University of Miami
Scholarly Repository

Open Access Dissertations

Electronic Theses and Dissertations

2013-12-15

Association Affinity Network Based Multi-Model Collaboration for Multimedia Big Data Management and Retrieval

Tao Meng
University of Miami, mengtao8688@gmail.com

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_dissertations

Recommended Citation

Meng, Tao, "Association Affinity Network Based Multi-Model Collaboration for Multimedia Big Data Management and Retrieval" (2013). *Open Access Dissertations*. 1115.
https://scholarlyrepository.miami.edu/oa_dissertations/1115

This Open access is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarly Repository. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of Scholarly Repository. For more information, please contact repository.library@miami.edu.

UNIVERSITY OF MIAMI

ASSOCIATION AFFINITY NETWORK BASED MULTI-MODEL COLLABORATION FOR MULTIMEDIA BIG DATA MANAGEMENT AND RETRIEVAL

By

Tao Meng

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

December 2013

©2013
Tao Meng
All Rights Reserved

UNIVERSITY OF MIAMI

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

ASSOCIATION AFFINITY NETWORK BASED MULTI-MODEL COLLABORATION FOR MULTIMEDIA BIG DATA MANAGEMENT AND RETRIEVAL

Tao Meng

Approved:

____
Mei-Ling Shyu, Ph.D.
Professor of Electrical and
Computer Engineering

____
Xiaodong Cai, Ph.D.
Associate Professor of Electrical
and Computer Engineering

____
Saman Aliari Zonouz, Ph.D.
Assistant Professor of Electrical
and Computer Engineering

____
Nigel John, Ph.D.
Lecturer of Electrical and
Computer Engineering

____
Shu-Ching Chen, Ph.D.
Professor of School of Computing
and Information Sciences
Florida International University

____
M. Brian Blake, Ph.D.
Dean of the Graduate School

MENG, TAO (Ph.D., Electrical and Computer Engineering)
Association Affinity Network Based Multi-Model Collaboration for Multimedia Big Data Management and Retrieval (December 2013)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Mei-Ling Shyu.
No. of pages in text. (185)

With the rapid development of smart devices and ever-increasing popularity of social media websites such as Flickr, YouTube, Twitter and Facebook, we have witnessed a huge increase of multimedia data. Recently, social media technology and big data have converged to provide rich content on what happens around the world via texts, images, videos, audios, etc. Given the enormous volumes of multimedia data, efficient and effective retrieval of relevant information according to users' needs poses great challenges for traditional text-oriented storage and retrieval systems. Since manually annotating and managing the huge amount of information have become infeasible, data-driven approaches have received more and more attention. As a result, mining interesting patterns and human understandable semantic features automatically from raw multimedia data to facilitate large-scale knowledge discovery and information retrieval has become an essential research task in today's multimedia big data analysis.

One of the central problems in multimedia big data analysis is automatic data annotation. The automatic annotation for human readable patterns such as the semantic concepts in video or image data sets provides the foundation for content-based search and retrieval. The biggest challenge that the researchers face now is the semantic gap problem,

which is the gap between low-level features and high-level concepts. A lot of efforts have been made in bridging this gap in the multimedia research field. This dissertation mainly focuses on utilizing content information and inter-label correlations to improve annotation accuracy. The multimedia data annotation problem is first converted to a multi-label or a multi-class classification problem. Next, in order to model correlations among labels mathematically, we design an association affinity network (AAN) to capture such correlations.

In multimedia data sets, the label correlation is usually hidden information. In addition, a large number of associations can be noisy. The scores from different models are also of different scales. Facing these challenges, we propose several steps in utilizing the AAN to address these issues. First, the output scores from different models are normalized using the Bayesian posterior probability approach. Second, the nodes and links are modeled properly under a given application scenario. Third, a link selection module is proposed to filter the noisy links using association rule mining and correlation mining. Next, the weights of different links are computed using different models, such as the collaboration model and the regression model. Finally, the newly computed scores are used for classification and/or data retrieval purposes. In addition, negative correlations, which have rarely been explored, are studied and utilized in this dissertation.

The proposed framework is applied to two real-world applications: the multi-label high level semantic concept detection and the multi-class biomedical image temporal stage annotation. Experiments utilizing the benchmark data sets, such as the TRECVID semantic indexing data sets and the IICBU biomedical image data set, have been conducted to evaluate the effectiveness of the proposed framework. Generally speaking, the proposed

framework achieves promising results. The contributions of different components are evaluated. The experimental results demonstrate that modeling and utilizing the inter-label correlation properly could help improve multimedia data annotation accuracy and bridge the semantic gap. As the extensions of the existing framework, several future research directions are also proposed.

To my dear parents

Acknowledgments

I would like to thank my dissertation supervisor and chairman of the committee Dr. Mei-Ling Shyu for her support, suggestions and guidance in my Ph.D. study. Also my great appreciation goes to Dr. Shu-Ching Chen of the School of Computing and Information Sciences at Florida International University (FIU) for his guidance and suggestions. In addition, my thanks go to Dr. Xiaodong Cai, Dr. Nigel John and Dr. Saman Aliari Zonouz of the Department of Electrical and Computer Engineering at University of Miami for accepting the appointment to the dissertation committee and their consistent help and suggestions. I would also like to thank the department for supporting my Ph.D. study. Moreover, my special thanks go to the National Oceanic and Atmospheric Administration and TCL Research America for the financial support throughout my Ph.D. study. I would like to thank all my friends and colleagues in the Data mining, Database and Multimedia Research Group at UM and the Distributed Multimedia Information Systems Laboratory at FIU, for their help, consideration and support, especially when I met difficulty in research. My thanks also go to Mr. Le An, Dr. Xiaochen Tang, Mr. Liang Peng, Dr. Lifan Guo, Ms. Jia Huang, Mr. Jia Xu, Mr. Yang Liu, and Ms. Ming Ma.

