the type of features used to the normalized STIP and gist combination and set k = 10. Figure 3.7 shows the results in comparing uncertainty sampling and the top ranked method. It can be seen from both graphs that for most feedback iterations, the two methods are comparable. It is not until the last feedback iteration that an advantage can be seen in using uncertainty sampling over the top ranked method. In looking at Figure 3.8, we can seen that although there might be a small advantage in using the random method over uncertainty sampling for the second feedback iteration (where user feedback is first incorporated into learning), there is a clear advantage in favor of uncertainty sampling for all subsequent feedback iterations. Comparing the top ranked and random method we can see from Figure 3.9, that the random method has a slight advantage initially while the advantage

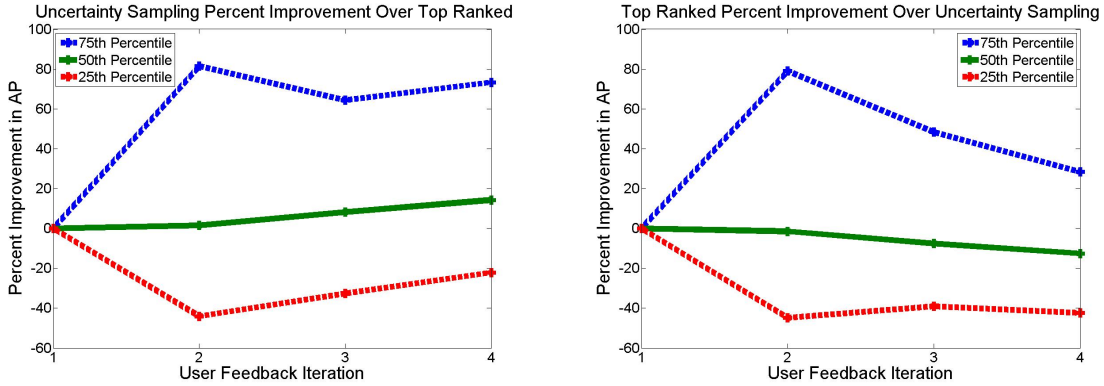

Figure 3.7: Quartile Plot of Percent Improvement between Uncertainty Sampling and Top Ranked

Feedback Iteration	1	2	3	4
Uncertainty Sampling	0.06 ± 0.06	0.09 ± 0.07	0.12 ± 0.08	0.15 ± 0.09
Top Ranked	0.06 ± 0.06	0.09 ± 0.07	0.11 ± 0.07	0.13 ± 0.08
Random	0.06 ± 0.06	0.09 ± 0.07	0.10 ± 0.07	0.12 ± 0.07

Table 3.2: Mean AP over Iterations for Different User Feedback Solicitation Methods

is gradually skewed in favor of the top ranked method in later feedback iterations.

These results indicate little difference between the different feedback solicitation methods in earlier feedback iterations and small differences are only seen in later iterations. Indeed, Table 3.2 also suggests the same trends observed in the quartile plots. We attribute the similarity in performance to the complexity of the classification tasks. If a classification task is too difficult, it may be that the amount of feedback for sufficient learning was too small. So regardless of the solicitation method used, overall there would be little difference in performance. Thus as more iterations go by, with more training data, we see some differences in the solicitation methods.

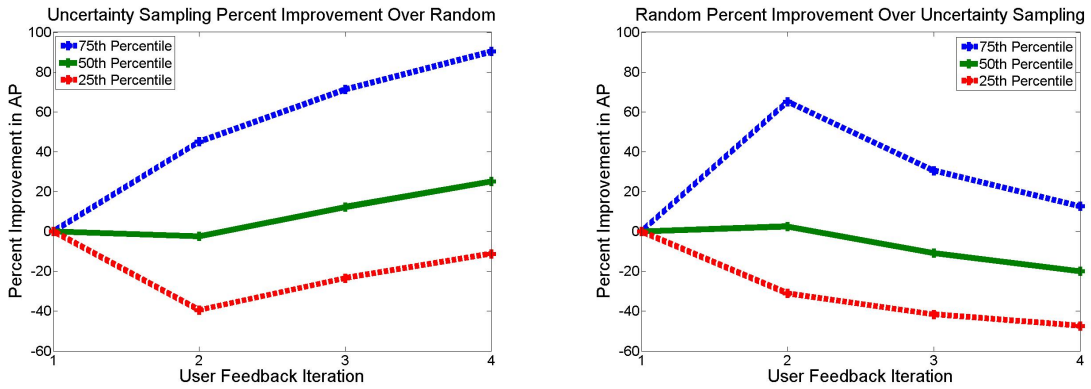

Figure 3.8: Quartile Plot of Percent Improvement between Uncertainty Sampling and Random

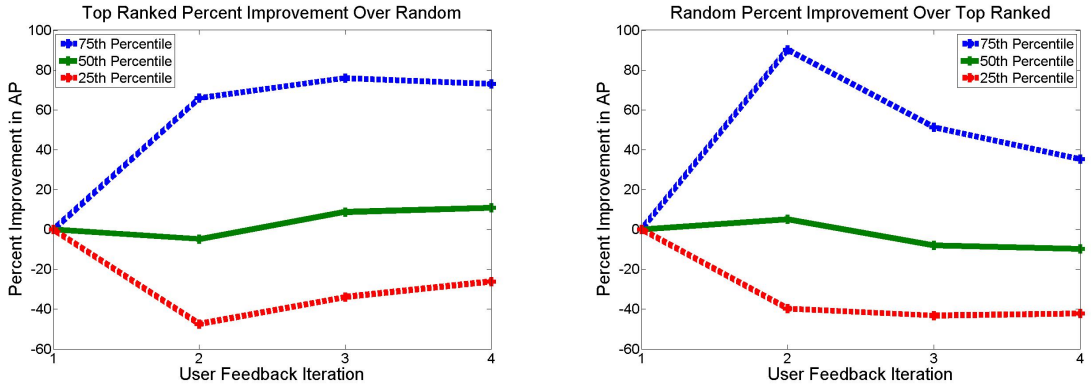

Figure 3.9: Quartile Plot of Percent Improvement between Top Ranked and Random

3.7 Conclusion

In this chapter, we explored different feature extraction and representations for RF search. We also investigated the use of different feedback solicitation methods and found only a slight improvement in using uncertainty sampling over the top ranked and random methods. We attribute this to the complexity of the classification tasks and the scarcity of training data from user feedback. There are some possible solutions to the data scarcity problem. One is to somehow increase the amount of training labels obtainable from the user. This could be done by simply requiring the user to label more videos but this would require too much labor

on the user's part. Another solution is to use clustering [Liu et al., 2008] to diversify the user obtained labels and hope to gain more information as a result. While clustering can help with RF by reducing the amount of redundant data learned from an iteration of user feedback, if a classification task is too complex, learning can still be limited. What is lacking in most RF systems presented in the literature is knowledge about the world. Thus we propose the use of transfer learning (TL) to take preexisting auxiliary knowledge about different classification tasks and use this information to improve learning generalization on just a handful of labels from the user. We begin the next chapter with an overview of TL.

Chapter 4

Related Work in Transfer Learning for Relevance Feedback

4.1 Transfer Learning

Transfer learning (TL) Pan and Yang [2009] is a machine learning formulation where knowledge learned from one or more classification tasks is transfered over to a target task where the target task training data is scarce. If the abundant training data of source task(s) are *related* to the target task, it can be used to bias the classifier for the target task so that generalization performance can be improved.

As an example, consider the related classes "volleyball" and "basketball" (see Fig. 4.1). Say we are interested in classifying whether videos are of basketball but the amount of training data available is very limited. If the amount of training data for the task of classifying

Figure 4.1: Example of similarity between two different classes. If training data for "volleyball" were abundant while training data for "basketball" were scarce, the knowledge on classifying volleyball could be used to supplement the basketball training process.

"volleyball" versus "not volleyball" were abundant, the knowledge from the volleyball classification task could be used to supplement the training of the basketball task in order to improve generalized accuracy on classifying basketball videos. Since volleyball and basketball are similar in that they share some characteristics, there should be a way to utilize the extensive knowledge of volleyball to help learn more about basketball. However simply including volleyball data in the training process would result in a classifier for volleyball and not basketball. (Since there would be much more data for volleyball.) The solution is to limit the amount of bias introduced by the source data to prevent overfitting to the source data. Thus with the right balance between bias and variance, the benefits of knowledge from the source data can be utilized to better learn the target task. (There are many standard approaches to limiting source data bias and we detail the approach adopted in this work in Chapter 5.)

Provided system designers built a set of source task datasets for a variety of classes (this set of classes would only account for a small fraction of possible queries users could make), we could use the source data within a TL framework and combine it with RF to reduce the amount of needed user feedback. However one of the key issues in combining RF and TL is determining which source task(s) are *related* to the target query. This is one of the main contributions of our work and is also a less explored problem in TL [Pan and Yang, 2009].

4.2 Basics of Concepts in Transfer Learning

Before discussing related work in TL, we introduce a few TL concepts to provide context. In TL, there can be different relationships between the source and target tasks. Let task S be the source task and D_s be the source training set and task T be the target task and D_t be the target task's training set (where $|D_s| >> |D_t|$). Then TL can be subclassed into the following scenarios of interest:

1. S and T have different distributions over their data but the source and target classes could be the same (e.g. "running"). This is called the *Cross-Domain* problem in some work. As an example, if the training data D_s had been collected with camera A and D_t had been collected with camera B, simply combining D_s with D_t to improve classification accuracy on videos taken with camera B may not work well. (The cameras may have been positioned differently or have other differing characteristics.) The goal is to adapt the knowledge from D_s to augment the knowledge from D_t.

2. S and T classify different but related classes. For example, S could be "volleyball" and T could be "basketball" (see Fig. 4.1). Since task S is related to task T, it should be possible to use the knowledge learned from D_s to improve generalization on D_t. This is the problem we focus on in this work and we call it the *Cross-Class* problem to distinguish it from past *Cross-Domain* work where the source and target classes are the same.

There are more relationships between source and target tasks in TL described by Pan and Yang but the above mentioned ones are the most pertinent to our discussion.

4.3 Transfer Learning in the Presence of Multiple Source Tasks

There has been much work in the literature on TL in numerous domains [Pan and Yang, 2009]. However, in order to utilize TL learning in RF search, there is a basic characteristic a TL algorithm would need to have. The TL algorithm would need to be able to transfer knowledge in the presence of multiple source tasks. For TL to be of any use in RF, we need to have a set of source tasks from a variety of categories in order to be able to improve on a large variety of user queries. We survey a few algorithms which were designed to work with multiple source data.

Quattoni et al. [2008, 2009], proposed finding jointly sparse solutions for a set of related image classification tasks. The idea is that if a set of tasks are related then they probably share

many of the same prototypes. The hope is that these shared prototypes would allow for better generalization. However, since this algorithm is a multitask learner (all training data are use to learn the tasks jointly), performing knowledge transfer between the set of source tasks and the query in question could be computationally expensive. This is especially problematic in an RF scenario where we expect a large variety of source tasks and to possibly use TL for multiple rounds of user feedback.

Yang et al. [2007] and Yang and Hauptmann [2008], propose the Adaptive-SVM for regularizing a target SVM hyperplane to be similar to a related source hyperplane while still being able to fit the scarce target training data. The particular problem they focus on is the case when the source and target tasks both classify for the same class but the distribution of the examples is different between the source and target. This is illustrated by their detection of concepts such as "anchor" or "weather" between news programs on different TV stations. The editing style and camera work of different TV stations could cause the data for the same classes to be distributed differently. To find the relevant source tasks, they determine which source classifiers have the best estimated performance on the target class. While their work focuses on the shift of distribution between the same class, some of their ideas (with a bit of adaptation) could still be applied to our Cross-Class problem so we explore this later in the dissertation.