Last but not the least, I would like to extend my utmost gratitude to my parents for their love and encouragement. It would have been impossible for me to finish this work without their consistent help and support.

TAO MENG

University of Miami

December 2013

Contents

List of Figures

List of Tables

Chapter 1

Introduction

At present, it is estimated that about 4 zettabytes of electronic data are being generated each year by everything from a DNA sequencing device in a biological laboratory to a digital camera on a family party to an electronic sensor on an aircraft [1]. The deluge of data presents both opportunities and challenges. On one hand, the large collection of data poses unprecedented opportunities by offering more information to researchers. By scrutinizing these data, people can learn deep insights and identify patterns to improve productivity. On the other hand, the rapid growth of data outpaces the development of data storage and searching technologies, leading to tremendous difficulties of retrieving useful information efficiently. Accordingly, the research about harnessing the power of “big data” has received significant academic, public, and media interest. Major technology associations, such as the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM), have started special conferences or workshops for big data in recent years. In addition, a number of research programs are funded to handle big data in different economic sectors, such as services and manufacturing, in United States [1].

Despite the popularity of the term “big data”, there is no standard definition of what big data is. For example, in the Meta (now Gartner) report [2], big data are described to be big in three aspects, which are volume, velocity, and variety, summarized as “3Vs”. Specifically, volume indicates the amount of data, velocity denotes the rate of which data are produced, and variety refers to the range of formats and representations of data. According to another source [3], big data are defined as a large amount of information of different forms which require special computing facility to be handled and analyzed. In [4], big data are referred to as large-volume, complicated, ever-growing data sets from many autonomous information sources. The national institute of standards and technology (NIST) gives a definition by describing big data as the deluge of data which exceed the capacity or capability of current or conventional methods and systems [3]. Intel further provides concrete figures by linking the concept to organizations which generate a median of 300 terabytes of data weekly [5]. Some other definitions can be found in [6], [7], and [8].

According to these definitions, some characteristics of big data can be summarized as follows. First, the volume of big data is huge. Second, they are generated in a fast rate. Third, they are complex and usually come in different formats. Last but not the least, the big data are beyond the capability of the commonly used tools to process in a reasonable amount of time. Special platforms or solutions are necessary to handle them.

Among all types of data today, multimedia data, like texts, images, videos, etc., are probably the most common data for the general public. Multimedia data contains information that can be represented through audio, graphics, images, videos, and animations in addition to traditional media (text and graphics) [9]. Thanks to the rapid development of social media websites, the popularity of smart mobile devices such as the smart phone and digital camera, and the continuous price drop per unit of storage, multimedia data have become ubiquitous. They carry the aforementioned four characteristics of big data. They are elaborated as follows.

First, the amount of raw data is huge. One reason is the easiness and low cost of creating, storing, and sharing of multimedia data make them ubiquitous. For example, Facebook users have uploaded more than 250 billion photos [10], and more than 16 billion photos are shared on Instagram [11]. Even in a relatively small-scale open source biomedical image database, the BDGP database [12], the number of microscopic images has already hit 115,647. The other reason is that the disk space one multimedia data instance occupies is relatively large. For example, a 640 by 480 color image with three channels could occupy up to 921,600 bytes without compression.

Second, multimedia data have been growing at a phenomenal rate. For instance, about 1.54 million photos are uploaded to Flickr on a daily basis [13]; 100 hours of videos are uploaded to YouTube every minute [14]; and up to 4,000 photos are uploaded to Facebook per second [10].

Third, multimedia data are complex. They usually contain data of different formats. For example, videos on YouTube websites contain visual information (frames), audio information (sounds), and contextual information (metadata). The three types of data are integrated naturally to convey information. In addition, multimedia data are unstructured by nature. There are no well-defined attributes with precise and non-ambiguous meanings. In order to derive attributes that provide content information, human annotations are needed. However, human subjectiveness leads to varied interpretations for the same data instance, which causes problems for data retrieval.

Fourth, the deluge of multimedia data requires special platforms for processing. Facing the massive amount of heterogeneous data, the major social media websites have deployed many state-of-the-art technologies such as NoSQL [15] and Hadoop [16] for data storage and processing.

Besides these features, multimedia data are big data because they are valuable sources for knowledge and information [17]. Compared to the traditional text-based information representation, multimedia data have richer content and are more user friendly. Nowadays, multimedia data play an important role in many different aspects of modern life. Quite a few multimedia applications and services have been developed to utilize different multimedia data for entertainment, online education, content sharing, electronic commerce, security monitoring, and medical diagnosis. For example, relying on the face detection algorithm, some social network websites such as Facebook can detect faces appearing in the photo automatically and provide the facility for photo tagging. In the biomedical domain, many biomedical image analysis software tools, such as OMEGR [18], provide the facility to perform automatic medical image analysis and diagnosis. In the security domain, face recognition techniques can be utilized to identify suspects from a surveillance video automatically and efficiently. Because of their broad applications, multimedia data management and retrieval have become an important research direction in both academia and industry domains.

Facing the sheer volume of the multimedia data, one of the important problems is how to store and index them in order to perform efficient search and retrieval based on users' queries. In order to solve this problem, multiple labels need to be given to the multimedia data to represent them in the database. One way of assigning these labels is manual annotation. However, the solution does not scale well and also suffers from human subjectivity. Therefore, the data mining-based approaches have received a lot of attention since they aim to identify the significant patterns and discover knowledge from raw data with minimal human interaction.

Data mining refers to the process of automatically finding interesting patterns in data that are not ordinarily accessible by basic queries with the objective of utilizing these patterns to improve decision making. For example, it might not be easy to retrieve all the videos related to politics from the video database without viewing each one in that database manually. However, the classification models that are trained properly using spatial and temporal features from videos could discover video clips or scenes related to politics automatically and return the results to the users.

Classification is one of the well defined research problems in the data mining research field. Classification algorithms aim at labeling the data using a set of predefined categories. Classification techniques find their applications in automatic multimedia data annotation. In the multimedia research area, labels corresponding to high-level semantic features such as “face,” “sky,” and “car” are also called semantic “concepts.” In Figure 1.1, a testing image can be annotated automatically by the trained concept detectors. In Figure 1.2, the classification system can automatically determine which type of cancer a patient suffers from based on histopathology images. The automatic annotation of the multimedia data is an important component for multimedia data analysis and provides the foundation for content-based multimedia search and retrieval.

Incorporating the data mining techniques into the multimedia retrieval area brings in strengths along with challenges. In terms of strength, the rich content in image, video, audio, and text files provides more information for data mining researchers to investigate than text data alone. On the other hand, the multimedia data, which are by nature unstructured, are more difficult to analyze. Recently, researchers find that the biggest challenge is to bridge the semantic gap [19], which refers to the lack of connection between the information that one can extract from the visual data and the interpretation

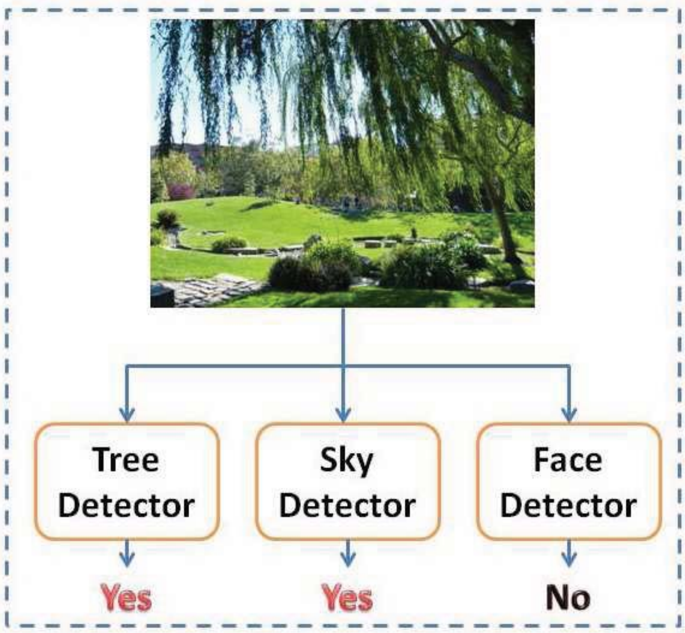


Figure 1.1: An example of labeling concepts in an image automatically

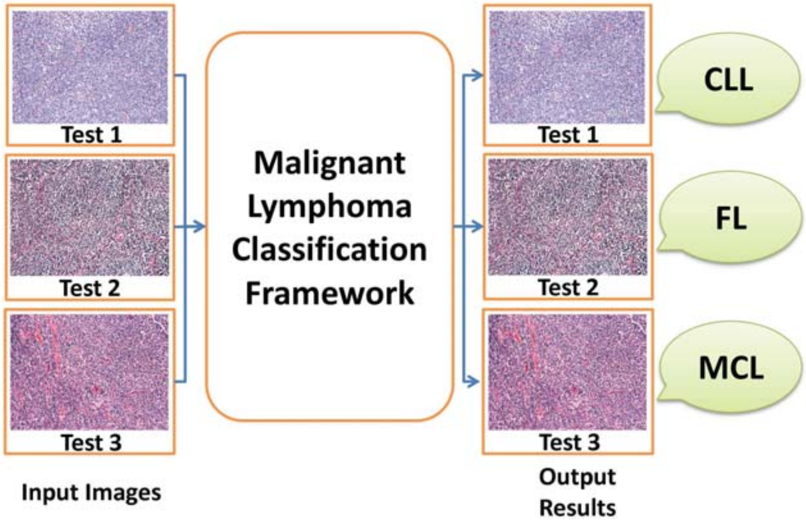


Figure 1.2: An example of lymphoma image classification

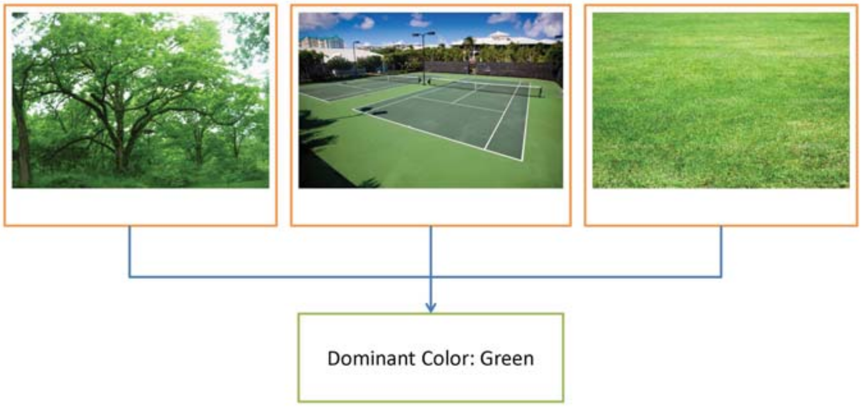


Figure 1.3: An example of dominance of green in three images containing different concepts

that the same data have for a user in a given situation. In other words, while a human understands the semantics in the multimedia data, a computer gets the information of the low-level features such as color, texture, energy, and pitch. Because of the lack of one-to-one correspondence between the low-level features and the high-level semantics, data annotation relying entirely on the low-level features could be problematic. For example, as shown in Figure 1.3, the dominance of the color "green" in an image could indicate the image depicts a tree, a tennis court or a soccer field.