Yao and Doretto [2010] extended the Transfer Adaboost (TrAdaBoost) [Dai et al., 2007] algorithm to learn from multiple source datasets where some of the source tasks can be unrelated to the target. They first introduce their MultiSourceTrAdaBoost algorithm which is

essentially TrAdaBoost extended to choose the best source task with respect to the weighted error of the target data at each round. When the number of source tasks is one, MultiSourceTrAdaBoost reduces to TrAdaBoost. The second algorithm they introduce is TaskTrAdaBoost. TaskTrAdaBoost simply involves pretraining all the source tasks independently using standard AdaBoost and then using the boosting framework to build a weighted linear combination of source classifiers that would classifier the target data well. There is no actual training of classifiers for the target data in this case. This work has the advantage that it was designed to adaptively select good source tasks to transfer from so unrelated tasks can be included in the set of source tasks. A drawback of MultiSourceTrAdaBoost is that a large amount of training involving all the source data has to be done at each iteration. TaskTrAdaBoost would be much more efficient as no training is actually done when learning to classify for a target task. This also has the advantage of not overfitting the small amount of target data but suffers from performance losses compared to MultiSourceTrAdaBoost when more target data is available. As the speed of MultiSourceTrAdaBoost would make it too inefficient for real-time RF, we will focus our attention on how TaskTrAdaBoost would perform in our RF work.

4.4 Combining Transfer Learning and Relevance Feedback Search

The current literature on transferring from source tasks to a target task for RF is focused on the Cross-Domain problem. Specifically for the case where one wishes to learn a classifier for the same class but the source task's abundant training data's distribution does not match the target task's distribution. Thus this is a special case of transfer learning. We briefly discuss a few examples of such work.

Setz and Snoek [2009], present a study on how social tagged images could aid video search. They use the photo sharing website Flickr to download 87K tagged images, extract Weibull and Gabor features, and use Linear Discriminant Analysis to learn classifiers for the tagged concepts. They then test these classifiers on searching video frames from the 2008 TRECVID dataset [Smeaton et al., 2006]. They also manually disambiguate the social tagged images so they will be more consistent and compare the performances of using classifiers trained based on the original social tags versus the disambiguated tags in Cross-Domain classification. Not surprisingly, the disambiguating tags results in better average precision over social tags. However, their experiments do indicate that classifiers trained on the social tagged images still perform reasonably well for Cross-Domain RF. However it is unlikely their framework would be applicable to our problem of transferring from one task to another because they never adapt the source classifiers to their target domain.

Liu et al. [2009b] present two Cross-Domain learning methods for RF. The first method involves an equal weighted linear combination of the source and target classifier outputs (after normalization to values in [-1,1]). The second method involves solving a regularized regression problem where they assume a linear classifier. The regularized regression problem aims to jointly minimize the empirical risk of the target data, the disagreement between the source and target classifiers, and a term is added in to control the complexity of the target classifier. The authors also propose a hybrid combination of their two methods where a heuristic is used to decide which method to use in a given round of feedback. Interestingly, the performances of either method are roughly the same but the hybrid method is overall better than either one by itself.

While RF is not used, Geng et al. [2009], present relevant work which is differentiated from Adaptive-SVM [Yang et al., 2007, Yang and Hauptmann, 2008] in that they focus on learning ranking functions rather than classifiers. They apply their Ranking Adaptive SVM to the problem of Cross-Domain searching text based data. While ranking could potentially be more suited to Information Retrieval than classifiers, it may not be as well suited to RF. One of the most well-known active learning strategies is to query for feedback on the most ambiguous items in terms of relevance as described by Tong and Chang. However with a ranking function, it is unclear which items are most ambiguously relevant since there is no obvious threshold value for classifying relevance.

Other recent work Cao et al. [2010], Duan et al. [2010], Saenko et al. [2010] present new mechanisms for Cross-Domain transfer of video actions and events. However, they do not

present methods for source task selection. However as mentioned earlier, our focus is on the Cross-Class transfer case and so use of these proposed mechanisms for TL may not be suited to our problem. Furthermore, these work focus on development of the transfer mechanism while our work (as will be seen later) focuses on the selection of what to transfer. Which is crucial to RF search.

4.5 Conclusion

RF is a framework that has existed since early 1970s [Rocchio, 1971] and there are a host of algorithms for TL [Pan and Yang, 2009]. A basic problem in RF is the scarce amount of training data due to limited user feedback. It would seem obvious we could apply TL to RF in order to alleviate the training data scarcity problem. However such work is uncommon. In fact, the RF surveys [Crucianu et al., 2004, Ruthven and Lalmas, 2003, Zhou and Huang, 2003] do not mention the use of TL. As noted earlier, recent work on RF and TL appears to be limited to the Cross-Domain case and there is no work in the Cross-Class case. In this work, we make one of the first explorations of the Cross-Class TL problem in RF search with an emphasis on the source task selection problem. We introduce our framework in the next chapter.

Chapter 5

Extending Relevance Feedback with Transfer Learning

Assuming that images or videos are represented as fixed-length vectors (as described in a previous chapter), we can use SVMs to learn scoring functions for ranking databases based on relevance with respect to user queries. As described in chapter 4, if a *target task* has only a scarce amount of training data, its performance can be improved by transferring knowledge from a *related source task*.

5.1 Transferring Knowledge from a Source Task to a Target Task

Let D_s and D_t be the training data for source task S and target task T respectively. (Where $|D_s| >> |D_t|$.) Then ideally if the source and target tasks were the same, we could just train a more powerful classifier for the target task by augmenting D_t with D_s. In practice, the source and target tasks are unlikely to be the same but they could still be related. Then we could still augment D_t with D_s but with less weight given to the data in D_s.

We accomplish this by adjusting the C parameter in the SVM formulation. Recall that training an SVM involves solving the following optimization problem:

$$\min_{\mathbf{w},\xi}\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i\} \tag{5.1}$$

$$\text{s.t. } y_i(\mathbf{x_i}\cdot\mathbf{w}+b) - 1 + \xi_i \geq 0,\ \xi_i \geq 0$$

where $\mathbf{x_i}$ is the i^{th} datapoint and y_i and ξ_i are the label and slack variable associated with $\mathbf{x_i}$. $\mathbf{w}$ is the normal to the hyperplane. C is the parameter that trades off between training accuracy (high C) and margin size (low C).

Let D_{aug} be D_t augmented with D_s and let the data from D_s be indexed from 1 to n in D_{aug} while the data from D_t be indexed from $n+1$ to $n+m$ in D_{aug}. Then to weight the

source data and target data in the SVM training of D_{aug} we solve the following:

$$\min_{\mathbf{w},\xi}\{\frac{1}{2}\|\mathbf{w}\|^2 + C_s\sum_{i=1}^{n}\xi_i + C_t\sum_{i=n+1}^{n+m}\xi_i\} \tag{5.2}$$

$$\text{s.t. } y_i(\mathbf{x_i}\cdot\mathbf{w}+b) - 1 + \xi_i \geq 0, \xi_i \geq 0$$

where all the variables are as described in Eqquation 5.1 and C_s and C_t are the different parameters trading off the "hardness" versus "softness" of fitting the associated datapoint. (Note that we set $C_s < C_t$.)

In our experiments, we saw little difference between using all the source data to bias the target SVM and using just the support vectors from the source SVM. Since using only the support vectors results in faster training speeds, we train only on the source task support vectors in our implementation.

The A-SVM [Yang and Hauptmann, 2008, Yang et al., 2007] could have been used in place of this section's proposed method of transfer (which they call the "aggregate approach"). However the A-SVM does not offer benefits in improved accuracy over the aggregate approach and can even perform worse in some tests. The main advantage of using A-SVM is shortened training time. As the focus of this paper is on the feasibility of combining RF and TL for improved accuracy and the aggregate approach is more standard, we chose to use the aggregate approach.

5.2 Source Task Selection

Section 5.1 assumed we knew which source classifier to transfer from. However, transferring from the wrong classifier can hurt performance on the target task. In this section, we first present our method for selecting from a set of source tasks. We then discuss some other methods of source task selection which will be compared against our method later.

5.2.1 Score Accuracy

If a task S_1 and task S_2 were the same (and each task had a plentiful amount of training data), we would expect that the classifier f_1 learned from the training data of task S_1 would perform well on the test data of task S_2 and vice versa. Similarly, if tasks S_1 and S_2 had been *related*, we would expect classifier T_1 to have decent performance in task S_2 and vice versa. Certainly, if an *unrelated* task S_3 were present, f_1 would be expected to perform poorly on task S_3. This intuition suggests a metric for determining the relatedness of tasks. If we trained classifier f_1 for task S_1, we could use f_1 to classify the training data of tasks S_2 and S_3. Since we know the ground truth for tasks S_2 and S_3, we can then determine how accurate f_1 is on each task. From there, the task f_1 is more accurate on indicates which task is more related to S_1. So if we chose the task f_1 is most accurate on, this would likely be the best available source task for TL. Using this proposed metric, we propose the following algorithm for choosing the most related source task for TL:

1. Given target task training data D_t, train an SVM f_t.

2. For each source task S_i, determine the classification error on the source task training set D_{si} with respect to SVM T_t.

3. Choose the training set D_{si} with the lowest error as the source task data to transfer.

We call this the **Score Accuracy (SA)** method.

5.2.2 Score Clustering

In Yang et al. [2007], a number of strategies for choosing which source classifier to transfer from were presented. One method was to use score aggregation from multiple source classifiers. The basic idea was to use the "average" of multiple source classifiers with the hope that this would result in a more accurate classifier for assigning pseudo-labels to the unlabeled data. These pseudo-labels would then be used to evaluate how much individual source classifiers help improve ranking performance on the unlabeled examples. This approach does not work in our case. Since the authors were transferring knowledge in a Cross-Domain setting, all the source classifiers were assumed to classify for the same class. In our case, the source classifiers can be very unrelated to each other and thus combining an "average" of the source classifiers results in very poor performance.

Another proposed method was to assign scores to all unlabeled items using a potential source classifier (one trained on source data) and use the Expectation Maximization (EM) algorithm [Dempster et al., 1977] to fit two Gaussian components to the scores. If the scores

separate the data well then the means of the found Gaussian components should have greater distance between them. While a good idea, this is still not directly applicable to our problem because the target data are never used in this process; thus the same source classifier would always be selected regardless of the user feedback. However, if we first transfer the source classifier to the target classifier and then use the resulting classifier to score the unlabeled data, EM can be used to determine how well the transferred classification separates the data. We use this new procedure for determining which source classifier would help the target classifier produce the best separation of items in the database.

Formally, let D_{si} and D_t be the i^{th} source and target training data and let $TL(D_{si}, D_t)$ be a function that produces a classifier where D_{si} was used to transfer knowledge to the target task (as described in Eq. 5.2).

Then the following steps are taken to determine the best source to transfer from:

1. For each source task S_i:
   (a) Produce SVM $f_{si} = TL(D_{si}, D_t)$.
   (b) Use SVM f_s to compute scores Sc on the unlabeled database.
   (c) Use EM to fit Gaussian components $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$ to scores Sc.
   (d) Determine the distance $d_i = (\mu_1 - \mu_2)^2$.
2. Choose the source task with greatest d_i.

The distance d_μ can be used to indicate how well transferring the given source task to the target task would separate the unlabeled data (larger values are better). This provides

an indication of whether the source task helps improve target task classification. The same procedure can be used to score the transfer for each of the available source tasks and the best source task could be chosen as the one to transfer from. We call this the **Score Clustering (SC)** method.

5.2.3 Max-Margin and Other Methods

The SC method uses EM as a component in determining what source task is best for TL. However, EM can be computationally expensive and its use is not ideal in an interactive search framework. Thus an alternative to SC might be to determine the separation of the two nearest datapoints on opposite sides of the hyperplane. (Rather than the separation of the means from the scores.) This can be done by computing the distance between the scores of the two positively and negatively classified examples nearest the SVM hyperplane. We call this the **Max-Margin (MM)** method. Comparisons to MM will be done in the experiments.

Another method we explored was a leave-two-out cross-validation procedure. The procedure worked as follows:

1. For N times do the following:
   (a) Randomly select one relevant example p and one irrelevant example n from the target training data.
   (b) For each source task S_i:
      i. Use TL to produce SVM $f_{si} = TL(D_{si}, D_t)$.

ii. Determine how much f_{si} separates examples p and n

(c) Add a vote for the source task S_i that best separates examples p and n.

2. Select the source task with the highest number of votes.

The problem with this cross-validation procedure is that it was too computationally expensive to feasibly run extensive experiments in a reasonable amount of time. Smaller scale tests also suggested that the SA and SC methods were superior in ranking performance. Thus we do not compare this method against the others. It is mentioned here as a reference for interested readers.

5.3 Integrating Relevance Feedback with Transfer Learning

With the mechanisms of knowledge transfer and source task selection methods defined, we now integrate these two components into our RF search framework. The basic idea is to use whatever data is available in the target training data D_t at each stage of RF and select the best source task for TL (this selection is made independent of selections from past iterations). TL would then be used to train a classifier f_{si} (biased with the source task data) and use it to provide a ranking of the database. (Classifier f_{si} would also be used to select examples for feedback solicitation from the user for the next round of RF.) The complete procedure for using TL in RF is

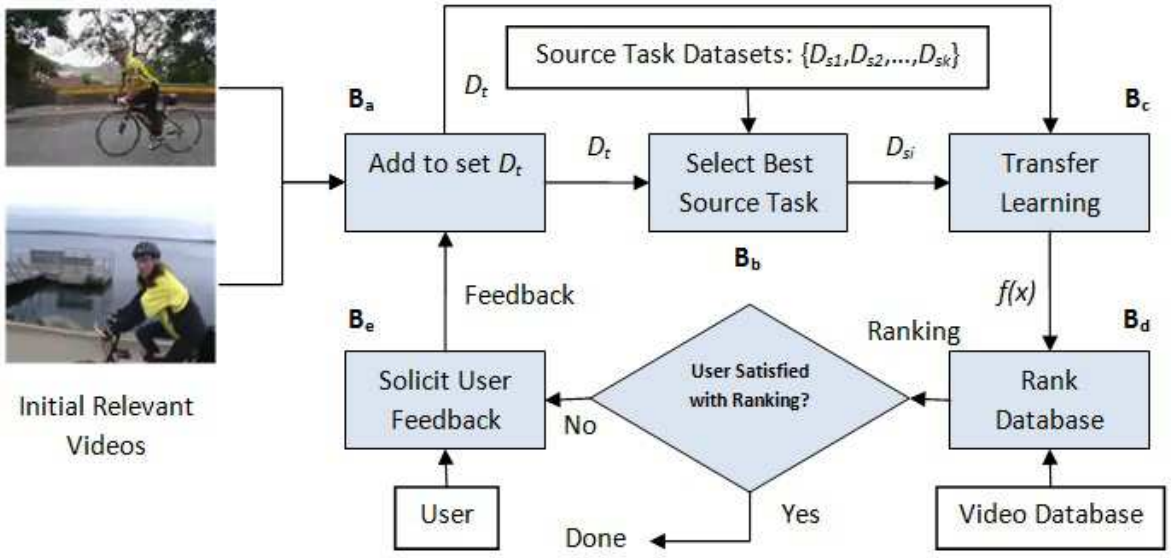

Figure 5.1: Flowchart of system. Set D_t is initially empty before execution. After the first execution of block $\mathbf{B_a}$, set D_t should consist only of the initial videos.

1. Let D_t be an empty set.

2. User submits a few initial query examples of relevant (and irrelevant) examples.

3. The initial query examples are added to set D_t. (Block $\mathbf{B_a}$.)

4. The source task datasets and D_t are then used in our algorithm (Sec. 5.2) for finding the best source task to transfer from. (Block $\mathbf{B_b}$.)

5. The best source task's training dataset and training data in D_t are then used in TL to obtain a classifier f for ranking the database. (Block $\mathbf{B_c}$.)

6. The classifier f is used to rank the database. (Block $\mathbf{B_d}$.)

7. The top N ranked items from the database are then shown to the user.

8. If the user is satisfied with the results, the process terminates. Otherwise the system solicits user feedback. (Block $\mathbf{B_e}$.)

9. The user feedback is added to set D_t (block $\mathbf{B_a}$) and the process continues from step 4 (block $\mathbf{B_a}$).

5.4 Comments on Source Task Selection Methods

We note that projecting all source training data onto the subspace of the unlabeled database was a helpful preprocessing step for determining what to transfer (for all methods). Thus we first performed principal components analysis (PCA) on the unlabeled database items to obtain a set of basis vectors $\mathbf{V}$ with non-zero eigenvalues. We then projected *all* source task data *and* unlabeled database items onto $\mathbf{V}$. So in our implementation, the projected data were used in all steps.

As discussed earlier, the SC method is an extension of a method from the literature for source task selection in the Cross-Domain transfer problem. Thus SC may be an effective tool for source task selection in our general TL scenario. However, SC has some drawbacks. The need to perform TL between all candidate source tasks and the target task added with the need for EM to be used to evaluate the transfer is computationally expensive. This problem is made worse by the need to repeat source task selection for each iteration of feedback. In our Matlab implementation, the total time used for processing in query sessions with three rounds of user feedback takes about 30 seconds to complete. The MM method is our approximation

to the SC method and requires slightly less processing time. (About 25 seconds compared to 30 seconds.) Later, we make comparisons to MM to establish whether the clustering criterion of SC is preferred.

SA is a simpler method than SC and only requires training on the small target training set (which would be computationally cheap) and then performing TL only once with the selected source task. In our implementation, the total training time used for processing in the same query sessions as tested with SC only require about 5-6 seconds. (Our implementation is not optimized for speed.) Thus considering only processing time, SA is a better candidate for real-world applications. As we shall see later in the experiments, SA also provides superior ranking performance over SC in a real-world dataset of Internet videos.

Chapter 6

Experiments in Relevance Feedback using Transfer Learning

In this chapter we show experimental results on our proposed RF and TL framework. In particular, we make comparisons between the different source task selection methods presented in the previous chapter. The experimental setup and datasets used are identical to what was presented in Chapter 3 with one exception. For each of the 10 randomly selected 13 class databases, we use the 10 classes that were left out as source tasks for TL. For example, if we consider DB1 from Table 3.1, the 13 classes in the database are 100m Dash, basketball, breast stroke, diving, dog walking, golf, golf swing, Hibachi, making sushi, motorcycle Grand Prix, swing, volleyball spiking, and Yellowstone National Park. Then the 10 source tasks used to test queries on DB1 are basketball shooting, biking, horse riding, soccer juggling, tennis swing, trampoline jumping, F1 racing, fighter plane, grilling steak, and mixed martial arts.

6.1 Feedback Solicitation Experiments Revisited

In Chapter 3, we saw that the choice of feedback solicitation method did not make much of a difference in the ranking performance of early user feedback iterations. It was not until the later iterations that some differences in performance could be seen. We hypothesized the lack of significant difference in performance between the different choices of feedback method was due to insufficient amounts of training data from too little user feedback. Thus the additional data from user feedback (regardless of solicitation method) would not have as much effect on classification performance. Since TL aims to alleviate the issue of training data scarcity, if one were to reevaluate the choice of user feedback method, there should be a clearer indication that uncertainty sampling is best. In this section, we reevaluate the choice of feedback method in the RF and TL scenario.

In Figures 6.1 – 6.3, it can be more clearly seen that uncertainty sampling is the best of the feedback solicitation methods. Table 6.1 also agrees with this trend. Thus TL appears to have alleviated some of the problems with scarce training data and we see the expected trend that uncertainty sampling is best. However, what is interesting is that the random and the top ranked feedback solicitation methods appear to be comparable to each other. There may be a reason for the lackluster improvement of the top ranked method. In selecting a new source task to bias the SVM decision surface for each iteration, the decision surface could be fluctuating more rapidly over iterations (than it normally would without TL). If this is the case, then the labeled top ranked items from previous iterations may not be as helpful

for training the current decision surface (since it could be very different from the previous iteration's decision surface). The top ranked method may even be a bit worse than labels on random items (since random items at least have more diversity). Uncertainty sampling works in our RF and TL framework so this indicates that either uncertainty sampling is robust to large fluctuations of the decision surface over iterations *or* it causes more stable selection of source tasks over feedback iterations due to more useful information from the user being fed into the system. Our current work is mainly concerned with the effective selection of source tasks for TL. However a study on the interplay of solicitation method and source task selection would make for interesting future study that could result in an even better RF and TL system.

Feedback Iteration	1	2	3	4
Max-Margin				
Uncertainty Sampling	0.11 ± 0.07	0.14 ± 0.08	0.16 ± 0.09	0.18 ± 0.09
Top Ranked	0.11 ± 0.07	0.12 ± 0.07	0.13 ± 0.07	0.13 ± 0.08
Random	0.11 ± 0.07	0.12 ± 0.08	0.13 ± 0.08	0.13 ± 0.07
Score Accuracy				
Uncertainty Sampling	0.10 ± 0.08	0.14 ± 0.08	0.17 ± 0.09	0.19 ± 0.09
Top Ranked	0.10 ± 0.08	0.12 ± 0.08	0.13 ± 0.08	0.13 ± 0.08
Random	0.10 ± 0.08	0.12 ± 0.08	0.13 ± 0.08	0.14 ± 0.08
Score Clustering				
Uncertainty Sampling	0.09 ± 0.07	0.12 ± 0.08	0.15 ± 0.09	0.17 ± 0.09
Top Ranked	0.09 ± 0.07	0.11 ± 0.08	0.12 ± 0.08	0.13 ± 0.08
Random	0.09 ± 0.07	0.10 ± 0.08	0.12 ± 0.08	0.13 ± 0.08

Table 6.1: Mean AP over Iterations for Different User Feedback Solicitation Methods for Different Source Task Selection Methods

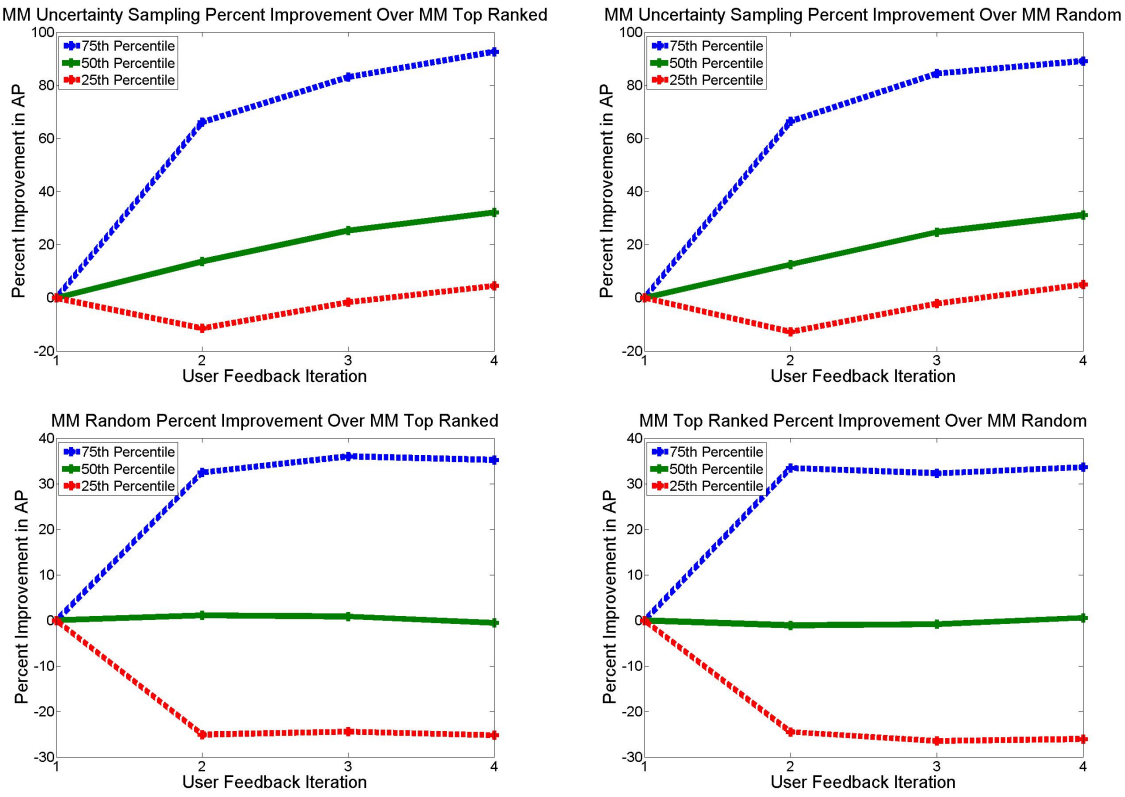

Figure 6.1: Quartile Plots of Percent Improvements for Max-Margin Source Task Selection