In order to address the semantic gap issue, a lot of research efforts have been devoted to extracting sophisticated features such as the scale invariant feature transform (SIFT) and the histogram of oriented gradients (HOG), increasing the ratio between positive data instances and negative data instances, and improving the classifiers for multimedia data annotation [20][21][22]. While all these efforts have pushed forward the frontiers of knowledge and contributed to the improvement of semantic label annotation, most of these approaches treat the annotation problem as a binary classification problem or a multi-class classification problem. However, labels are inter-correlated. Using Figure 1.4 as an example, the labels "sky" and "cloud" often co-occur in the same image

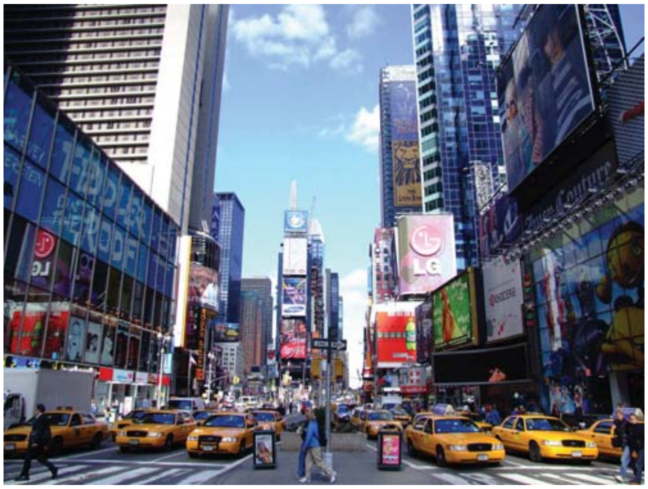


Figure 1.4: Co-occurrence of “sky” and “cloud” in an image

or video clip. Therefore, if an image is labeled as “sky,” it is highly likely that it also contains the label “cloud.” Therefore, modeling the correlations between labels properly to improve the classification accuracy has become an important research direction in multimedia data annotation and received a lot of attention. In this dissertation, a framework that utilizes label correlations to assist automatic annotation of multimedia data is proposed.

In the multimedia research domain, the semantic labels are also named concepts. Therefore, in this dissertation, the terms “concept” and “label” are used interchangeably.

1.1 Motivations and Challenges

In most real-world applications, concepts often exist correlatively with each other, rather than appearing in isolation. Therefore, individual annotation only achieves limited success. For example, in a video database, the presence of the label “crowd” often occurs together with the presence of “people,” while “boat_ship” and “truck” do not co-occur commonly. On the other hand, compared to simple labels that can be directly modeled from low-level features, some complex labels, for example, "people-marching," are extremely difficult to model individually due to the semantic gap between these labels and low-level features. Instead, these complex concepts can be better inferred based on the label correlations with other concepts. For instance, the presence of "people-marching" can be boosted if both "crowd" and "walking_running" occur in a video. These observations motivate us to explore the usage of inter-label correlation for improving the accuracy of multimedia data annotation.

Utilizing inter-label correlation for improving the accuracy of multimedia data annotation is a relatively new research direction in multimedia data mining. In the field of multimedia semantic concept detection, which is essentially a multi-label classification problem, there has been research that modeled the correlations between different concepts using model vectors [23], ontology [24], etc. Although previous studies show improvement of performance by incorporating the inter-concept relationship, there are still several challenges to be addressed.

- Correlation selection challenge, *i.e.*, the challenge of selecting the significant correlations among different concepts that could help improve the annotation accuracy

In a large multimedia database, there are many correlations among different labels. Some correlations are natural and can be utilized for enhancing the accuracy of multimedia annotation, while others are casual and they could adversely affect the performance if they are also considered. Therefore, how to select significant correlations becomes an important research problem. The most intuitive solution is to rely on domain knowledge. While domain knowledge is helpful in a sense, it may not adapt to a specific data set.

- Score heterogeneity challenge, *i.e.*, the challenge of handling scores output from distinct models

In the field of multimedia data mining and retrieval, the annotation could be categorized as the hard label or the soft label. The hard label indicates whether an instance belongs to a certain class explicitly. On the contrary, the soft label gives a confidence score to an instance, which shows how likely that instance contains a certain class label. Compared with hard labels, soft labels can not only avoid the process of threshold determination but also can serve as a foundation for retrieval. Therefore, the models used in multimedia data mining usually give the soft labels. In practice, the researchers usually use different models for different concepts. For example, the Haar cascade classifier is routinely used for detecting the concept of "face," while the concept "sky" is usually detected by the support vector machine classification model. As the scores from the two models are heterogeneous by nature, it does not make sense to fuse them directly. Therefore, one research challenge is to come up with proper normalization strategies for heterogeneous scores output from different models.

- Weight computation and score integration challenge, *i.e.*, the challenge of computing weight, which represents how much influence one concept model could exert on the other concept model, and further, the challenge of integrating the weights and scores together

Supposing the related models are selected, it is important to come up with a systematic strategy to decide how much one concept model should affect the other one. In order to model the influence, the weights need to be represented numerically. These weights are important for combining different models and should be computed with caution.