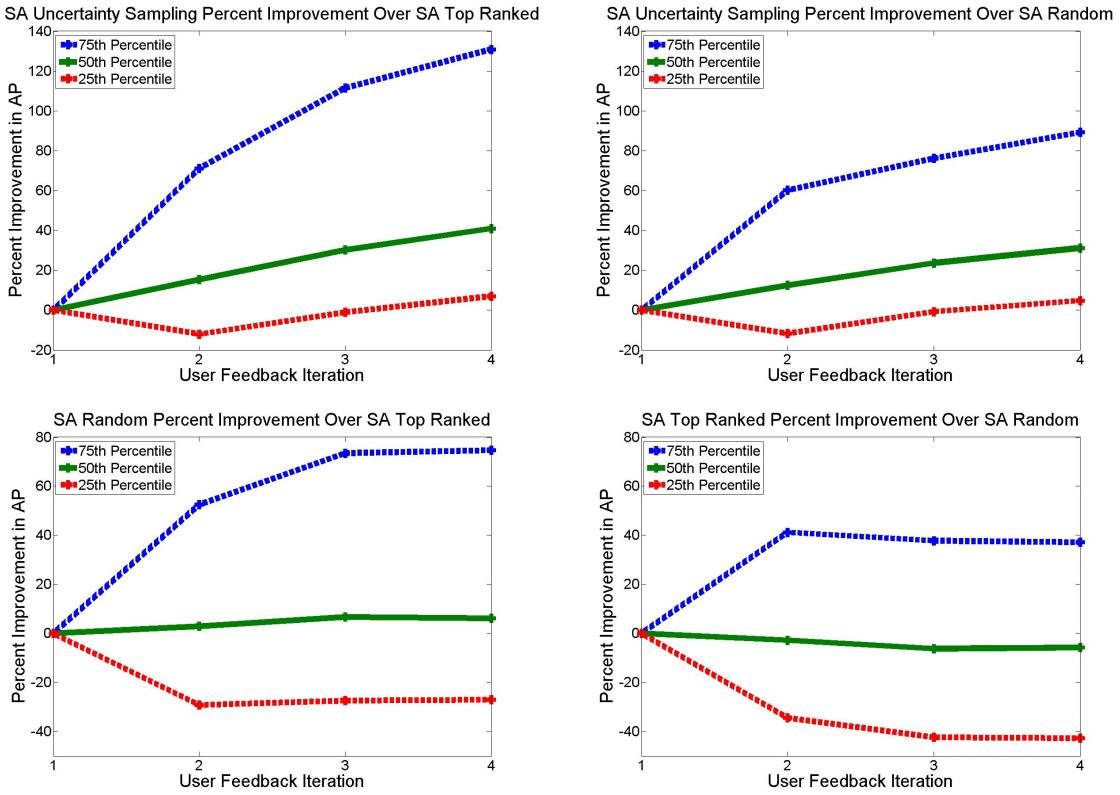

Figure 6.2: Quartile Plots of Percent Improvements for Score Accuracy Source Task Selection

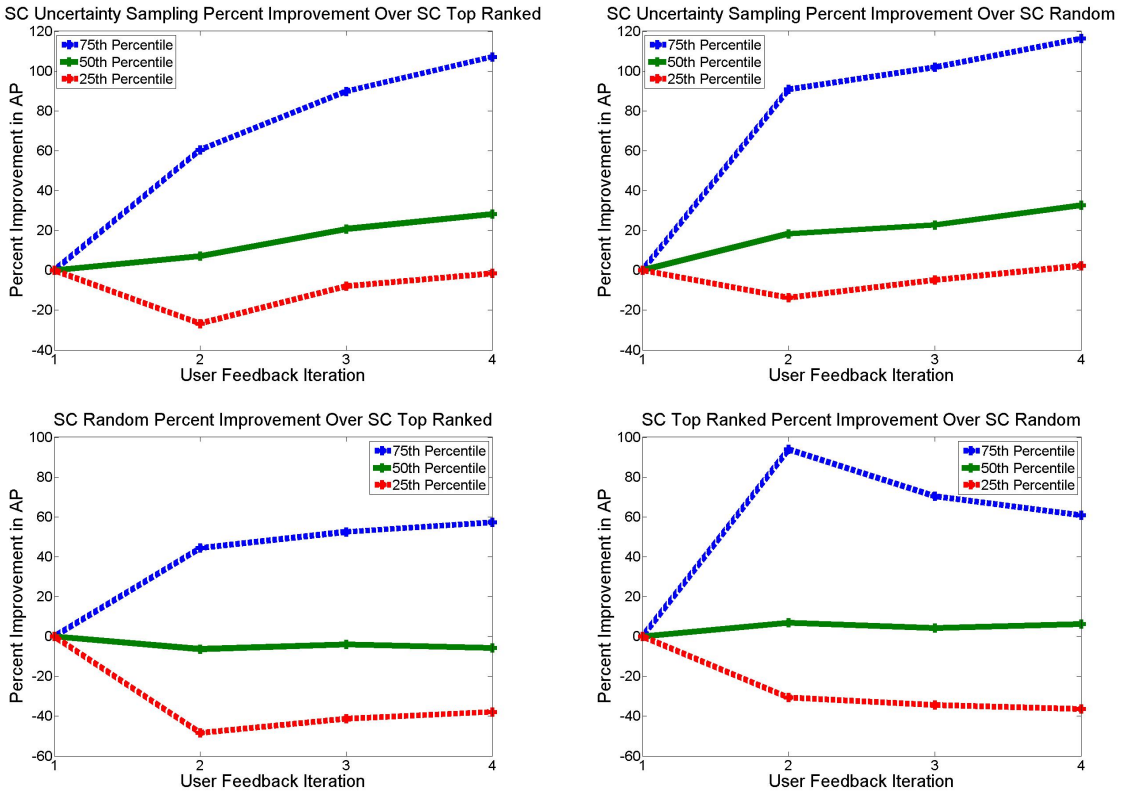

Figure 6.3: Quartile Plots of Percent Improvements for Score Clustering Source Task Selection

6.2 Source Task Selection Methods versus No Transfer Learning

In this section, we compare how effective each source task selection method is in improving RF search. To do so, we plot the percent improvements between different source task selection methods and tests where no TL was used at all. For all comparisons, we will use uncertainty sampling for feedback solicitation since the previous section established it as the best user feedback solicitation method. The performances of MM, SA, and SC can be seen in Figure 6.4. It can be seen that in the first iteration, MM is actually better than SA. However in subsequent iterations, their performances are comparable. SC is much worse than MM and SA initially and ends up slightly worse in later iterations. In all cases, percent improvement decreases as the iterations increase. This is not surprising since as more training data becomes available, the RF algorithm without TL should become increasingly accurate. Thus there would be less need for TL. This may also explain why in later iterations, all three methods had comparable performance (although SC still performed slightly worse than MM and SA). The fact that SC performed worse indicates that trying to find good clusters of the data is not a good criterion for learning target queries from users. In a relevance ranking of database items we should not expect all relevance scores to be tightly clustered. This is especially true if the target query has a lot of in class variance. Although MM outperformed SA in earlier iterations, in practice SA would be a better choice. This is because MM requires training each source task's data with the target data to evaluate which source task would be

best for TL. Thus in MM, to evaluate the quality of TL with respect to a given source task, the transfer learning with that source task has to be done. This is computationally expensive and would not be effective in a real application where we would expect many more source tasks than those used in the experiments. SA on the other hand can evaluate the quality of TL from a given source task without doing the transfer learning. As an example, in our experiments, MM would typically require 20-25 seconds of processing time for all iterations of feedback in each test. The same tests would only take 5-6 seconds for SA. (SC would take even longer than MM due to its need to run EM as well.) We present a set of sample results for the query "mixed martial arts" (MMA) showing SA based TL versus standard No TL in Figures 6.5 – 6.8. It can be seen that with TL, the ranking performances of the top 20 list is better (particularly in earlier iterations) and that overall, as feedback iterations increase, both sets of rankings improve. What is interesting to note is that without TL, much of the time was spent having the user clarify what is *not* MMA from the list of most ambiguous examples. In the TL case, we can see more positive feedback being solicited from the user which helps the system better learn the concept of MMA.

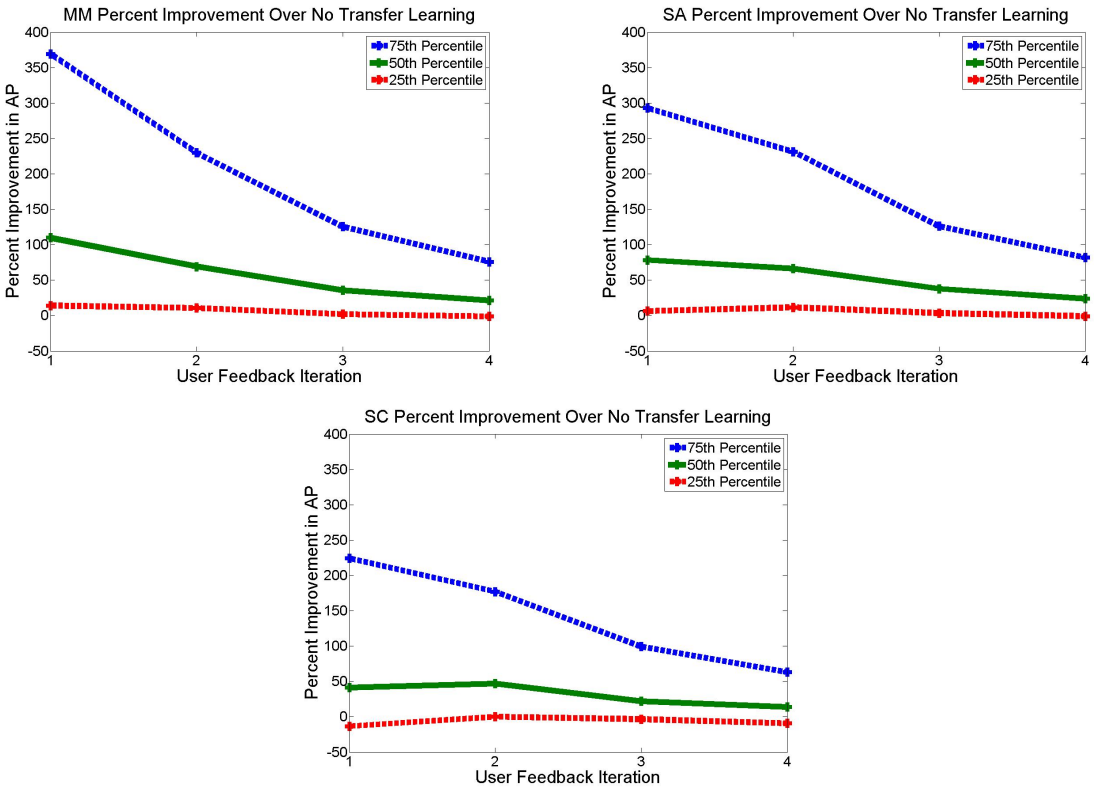

Figure 6.4: Quartile Plot of Percent Improvement for Transfer Learning Over No Transfer Learning

6.3 Conclusion

We have shown the strength of our RF and TL approach in addressing the limited user feedback issue (a main criticism of RF). Each source task selection method showed significant improvements in AP over not using TL. (Although only SA is efficient enough for real applications.) As expected, TL results showed uncertainty sampling to be best. Interestingly the random method outperformed the top ranked method in our SA experiments. Trends in our other experiments indicate the random method should perform worse than the top ranked method. A study of how TL affects active learning is out of the scope of the current work (we focus on the selection of source tasks) but investigating this would be good for future work.

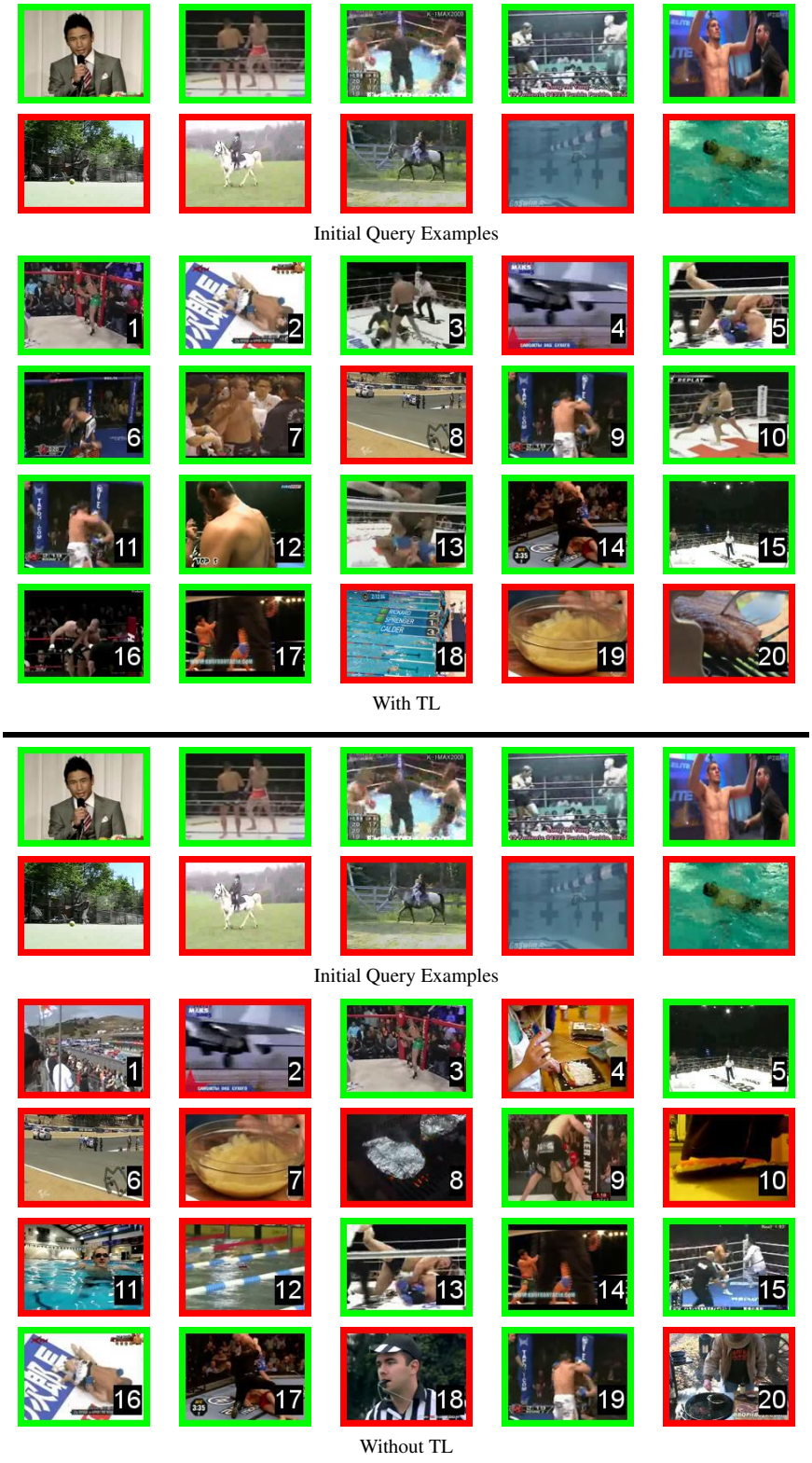

Figure 6.5: Sample ranking for feedback iteration 1 of the query “Mixed Martial Arts.” Correct results are indicated in green while incorrect results are indicated in red.

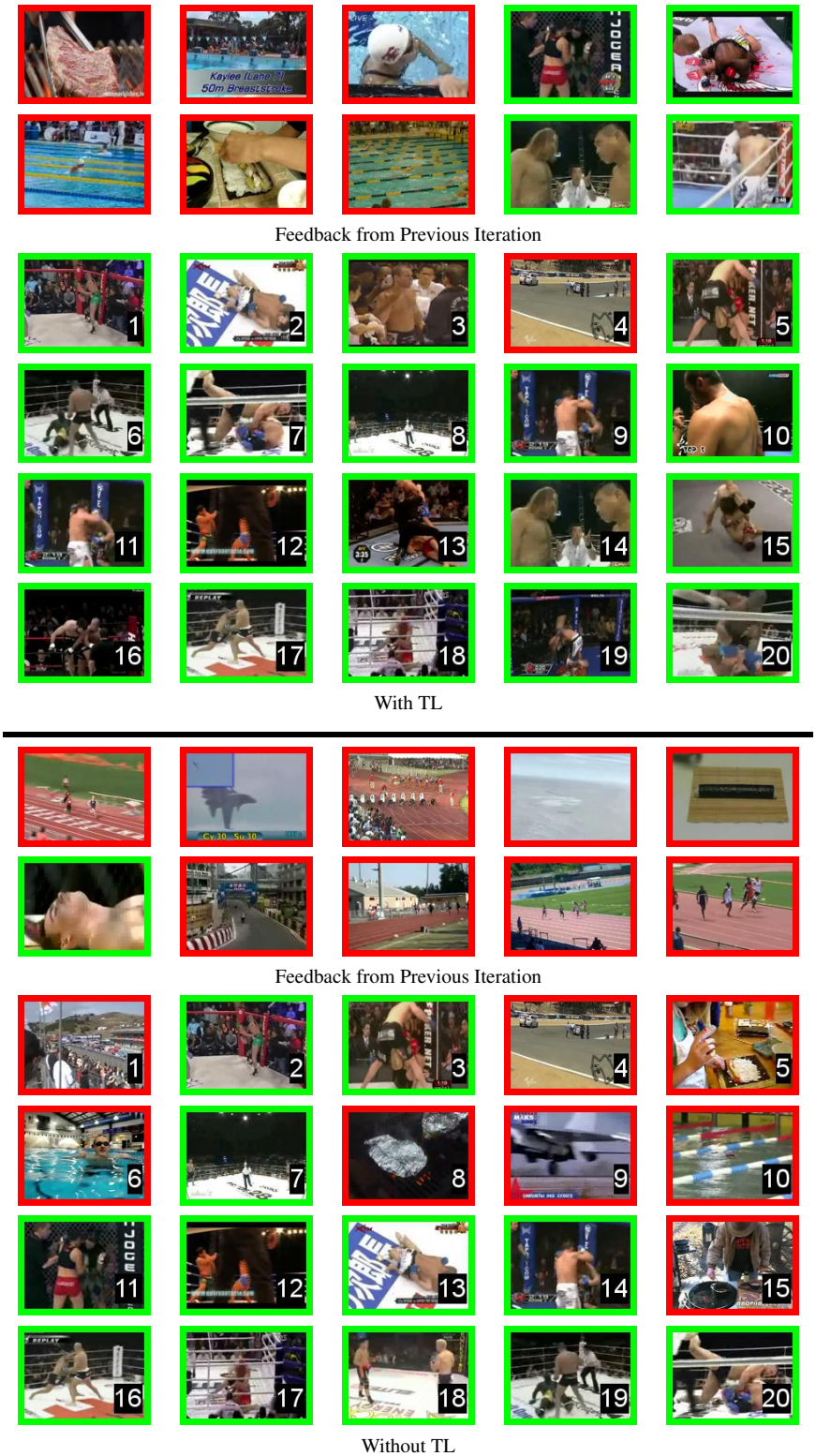

Figure 6.6: Sample ranking for feedback iteration 2 of the query “Mixed Martial Arts.” Correct results are indicated in green while incorrect results are indicated in red.

Figure 6.7: Sample ranking for feedback iteration 3 of the query “Mixed Martial Arts.” Correct results are indicated in green while incorrect results are indicated in red.

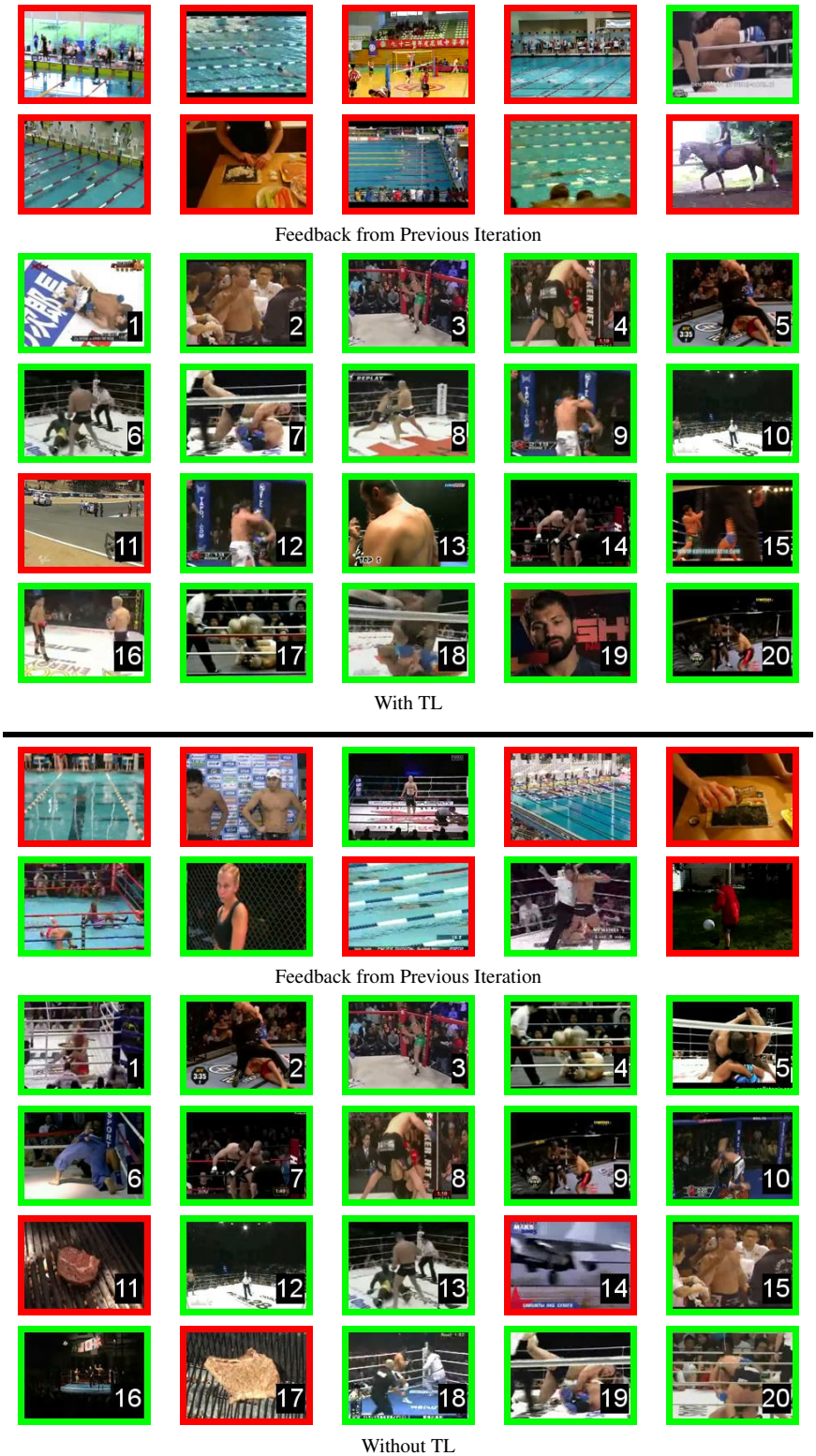

Figure 6.8: Sample ranking for feedback iteration 4 of the query “Mixed Martial Arts.” Correct results are indicated in green while incorrect results are indicated in red.

Chapter 7

Face Recognition

In the search of Internet images and videos, humans are primary subjects of interest. Thus it is sensible to construct specific algorithms for the purpose of analyzing humans. (As opposed to a general learning method for analyzing a wide variety of topics.) The number of problems in human analysis is vast and so covering them all is out of the scope of this dissertation. However, we will cover one problem in human analysis as an example of a specifically engineered ranking function. The problem is in the robust identification of humans via face recognition. In particular, we focus on recognition between out-of-plane rotated faces. This chapter describes a face alignment method and a ranking function for comparing how similar two novel faces are even when their poses are different. [1]

[1] The work and figures shown here are our own work and are reprinted with permission from Lam and Shelton [2008] © IEEE.