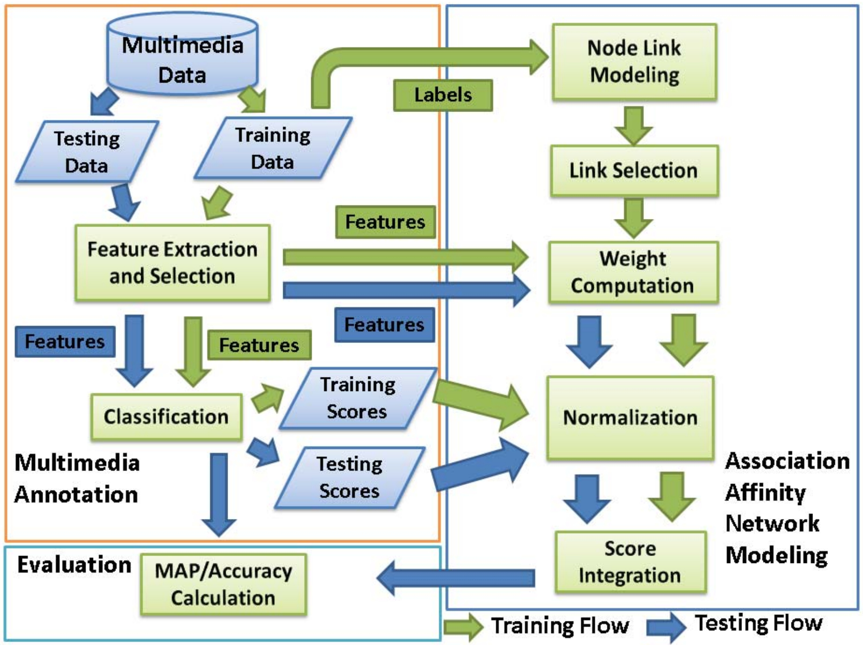


Figure 1.5: The proposed framework

- Multi-class model collaboration challenge, *i.e.*, the challenge of using label correlations in multi-class classification problems

Multi-class classification techniques are widely used in multimedia data annotation, especially in biomedical image annotation such as labeling the subtypes of cancer cells. Usually, class labels are mutually exclusive, and one instance could only be assigned to one class label. However, in some applications, several classes could be grouped together to form a middle class. As the classification problem for the middle classes is usually simpler and the classification accuracy is higher, the results from middle class classification could be used to improve the overall classification performance. Under this scenario, how to form middle classes and how to integrate output from middle classifiers are two challenging problems.

1.2 Proposed Solutions

In order to address the aforementioned challenges, a multi-model collaboration framework is proposed to make use of the correlation among different models. In this section, the proposed framework is introduced on a conceptual level. The specific details of each module and mathematical deduction are provided in Chapter 3 and Chapter 4. Figure 1.5 shows the architecture of the proposed framework. It generally consists of three components described as follows:

1.2.1 Multimedia Annotation Component

The “Multimedia Annotation” component follows the classic data mining procedure. Multimedia data are preprocessed to eliminate noise and separated into a training data set and a testing data set. Next, a set of numerical or nominal features are computed for each data instance. A feature selection step is useful for filtering the features which are not relevant; the classification model receives the selected features as well as labels and computes the confidence scores for both training data and testing data. The “Multimedia Annotation” component is added here to illustrate the global picture of the framework but is not the focus of this work.

1.2.2 Association Affinity Network (AAN) Modeling Component

The AAN refers to the graphic model representing the class labels and correlation between labels. Figure 1.6 shows a schematic example of the proposed AAN. In this figure, C_1, C_2, ..., C_5 indicate five different class labels. They are represented as the nodes in an AAN. The links represent the correlation between two concept labels. There are generally two kinds of links, positive links and negative links. $A_{p,q}$ indicates the affinity of the link from node C_p to node C_q. The positive link indicates that the two class

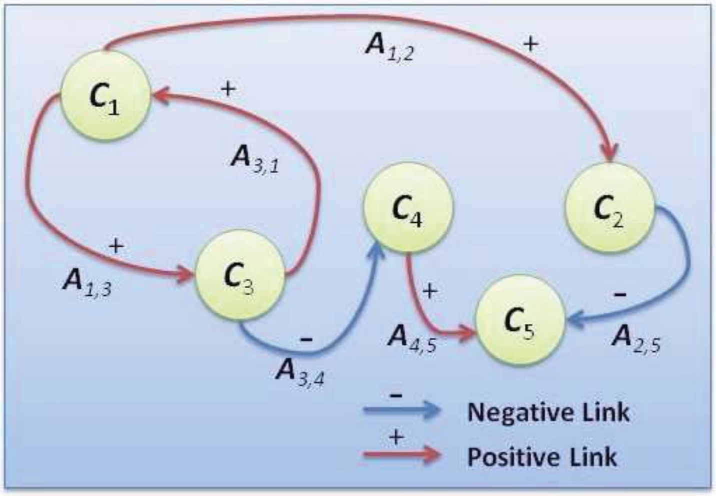


Figure 1.6: A schematic figure of an AAN

labels are more likely to appear together and the affinity is positive, while the negative link indicates that the two class labels usually do not appear together and the affinity is negative. Figure 1.7 shows an example of an AAN in the multimedia concept detection application. In this figure, weights of links are omitted for better visibility, and only positive links are demonstrated. This figure gives an overview of the proposed AAN.

The AAN is the main focus of this dissertation. In order to illustrate the idea of an AAN more clearly, the steps of constructing and utilizing AAN are introduced. They are summarized as five main modules from "Node Link Modeling" to "Score Integration" on a conceptual level. These modules discover the significant relationship between different class labels and utilize them in the final score integration step. The five modules are introduced as follows.

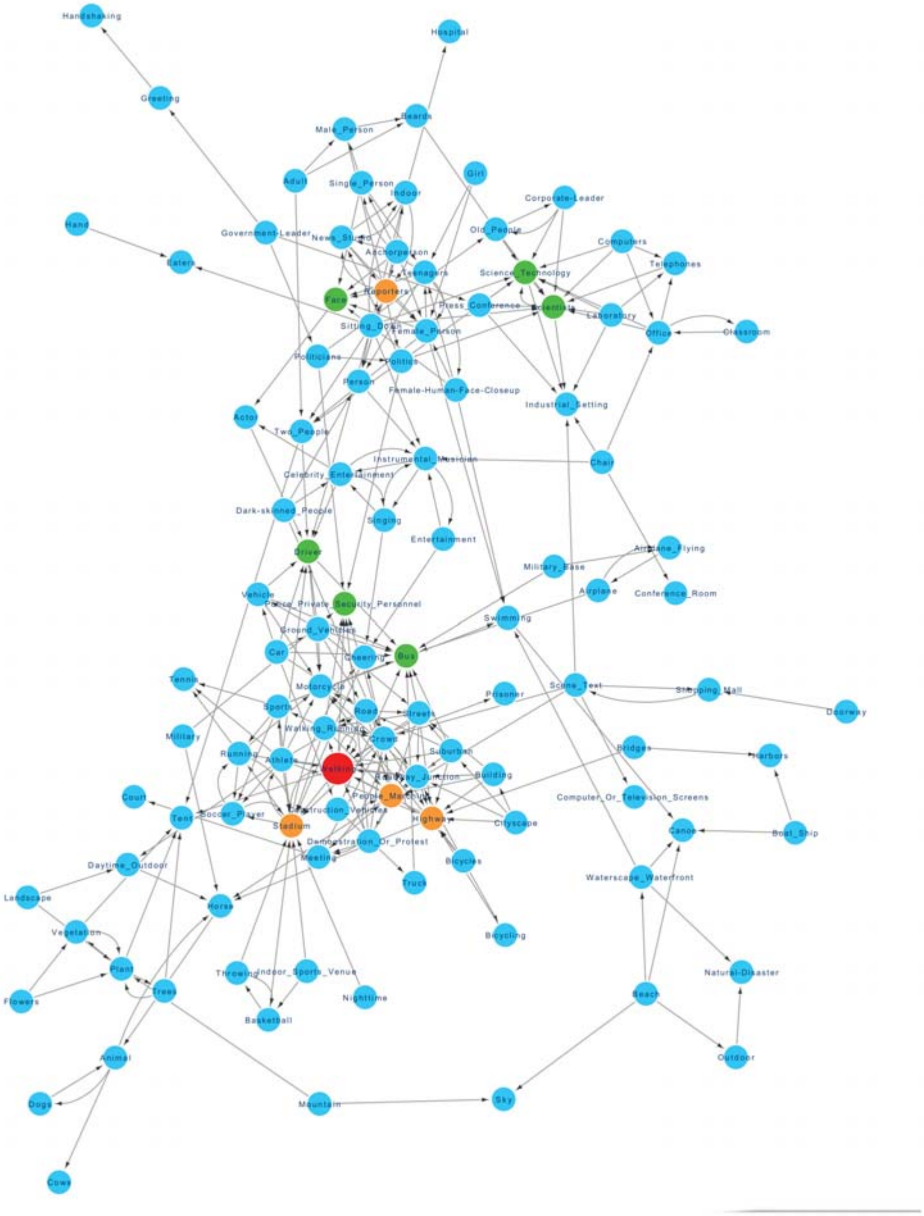


Figure 1.7: A sample AAN in multimedia concept detection

- The “Node Link Modeling” module forms the backbone of an AAN.

The first step of building the AAN is to identify the nodes and links in a specific problem. Proper modeling of the nodes is one of the key factors to the success of the application of an AAN. In the multimedia high-level semantic concept detection problem introduced in Chapter 3, one data instance could be annotated with multiple concepts. Therefore, each node represents one semantic concept such as “road,” “car” or “sky”; the link between two nodes represents the correlation between two semantic concepts. For example, the “car” and “road” often co-occur in a shot, so there is a positive link from “car” to “road” and vice versa. In the multi-class classification task for annotating the temporal stages in biomedical data sets introduced in Chapter 4, the modeling of the “node” is trickier. Here, categories of the original multi-class classification problem are named as end classes. On the other hand, newly “created” classes are defined as middle classes. Both middle classes and end classes are modeled as nodes. Since only the end classes are of interest, only links from the middle classes to end classes are modeled. In summary, this node and link modeling module form the backbone of the AAN and should be carefully designed under different scenarios.

- The “Link Selection” module filters irrelevant links to keep significant links.

After nodes and links are modeled, the next step is to select helpful and significant links from all possible connections. The reasons for doing so are twofold. First, filtering the noisy input from the casual links can improve the performance of the framework. Second, decreasing the number of links can decrease the computational complexity. The filtering process falls into two categories: the hard selection and the soft selection. In the hard selection approach, some links are labeled as useless and filtered completely.

This approach is usually adopted when there is a relatively large number of links. In Chapter 3, the association rule-based link selection module is proposed. On the other hand, soft selection uses the numerical measurement to model the importance of each link, and there is no hard cut-off to remove links. This approach is more precise and is used when there is a relatively small number of links. If the soft selection approach is used, the "Link Selection," and the "Weight Computation" modules are actually combined into one module. In the multi-class classification framework in Chapter 4, the soft selection approach is utilized.

- The "Score Normalization" normalizes the heterogeneous scores and makes them comparable.

As scores from the multimedia annotation component could be heterogeneous, a proper normalization step is needed. The common normalization approaches include min-max normalization, Z-score normalization, etc. These approaches target at converting all values into the same range. However, they ignore the physical meaning of the score value. Accordingly, the Bayesian posterior probability score normalization approach is proposed in Chapter 3. It should be pointed out that this module is optional. If training scores and testing scores are generated by the same models and they are already probabilities, this step could be skipped. For the generated weights, normalization is sometimes necessary to convert them into the same range.

- The "Weight Computation" computes the weights of the links.

This module tries to compute the weights of selected links. In this procedure, the weights are computed depending on the normalized training scores and labels. Different strategies of computing weights, such as the probability-based approach and regression-based approach, are proposed. A novel multi-class collaboration framework is also proposed. In addition, compared with labels that represent the prior probabilities of the occurrence of different classes, features provide data-specific information. A feature-based weight model is incorporated to take features into consideration. In the negative correlation-enhanced module, a multiple correspondence analysis-based (MCA-based) weight estimation modeling is proposed.

- The "Score Integration" computes the final score after applying the AAN.