7.1 Related Work

Automatic face recognition has numerous applications in areas as diverse as security, human computer interaction, and of course image search engines. As such, there has been much work on face recognition in the past decades and tremendous progress has been made. There already exist systems that can perform in excess of 90% accuracy under controlled conditions [Gross et al., 2001]. However, changes in illumination or pose remain largely unsolved problems [Zhao et al., 2003]. In this paper we focus on the issue of recognition across pose. The general problem we wish to address is as follows: given two images of faces in arbitrary poses, indicate how likely they are to be the same person. This is similar to the "one sample per person" problem mentioned by Tan et al. [2006]. This is an important problem because it is not always possible to have multiple images of the same person. This is especially true if one were attempting to determine the identity of a stranger. In this problem, the stranger may have pictures of himself scattered all over the Internet with no clear organization of those images but there still must be a way to query a search engine with just one facial image and get ranked results of other similar faces.

In recent years, there has been much work related to the problem of face recognition with pose changes. For example, Blanz and Vetter [2003] built a system that uses a 3D morphable model to perform face recognition. In their work they built explicit 3D models of the head and face which has the advantage that their models can be very accurate. However, there have been other works using simpler models that have proven effective [Du and Ward,

2006, Gonzlez-Jimnez and Alba-Castro, 2007]. (Although it should be noted that Gonzlez-Jimnez and Alba-Castro [2007] did not perform as well on full profile views. Our system performs competitively on such views.) Eigen light-fields [Gross et al., 2004] addresses the problem by computing an eigenspace from the light-fields of the head and recognizing based on eigen light-fields in a manner analogous to Eigenfaces [Turk and Pentland, 1991]. It has the advantage of being able to use as many images as are available to improve its accuracy and does not require the gallery images to be in a canonical pose. However, it may be fruitful to investigate how a more explicit image-based representation of relations between pose can improve accuracy. The Eigen Light-Fields approach also does not make use of a component based decomposition of the face which has been shown in some cases to be more beneficial than a global approach to recognition [Heisele et al., 2003].

There have also been other works that attempt to solve this problem using more explicit learning of pose relations through patch decompositions. Kanade and Yamada [2003] presented a multi-subregion based approach which decomposes faces across pose into patches and learns the relations between corresponding patches from one pose to another under a Gaussian model. Their work showed that recognition between poses separated by as much as 45 degrees can still be done with accuracy in excess of 80%. However, a drawback to the multi-subregion system is its reliance on similarity in appearance between corresponding patches of the same people. Not surprisingly, at extreme differences in pose, accuracy drops. Liu and Chen [2005] extended the work of Kanade and Yamada [2003] by introducing a texture map representation of the face. They assume the head to be an ellipsoid and deter-

mine what the texture map of such an ellipsoid head would be. The basic idea is that their transformation allows for facial features to maintain greater similarity over a wider range of pose changes. While this does improve results, their alignment procedure has to optimize over a total of eight parameters, for each image. There is also the drawback that at extreme poses, there is a limit to how much texture mapped ellipsoids can faithfully transform facial features.

In this chapter, we offer our extension to the work of Kanade and Yamada by modeling the relations between patches not based on their similarity but on their joint appearances. This is achieved through the application of Support Vector Machines (SVMs) [Burges, 1998] for capturing patch relations between poses. We first present an overviews of Kanade and Yamada's algorithm and the related work by Liu and Chen. We the present our algorithm and experimental results on manually aligned faces and preliminary results on recognition performance with automatically cropped and aligned faces.

7.1.1 Kanade and Yamada's Multi-Subregions

Kanade and Yamada developed a system for recognition across pose based on the similarity of local regions on the face between different poses of the same people and different people. They first manually determined the locations of the eyes and mouth for all facial images and used those locations to define a 7-by-3 lattice of points on the face starting from the eyebrow and extending down to the chin. These points were then used to define 9-by-15 pixel subregions on the face and the similarities between corresponding regions between

poses were then modeled using two Gaussians, one for the similarities of image patches between same identities and the other for different identities. In our implementation of their system, we needed to make one modification because they assume all regions are 9-by-15 pixels in size. Our regions are variably sized so we had to normalize the sizes between corresponding regions. We do this normalization by resizing the smaller patch to be the same size as the larger patch. In a way, this illustrates a strength of our approach. Our system can learn relations between subregions of different poses without the need to fix patch sizes to be the same between all poses.

7.1.2 Liu and Chen's Texture Maps

Liu and Chen, developed a transformation in which the head is assumed to be an ellipsoid and a texture map based on this assumption is computed for each face. The idea is that if the head were an ellipsoid, the out of plane rotation of the head would be easier to recognize after this transformation. The hope is that the texture maps would help to preserve similarities of facial features between the same person across a wider range of poses than in Kanade and Yamada's work. Liu and Chen first manually cropped out faces. They then applied texture map transformations to the faces based on a best fit to their Universal Mosaic Model. As we did not have such a model available to us, we used our manually aligned faces and determined (manually) the texture mapping parameters for fitting to a canonical mosaic model. We spent a long time optimizing these parameters to achieve good results and visually inspected the texture map results to verify that they were reasonable fits.

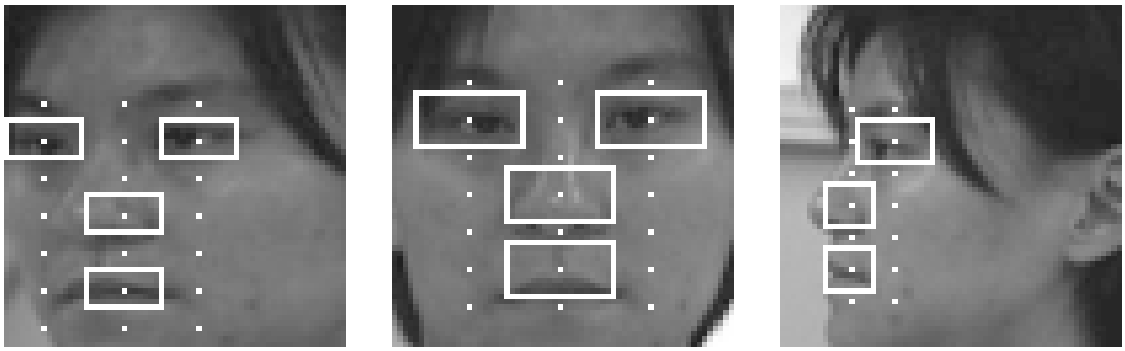

Figure 7.1: Centers of the Salient Regions and some of the Bounding Boxes

7.2 Our SVM Based Face Recognition

SVMs have been shown to be a powerful tool for face recognition [Jonsson et al., 2000, Heisele et al., 2003]. Jonsson et al. [2000] showed empirical evidence that SVMs could effectively extract relevant discriminatory information from training images of frontal images. Heisele et al. [2003] even applied SVMs to recognition in the presence of pose change. However, they only tested their system on a database of 10 subjects as their goal was to establish the strength of SVMs for face recognition and to compare the accuracies of global versus local approaches to analyzing faces. In our work, we continue this line of investigation by testing SVMs trained on local image patches of the face using the CMU PIE database because it has more individuals (68 subjects) and a greater set of ranges in face pose. As Kanade and Yamada [2003] presented promising initial work on this problem, we will adopt their testing methodology so our work can be compared.

7.2.1 Local Patch Representation of Faces

In our work, we first chose to manually decompose frontal faces into 21 salient rectangular regions. We then manually located the corresponding salient regions to the frontal regions for all other poses. In the case where only a part of the face was visible due to self occlusion in more extreme poses, we only located the visible corresponding regions (see Figure 7.1). This is essentially the same as the decomposition performed by Kanade and Yamada except we did not define all regions to be of the same size. This is because corresponding regions across pose do not maintain the same size due to foreshortening and self-occlusion as the pose change deviates from one view to the next. A good example of this effect is in the eye. If we imagine a head turning from a frontal view towards the right, the bounding box around the person's right eye would appear increasingly smaller.

Note that this step is performed once per pose, not once for each image of a pose. We assume that the images are aligned, either manually or automatically (see section 7.2.4). Thus, this step represents the input of knowledge about the position (although not appearance) of facial features under rotation. Currently this is manual, but we can imagine future systems automatically finding the salient regions of all poses.

7.2.2 Discretely Separated Poses for Training Face Recognition

Kanade and Yamada showed that recognition accuracies beyond 90% between images of faces in poses differing by as much as 30 degrees of out of plane rotation can be done by just measuring the sum squared error between corresponding patches (as part of a Gaussian

model). This suggests that computers can tolerate some degree of pose change without very precise modeling of facial geometry. Intuitively, this makes sense because minor changes in facial pose do not change facial appearance dramatically. As long as we know the general locations of corresponding salient regions in the face, direct comparisons between the pixel values of the regions suffices when pose change is minor. Recognition accuracy mainly drops when the poses of two given facial images are too different. This suggests that there may be no need to model pose change as a continuous process for face recognition purposes. Instead, modeling how a discrete set of specific poses (provided they cover a wide range of rotations) relate to each other may be all that is needed for recognition across pose.

Thus, we only define the 21 regions above for a set of discrete poses. We feel this is enough to gain good accuracy on any intermediate poses. For this work, we chose the 13 poses from the CMU PIE Database (see Figure 7.3).

7.2.3 SVMs for Learning Pose Relations

We now define the main contribution of our this chapter: a pose relation SVM approach to face recognition. The basic idea behind our application of SVMs to learning pose relations is to train SVMs to answer the question of whether two faces in pose m and pose n are of the same person or not. As our analysis of faces is based on the patch decomposition of Kanade and Yamada, learning a relation from two given poses m and n requires training an SVM for each of the corresponding regions between the two poses. These SVMs then allow us to decide if the corresponding regions belong to the same person or not.

To apply SVMs to the problem of learning pose relations, our algorithm first takes in images of multiple known subjects in different poses as training data. These people are not part of the set to be recognized (nor the query set). This set is only used to learn a general relationship between the appearance of a region in one pose and the appearance of the same region in a different pose. The relations between every pair of poses, m and n, are independently learned.

The training procedure will be presented shortly but we shall first introduce some notation. Let $p_m(i)$ be an image of subject i in pose m and $p_m^k(i)$ be the k^{th} region of the image $p_m(i)$. Let $v_m^k(i)$ be region $p_m^k(i)$ represented as a vector and $v_{m,n}^k(i,j)$ be the concatenation of $v_m^k(i)$ and $v_n^k(j)$. The training procedure between poses is as follows.

1. For each k, m, n, i, j if $i = j$, consider $v_{m,n}^k(i,j)$ to be a positive case for region k; otherwise, consider it to be a negative case for region k.

2. For each region k, train an SVM $R_{m,n}^k$ (with a radial basis function kernel) using the corresponding dataset.

This procedure aims to build independent pose relation SVMs for each of the corresponding 21 regions between two given poses. The learned functions are used to determine a score for how likely two novel images of faces are to be the same person. To employ the learned functions, we take two novel facial images q and t in poses a and b respectively and subdivide each of them into the regions associated with their pose. For each pair of the corresponding regions l in the two images, we concatenate the vectors describing each of the regions l into a

single vector $v^l_{a,b}(q,t)$ and feed it to the function $R^l_{a,b}$ which returns whether the two regions match.

It would be natural to sum the resulting number of positive classifications (across the region l) from the SVMs to determine how likely it is that the two images are of the same person. However, we discovered that due to the small training data sizes and the relatively large portion of negative examples in the training sets, all of the R outputs would be -1 (*i.e.* not a match) for all our test data.

Instead, we discovered that the raw distances to the hyperplane for each SVM provided indications of which subjects were likely to be the same person. Thus if we let $R^l_{a,b}$ be not the thresholded output of the support vector machine, but rather the distance to the hyperplane (*i.e.* the value prior to thresholding), although all of the outputs would be negative, the total sum would still be a good measure. Thus we use

$$s_r(q,t) = \sum_{k=1}^{K} R^k_{a,b}(v^k_{a,b}(q,t)) \tag{7.1}$$

to score whether two images q and t with poses a and b respectively are of the same person. The higher the score, the more likely it is that the two observed faces belong to the same person.