This module receives normalized testing scores and utilizes the established AAN to integrate the testing scores from related nodes together. The regression-based models are used to integrate the scores together. The logistic regression estimates the posterior probability and serves as a proper module for score integration.

The five modules form the core of the proposed AAN. The AAN is a relatively flexible model, so it can adapt to different multimedia annotation frameworks. After the new scores are computed, they are used for framework evaluation.

1.2.3 Evaluation Component

In order to evaluate the performance of the framework, proper measurement of the framework is desired. Since the classification models could output the hard labels like classification results or soft labels like confidence scores, different evaluation criteria are designed for different cases.

For the binary classification problem, a set of evaluation criteria are defined as follows.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}; \tag{1.1}$$

$$Precision = \frac{TP}{TP+FP}; \tag{1.2}$$

$$Recall = \frac{TP}{TP+FN}; \tag{1.3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}; \tag{1.4}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TN+FN)(TP+FN)(TN+FP)}}. \tag{1.5}$$

Here, TP, FP, TN, and FN stand for the number of true positive instances, which are the number of items correctly labeled as positive class; false positive instances, which are the number of items incorrectly labeled as positive class; true negative instances, which are the number of items correctly labeled as negative class; and false negative instances, which are the number of items incorrectly labeled as negative class. *Accuracy* represents the ratio of the correctly classified instances to the total number of instances. *Precision* represents the quality of the prediction for the positive class. *Recall* represents the capability to retrieve positive instances from all the data instances. $F1$ is the harmonic mean of *Precision* and *Recall* and balances the two evaluation criteria. The *MCC* stands for the Matthew's correlation coefficient. The range of *MCC* is from -1 to 1. $MCC = 1$ indicates that the algorithm gives the best possible predictions, while $MCC = -1$ indicates that the algorithm gives the worst possible predictions. The *Accuracy* and *MCC* evaluate the performance of the framework from the perspectives of both the positive instances and the negative instances. $F1$ focuses on the positive instances only.

For the multi-class classification, the most commonly used criterion is the classification accuracy. Assume there are K classes and the number of correctly classified instances in class k is TP_k, m is the total number of instances, the accuracy is defined in the following equation. It could be seen that when n equals 2, it is the same accuracy defined in 1.6.

$$Accuracy = \frac{\sum\limits_{k=1}^{K} TP_k}{m} \tag{1.6}$$

In terms of soft labels, the classification models output a score that measures the confidence of the model's prediction. From the classification point of view, the performance of a classifier needs to be evaluated from a global perspective. The precision-recall curve and area under this curve (AUC) give a comprehensive measurement of classification performance. The precision-recall curve is a plot with the x-axis as recall values and y-axis as precision values. These values are computed by setting different thresholds. Figure 1.8 shows an example of a precision recall curve. The AUC is also given in the figure.

From the perspective of information retrieval, users want to see the most relevant data in top retrieved results. Therefore, the ranking of the results matters. In order to capture this information, average precision (AP) is a widely used metric. Assume for a given query, ψ represents the number of the retrieved instances and G_n represents the total number of relevant instances in a database. $Pre(a)$ indicates the precision at

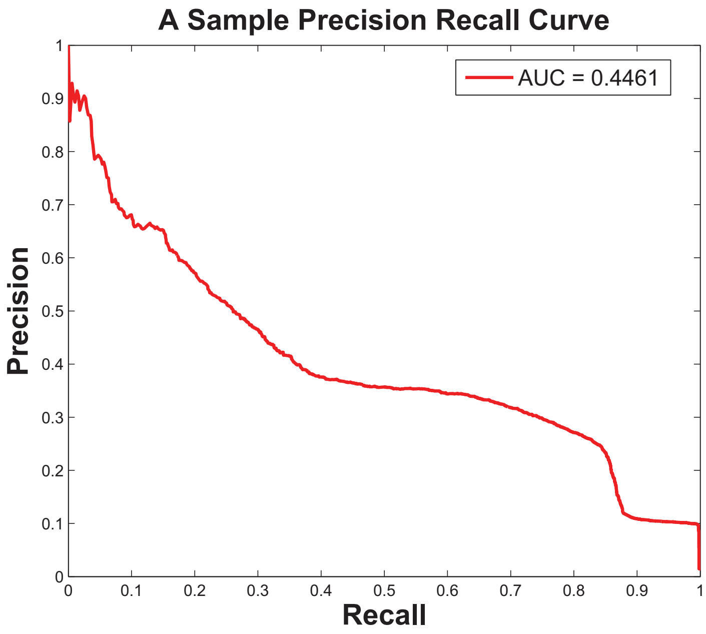


Figure 1.8: A sample precision recall curve

the a-th retrieved instance. $Min(G_n, \psi)$ indicates the minimal value of G_n and ψ. The average precision at ψ (i.e., $AP@\psi$) is defined in Equation 1.7.

$$\begin{aligned} AP@\psi &= \sum_{a=1}^{\psi} \frac{Pre(a) \times rel(a)}{Min(G_n, \psi)}. \\ \text{where,} rel(a) &= \begin{cases} 1, & \text{if instance } a \text{ is relevant,} \\ 0, & \text{if instance } a \text{ is irrelevant.} \end{cases} \end{aligned} \tag{1.7}$$

The mean value of AP among all queries is defined as the mean average precision (MAP).

1.3 Contributions and Limitations

Generally speaking, there are four main contributions and highlights of the current framework, which are summarized and listed as follows.

1. A novel framework of building and utilizing the AAN to model the label correlation is designed.

   The label correlation provides useful context information to assist multimedia concept detection. The AAN is designed to model the correlation of different labels. Facing multiple challenges such as noisy associations and heterogeneity of scores, five modules are designed to address these difficulties. To the best of our knowledge, this is the first work to utilize the AAN to model the inter-label relationship in the multimedia data mining field. In addition, negative correlations, which have not been studied well, are investigated thoroughly in this dissertation. They are also utilized to help concept detection.