Figure 7.2: Examples of the Alignment Grids

7.2.4 Automated Face Cropping and Alignment

We performed experiments using manually cropped and aligned images of the faces but we also developed an automation of that procedure. To automate the process, we first used the Viola-Jones face detector Viola and Jones [2001] to automatically crop faces from images. We found the detector to be robust enough to crop faces in a wide range of poses (including profile views) but the faces would typically have some variation in alignment. This is partially due to the detector not always cropping faces consistently and also from differences in the way the subjects positioned their heads in the images. As such, an alignment procedure was found to be necessary to position the faces in canonical positions so that their salient regions could be better compared.

Our simple SVM based alignment procedure aligns image crops reasonably well for a large number of poses. The procedure learns canonical alignments for a set of discretely separated poses based on provided manual alignments as training data. The alignment algorithm uses a set of "alignment SVMs" for each pose p_n which are trained as follows.

1. Gather a training set of manually aligned faces in pose n.

2. For each image, divide the entire image into K evenly sized regions $r_n^1, r_n^2, ..., r_n^K$ over the entire image.

3. For each region r_n^k, train an SVM A_n^k (using a radial basis function kernel) to classify all examples of region r_n^k as positive and any other region r_n^l where $l \neq k$ as negative.

In our experiments, we set $K = 25$ (see Figure 7.2). Once the training procedure is complete, the 25 SVMs can be used to score alignments of facial images $p_n(i)$ through observation of its 25 evenly sized regions. The scoring function is defined as

$$s_a(p) = \sum_{k=1}^{K} A_n^k(r_n^k(p)) \tag{7.2}$$

where $r_n^k(p)$ is the region k in image p (assuming pose n). We note that the SVMs here are separate from the SVMs for recognition. Each pose n has a set of alignment SVMs specifically trained for it. These determine what type of appearance each local region r_n^k should have and the SVMs vote using their raw distances to the hyperplane on whether their own region is consistent with a good alignment.

Using the alignment scoring procedure, a search for a good alignment can be done simply by a brute force search over the alignment parameters of translation, scale, and in-plane rotation. However, a brute force search is very slow so we adopted a heuristic. We observed that facial crops found by the Viola-Jones detector are restricted to a certain range of differences in translation, scale, and rotation. We decided to perform a coordinate ascent by iteratively

optimizing over each parameter independently and observed good results. However, the automatic face crops were less consistent in profile views and those images did not automatically align as well. To address the problem with the profile views, we introduced random restarts into our search procedure and applied the same procedure (with random restarts) to all the poses (including frontal views). We outline the specific details of our alignment procedure below.

1. Maximize Equation 7.2 over each alignment parameter (scale, rotation, x-translation, and y-translation) in turn, while keeping the others fixed. (All parameter searches on the variables are within some bounded region from the variable's current value.)

2. Repeat the above step until convergence or a maximum number of iterations has been reached.

3. If the score of the found alignment exceeds a minimum alignment score threshold, accept the found alignment.

4. Otherwise, perform a random restart to some other point in the parameter space within a bounded region from the alignment that was just found and repeat from step 1. (We do a maximum of 10 random restarts.)

5. If no alignment with a score exceeding the minimum score threshold was found, select the alignment with the highest score.

In our outline, we introduced a minimum alignment score threshold. This threshold is

determined based on the training data used for the particular pose being aligned. We set the threshold to be the mean of the training scores minus half of the standard deviation of the same scores. (Higher scores indicate better alignment. This threshold serves to trade-off accuracy for speed.)

While there exist 3D facial alignment methods such as the work of Gu and Kanade [2006], these methods rely on 3D laser scans to use as training data. Our alignment algorithm only requires a set of 2D images. (Although we should note that the work here on alignment is still somewhat preliminary.) While this chapter's emphasis is mainly on exploring the performance of SVMs in face recognition across pose, we include this alignment procedure and corresponding results to show more general use of region-based SVMs in face recognition.

7.3 Experiments

We tested our system on the CMU PIE database and adopted the general protocol used by Kanade and Yamada. The protocol is to choose half of the subjects for training the recognition system and the other half for testing. However, when researchers develop such systems and are using the same test data to verify their algorithms (during development), they may unknowingly overfit their test data. This is due to the fact that algorithms under development can be adjusted and modified in many ways. We first developed our system using the first half of the subjects as the training set and the last half as the testing set (in which the frontal pose was used as the gallery database and the non-frontal poses used as queries) and only coarsely

Figure 7.3: Examples of the Poses in the CMU PIE DB (From the CMU PIE DB Website)

tuned the parameters of our algorithm. After we were satisfied with our system, we selected five random splits of the subjects. Each split would have half the subjects randomly selected for training and the other half for testing. We then tested our system without adjusting any parameters on these five random splits and determined the mean and standard deviations of our accuracies for recognition. The same splits were used to test our implementations of the systems described in Kanade and Yamada [2003] and Liu and Chen [2005] for comparison purposes.

7.3.1 The CMU PIE Database

The CMU PIE Database [Sim et al., 2003] contains images of 68 subjects taken under 13 different poses, 21 different illuminations, and 2 occasions resulting in over 37,000 images of people. In our experiments, we use only the frontal illumination, neutral expression, no glasses subset of the database. This means we work with the 68 subjects where each one has 13 poses. The poses are denoted by their camera labels (*e.g.* c27 for the frontal view and c11 for one of the 45 degree views).

7.3.2 Results

In our experiments, we tested each split independently with all facial crop sizes normalized to 64-by-64 pixels. We kept galleries of frontal images from each of the five splits and used all the other poses as probe images. For SVM training, we used Gaussian kernels with $C = 1$ in all cases. At the moment, we assume that the pose of each image is specified so the recognition accuracies determined were done independently for each pose for all three systems. There exist systems in the literature [Osadchy et al., 2007] where face detection and pose estimation are done simultaneously and such systems could be integrated into ours.

Figure 7.4 compares our system to the other two systems with manually aligned images. It can be seen that our method outperforms the other two methods as the pose becomes increasingly distant from the frontal view. The reason we do not have as much advantage over the texture map method in the less extreme poses is because the appearance of subregions does not vary greatly when pose change is minor. In this case, the other systems may actually be more generally applicable than our SVM-based system since they are based on direct measurements of similarity between image patches. (The multi-subregion method would have been expected to perform well in the closer to frontal cases but did not. This is likely due to the differently sized subregions that we selected.) SVMs on the other hand require sufficient training data to choose support vectors that would cover a wide enough range of cases in facial appearance.

However in the case of extreme pose change, the use of similarity between images patches fails because the same facial feature can appear dramatically different (*e.g.* the nose). In this

case, use of SVMs provides better accuracy. It should be noted that although we made a best effort to produce a fair comparison of our work to the work of Liu and Chen, many factors such as the way images were cropped or our choice of texture parameters can have an effect on their accuracy. We note that Liu and Chen [2005] reported accuracies of 60% and 70% for their most extreme poses so it may be that our implementation of their system is not optimally tuned. However, they use more facial features. For example, they note the forehead as being a strong indicator of identity between poses. We chose to only limit ourselves to the regions defined in Kanade and Yamada which focuses primarily on the face. It can be seen that even with fewer features and a more rigorous 5-split testing procedure we still achieve about 70% accuracy for both extreme poses.

Figure 7.5 compares our automatic alignment system's performance to that of manually aligned images and automatic crops without automatic alignment. For these results, we also retrained the alignment system based on each of the random splits. It is not surprising that there is a degrading of performance. The performance is especially degraded in the extreme poses. Although our automatic alignments were actually quite close to the manual alignments, accuracy was likely affected by slight inconsistencies in scale and translation. Our current use of SVMs examines the pixel values directly and is thus more sensitive to misalignments. However, it is encouraging that the rank 2 accuracies of the automatically aligned images can be 15% greater than the rank 1 accuracies. If a user were using such a face recognition system in a search engine, a set of top ranked images would still provide useful results to the user.

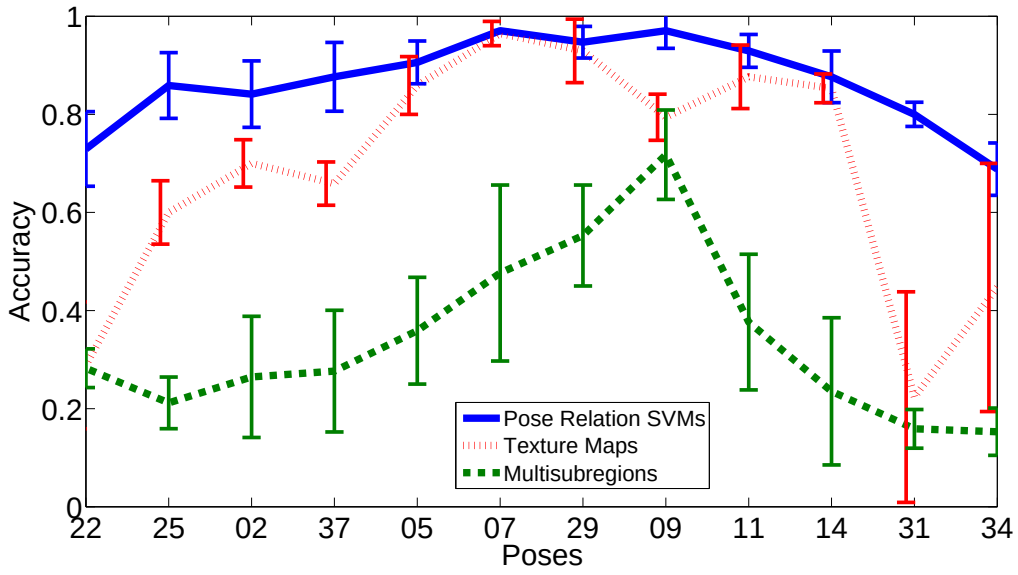

Figure 7.4: Rank 1 Recognition Results on Manually Aligned Faces (means with standard deviations) Note that poses 05, 07, 29, and 09 are closest to frontal while poses 22 and 34 are profile views.

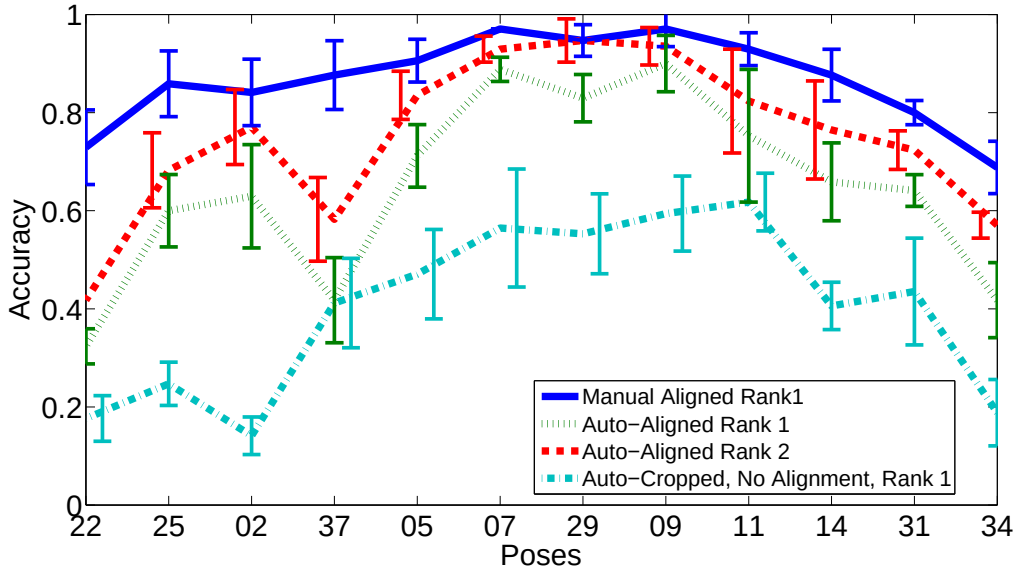

Figure 7.5: Recognition Results on Automatically Cropped and Aligned Faces (means with standard deviations) Note that poses 05, 07, 29, and 09 are closest to frontal while poses 22 and 34 are profile views.

7.4 Conclusions

This chapter presented a method for using region-based pose relation SVMs to learn aligning and recognition scoring functions. The results are good and do not require any manual intervention except the definition of 21 regions for each pose. We are especially encouraged by the quality of the results given the small training set size (only 34 images per pose). In most applications, a larger database of images would be available. In future work, our robust face recognition method (as well as other human analysis methods) will be integrated into a complete system for the search of Internet images and videos.

Chapter 8

Conclusion

In this dissertation, we explored different ways of building ranking functions for computer vision based search. In particular, we focused on the search of Internet video which posed a number of challenges. The main challenges in this application were in the question of how to accommodate the vast types of things users could query and how to learn classifications for "real-world" videos. To address the first problem, we proposed the use of relevance feedback (RF), a type of interactive information retrieval framework that could learn the subjectivity in user queries. However a major weakness of RF is that users are not expected to give much feedback to the system. This limits the utility of RF search in complex scenarios such as the retrieval of Internet video.

To alleviate this training data scarcity problem, we proposed the use of transfer learning (TL) in combination with RF. In our experiments, we found our proposed framework to be very capable in improving learning from just a few examples. Despite the improvement

in ranking performance for our retrieval system, there are still many questions that can be addressed. The current work is basically a combination of TL and active learning (AL) for good RF search. However the question of how TL and AL interact with each other is left open. In our results, we confirmed that uncertainty sampling was most effective for soliciting user feedback. When used in our RF and TL framework, polling the user on the relevance of items nearest the current SVM decision surface results in labels on what is most ambiguous with respect to the current decision surface. However, the new decision surface used to re-rank the database is not just based on the new user feedback as in the no TL case. Instead, we use all available relevance labels to decide which source task should be used to bias the decision surface as well. This could result in decision surfaces that (over user feedback iterations) "jump" all over the feature space much more than would occur normally without TL. How this much "jumping" occurs and how this affects performance are still not clear. We speculate that there should be a better AL based feedback solicitation strategy which takes into account the use of TL and source task selection algorithms. With such a feedback strategy, it should be possible ask the user for relevance labels that would better help the source task selection algorithm find appropriate tasks to transfer from. In addition, such a feedback strategy should take into account the need to diversify relevance labels obtained from the user so that effective exploration of the solution space can be done.

In this dissertation, we have explored different ranking functions for web search. In particular, we focused on improving the learning component of an interactive search framework. We showed strong performance on a challenging real-life collection of nearly 3,600 YouTube

videos with 127.5 hours of footage in total and have suggested interesting directions for future research. In addition to providing a powerful framework for search, we believe widespread use of our framework could also lead to more human labeling of the vast amount of data on the Internet. Such historic labels could also be used as source data in any computer vision technologies that would utilize TL. Thus the impact of our proposed framework could extend well beyond retrieval applications.

Bibliography

Donald A. Adjeroh, M. C. Lee, and Irwin King. A distance measure for video sequences. *Computer Vision and Image Understanding*, 75(1-2):25–45, 1999.

Jürgen Assfalg, Alberto Del Bimbo, and Pietro Pala. Three-dimensional interfaces for querying by example in content-based image retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 8(4):305–318, 2002.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Stphane Canu, Yves Grandvalet, Vincent Guigue, and Alain Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France, 2005.

Liangliang Cao, Zicheng Liu, and Thomas S. Huang. Cross dataset action detection. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. IEEE Computer Society, 2010.

Abdolah Chalechale, Golshah Naghdy, and Alfred Mertins. Sketch-based image matching using angular partitioning. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(1): 28–41, January 2005.

Liang-Hua Chen, Kuo-Hao Chin, and Hong-Yuan Liao. An integrated approach to video retrieval. In *Proceedings of the Conference on Australasian Database*, pages 49–55. Australian Computer Society, Inc., 2008.

Yunqiang Chen, Xiang Zhou, and Thomas S. Huang. One-class svm for learning in image retrieval. In *Proceedings of the IEEE International Conference on Image Processing*, pages 34–37. IEEE Computer Society, 2002.

Michel Crucianu, Marin Ferecatu, and Nozha Boujemaa. Relevance feedback for image retrieval: A short survey. State of the art in audiovisual content-based retrieval, information universal access and interaction including data models and languages, DELOS2 Report (FP6 NoE), 2004.

Wenyuan Dai, Qiang Yang, Gui R. Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the International Conference on Machine Learning*, pages 193–200. ACM, 2007.

Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1): 1–38, 1977.

Shan Du and Rabab Ward. Face recognition under pose variations. *Journal of the Franklin Institute*, 343(6):596–613, 2006.

Lixin Duan, Dong Xu, Ivor W. Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. IEEE Computer Society, 2010.

Graham D. Finalyson. Color in perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1034–1038, 1996.

Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):23–32, 1995.

P. Geetha and Vasumathi Narayanan. A survey of content-based video retrieval. *Journal of Computer Science*, 4(6):474–486, 2008.

Bo Geng, Linjun Yang, Chao Xu, and Xian-Sheng Hua. Ranking model adaptation for domain-specific search. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 197–206. ACM, 2009.

King-Shy Goh, Edward Y. Chang, and Wei-Cheng Lai. Multimodal concept-dependent active learning for image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 564–571. ACM, 2004.

Daniel Gonzlez-Jimnez and Jos Luis Alba-Castro. Toward pose-invariant 2-d face recognition through point distribution models and facial symmetry. *IEEE Transactions on Information Forensics and Security*, 2:413–429, 2007.

Ralph Gross, Jianbo Shi, and Jeffrey Cohn. Quo vadis face recognition? Technical Report CMU-RI-TR-01-17, Robotics Institute, Carnegie Mellon University, 2001.

Ralph Gross, Iain Matthews, and Simon Baker. Appearance-based face recognition and light-fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):449–465, 2004.

Lie Gu and Takeo Kanade. 3D alignment of face in a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1305–1312. IEEE Computer Society, 2006.

Alexander G. Hauptmann, Wei H. Lin, Rong Yan, Jun Yang, and Ming Y. Chen. Extreme video retrieval: Joint maximization of human and computer performance. In *Proceedings of the ACM International Conference on Multimedia*, pages 385–394. ACM, 2006.

Jingrui He, Hanghang Tong, Mingjing Li, Hong-Jiang Zhang, and Changshui Zhang. Mean version space: A new active learning method for content-based image retrieval. In *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 15–22. ACM, 2004.

Bernd Heisele, Purdy Ho, Jane Wu, and Tomaso Poggio. Face recognition: Component-based versus global approaches. *Computer Vision and Image Understanding*, 91(1-2): 6–21, 2003.

Yuxiao Hu, Liangliang Cao, Fengjun Lv, Shuicheng Yan, Yihong Gong, and Thomas S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 2009.

Rong Jin and Er G. Hauptmann. Using a probabilistic source model for comparing images. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE Computer Society, 2002.

Kenneth Jonsson, Josef Kittler, Yongping Li, and Jiri Matas. Learning support vectors for face verification and recognition. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 208–213. IEEE Computer Society, 2000.

Takeo Kanade and Akihiko Yamada. Multi-subregion based probabilistic approach toward pose-invariant face recognition. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 2, pages 954–959. IEEE, 2003.

Kris M. Kitani, Yoichi Sato, and Akihiro Sugimoto. Recognizing overlapped human activities from a sequence of primitive actions via deleted interpolation. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(7):1343–1362, 2008a.

Kris M. Kitani, Yoichi Sato, and Akihiro Sugimoto. Recovering the basic structure of human activities from noisy video-based symbol strings. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(8):1621–1646, 2008b.

Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64 (2-3):107–123, 2005.

Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2008.

Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2009a.

Rujie Liu, Yuehong Wang, Takayuki Baba, Daiki Masumoto, and Shigemi Nagata. Svm-based active feedback in image retrieval using clustering and unlabeled data. *Pattern Recognition*, 41(8):2645–2655, 2008.

Xiaoming Liu and Tsuhan Chen. Pose-robust face recognition using geometry assisted probabilistic modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 502–509. IEEE Computer Society, 2005.

Xiaoming Liu, Yueting Zhuang, and Yunhe Pan. A new approach to retrieve video by example video clip. In *Proceedings of the ACM International Conference on Multimedia*, pages 41–44. ACM, 1999.

Yiming Liu, Dong Xu, Ivor W. Tsang, and Jiebo Luo. Using large-scale web data to facilitate textual query based retrieval of consumer photos. In *Proceedings of the ACM International Conference on Multimedia*, pages 55–64. ACM, 2009b.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Huanbo Luan, Yantao Zheng, Shi-Yong Neo, Yongdong Zhang, Shouxun Lin, and Tat-Seng Chua. Adaptive multiple feedback strategies for interactive video search. In *Proceedings of the International Conference on Content-based Image and Video Retrieval*, pages 457–464. ACM, 2008.

B. S. Manjunath and Wei Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, May 2001.

Margarita Osadchy, Yann Le Cun, and Matthew L. Miller. Synergistic face detection and pose estimation with energy-based models. *Joural of Machine Learning Reserach*, 8: 1197–1215, 2007.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

Ariadna Quattoni, Michael Collins, and Trevor Darrell. Transfer learning for image classification with sparse prototype representations. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. IEEE Computer Society, 2008.

Ariadna Quattoni, Xavier Carreras, Michael Collins, and Trevor Darrell. An efficient projection for l1, ∞ regularization. In *Proceedings of the International Conference on Machine Learning*, pages 857–864. ACM, 2009.

Wei Ren, Sameer Singh, Maneesha Singh, and Yuesheng Zhu. State-of-the-art on spatio-temporal information-based video retrieval. *Pattern Recognition*, 42(2):267–282, 2008.

J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. Prentice-Hall, Inc., 1971.

Yong Rui, Thomas S. Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, Sep 1998.

Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 18(2):95–145, 2003.

Michael S. Ryoo and Jake K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 2009.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision*. Springer, 2010.

Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–535, 1997.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2010.

Arjan T. Setz and Cees G. M. Snoek. Can social tagged images aid concept-based video search? In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1460–1463. IEEE, 2009.

Terence Sim, Simon Baker, and Maan Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and TRECVid. In *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, pages 321–330. ACM, 2006.

Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

Robert Spence. Rapid, serial and visual: A presentation technique with potential. *Information Visualization*, 1(1):13–19, 2002.

Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.

Xiaoyang Tan, Songcan Chen, Zhi-Hua Zhou, and Fuyan Zhang. Face recognition from a single image per person: A survey. *Pattern Recognition*, 39(9):1725–1745, 2006.

Kinh Tieu and Paul Viola. Boosting image retrieval. *International Jouranal of Computer Vision*, 56(1-2):17–36, 2004.

Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 107–118. ACM, 2001.

Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

Nuno Vasconcelos and Andrew Lippman. A probabilistic architecture for content-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 216–221. IEEE Computer Society, 2000.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518. IEEE Computer Society, 2001.

Y. Wu, Q. Tian, and Thomas S. Huang. Discriminant-em algorithm with application to image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1222–1227. IEEE Computer Society, 2000.

Jun Yang and Alexander G. Hauptmann. A framework for classifier adaptation and its applications in concept detection. In *Proceedings of the 1st International Conference on Multimedia Information Retrieval*, pages 467–474. ACM, 2008.

Jun Yang, Rong Yan, and Alexander G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *Proceedings of the ACM International Conference on Multimedia*, pages 188–197. ACM, 2007.

Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. IEEE Computer Society, 2010.

Qi Zhang, Sally A. Goldman, Wei Yu, and Jason Fritts. Content-based image retrieval using multiple-instance learning. In *Proceedings of the International Conference on Machine Learning*, pages 682–689. Morgan Kaufmann Publishers Inc., 2002.

Tong Zhang, Jun Xiao, Di Wen, and Xiaoqing Ding. Face based image navigation and search. In *Proceedings of the ACM International Conference on Multimedia*, pages 597–600. ACM, 2009.

Tong Zhang, Hui Chao, Chris Willis, and Dan Tretter. Consumer image retrieval by estimating relation tree from family photo collections. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 143–150. ACM, 2010.

Wenyi Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

Xiang S. Zhou and Thomas S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.