2. A Bayesian posterior probability-based module for normalizing the heterogeneous scores is proposed.

   The confidence scores generated by different models could be heterogeneous. In order to address this issue, a Bayesian posterior probability-based module is invented to normalize the score. Compared with the commonly used normalization approach, the proposed approach not only converts the scores into the same range but also contains the information of model accuracy.

3. Both supervised and unsupervised middle classes modeling approaches are proposed to assist multi-class classification. A closed-form cost function is defined to model the multi-class classification error, and numerical solutions are provided to estimate relevant weights.

   Under the multi-class classification scenario, the categories are mutually exclusive, so each instance belongs to one class. Inspired by the strategy of "divide and conquer," we propose two approaches to form middle classes, which are relatively easy for classifiers to handle. Next, based on the output from middle classes and end classes, we propose a multi-layer score integration approach based on solving a convex optimization problem. A novel cost function is defined for this problem, and a systematic optimization algorithm is proposed.

4. A new multi-model collaboration for weight computation of links for the multi-label classification problem in multimedia concept detection is presented.

   As discussed in Section 1.2.2, the weights of the links in the proposed framework are important. We propose a multi-model collaboration approach to compute the weights inspired by the idea of collaborative filtering. Using this approach, the labels and the scores are taken into consideration at the same time. To the best of our knowledge, this is the first approach that utilizes collaborative filtering in assisting the multi-model collaboration in this domain. The experimental results indicate the effectiveness of the proposed multi-model collaboration strategy.

There are some limitations of the proposed framework:

- The proposed framework is based on the assumption that the distributions of training data and testing data are the same.

  As in many data-oriented approaches, the proposed framework relies on the assumption that the distribution of the training data and testing data are the same. In the real-world application, many applications conform to this assumption. For example, in order to analyze a relatively large multimedia database, we can randomly sample a significant amount of data as the training data set to train our model and use the model to make predictions for the rest of the data. The current framework has difficulty in handling the so-called domain change problem [25]. Actually, the domain change problem is another research direction that should be explored in the data mining area.

- The proposed framework relies on the classifiers, which give numerical values to measure confidence of classification.

  As the confidence scores output from classifiers are used as the inputs into our proposed framework, classification models that do not have the capability to output the confidence scores can not be used. Currently, most of the state-of-the-art classification models such as logistic regression, support vector machine, neural network, etc. could not only give the hard label but also outputs the numerical measurement of the likelihood that a data instance belongs to a certain class. However, there are some classification models that output decision labels only. For example, the one-rule classifier [26] only outputs classification results without any probabilistic information.

- The supervised middle class creation strategy used in the multi-layer collaboration model capitalizes on domain knowledge.

  The supervised middle class creation strategy now used in the multi-layer collaboration model relies on domain knowledge. Specifically, because the classes, such as the temporal stages, have a sequential relationship, the supervised middle class creation strategy incorporates the sequential information in the grouping process. This piece of domain knowledge is specific in the biological stage annotation task.

- Some parameters and thresholds are determined based on iterative search.

  In the proposed framework, there are some parameters that need to be tuned by using the iterative search approach. For example, in the association rule-based link selection module, the thresholds for support ratio and confidence ratio need to be tuned using the grid search approach to maximize the cross-validation accuracy for each individual concept. The search process increases the time complexity of the training process. Therefore, the proposed approach currently works offline in the training stage.

1.4 Outline of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 contains the literature review on multimedia big data mining, enhancing high-level semantic concept detection using the inter-concept relationship, and the existing approaches addressing multi-class classification. As we are going to work on the biomedical image data set, a specific section of the literature review is about the current state-of-the-art biomedical image classification approaches.

In Chapter 3, the framework of using the AAN in multimedia high-level semantic concept detection is introduced. The implementation details for each component of the AAN under the multi-label classification scenario are introduced. The two enhancement modules, which are the multi-model collaboration module and the Pearson product moment correlation-based filtering module, are introduced based on the original framework. Experimental results are provided to evaluate the proposed framework and the enhancement.

Biomedical image informatics is an important interdisciplinary field that combines multimedia research, data mining, machine learning and biomedical science. In Chapter 4, the proposed AAN is applied to solve a multi-class classification problem in the biomedical image mining domain, to annotate the temporal stage information of different biomedical imaging data. Compared with the multi-label problem addressed in Chapter 3, there is relatively less work in using the label relationship in a multi-class classification problem because of its difficulty. We design several modules to address the specialty of the multi-class classification problem and apply our proposed AAN in this task. Experimental analysis shows that the proposed framework achieves better performance.

Chapter 5 concludes the dissertation by summarizing the overall framework and providing details of future directions to be investigated.

Chapter 2

Literature Review

In this chapter, we provide a thorough literature review on the related work. In Section 2.1, recent progress of multimedia big data mining is introduced. In Section 2.2, the state-of-the-art approaches of utilizing the inter-concept relationship to help high-level semantic concept detection are reviewed. In Section 2.3, the common approaches for the multi-class classification algorithms are introduced. Because a significant part of my research work focuses on biomedical image classification, a literature survey is provided in this area in Section 2.4.

2.1 Large-Scale Multimedia Data Mining

As huge amounts of multimedia data are generated every day, it is infeasible to inspect and annotate them manually. Therefore, data mining technology is necessary for identifying significant patterns and extracting useful information from raw multimedia in an efficient way. In a sense, automatically analyzing huge amounts of multimedia data can be viewed as one specific application of big data mining technology. Therefore, in order to present a complete picture, the motivations and current progress of large-scale data mining are first introduced in Section 2.1.1. Following that, the recent research efforts in multimedia big data analysis are summarized in Section 2.1.2. In order to help researchers in this domain, commonly used large multimedia data sets are surveyed in Section 2.1.3.

2.1.1 Data Mining for Big Data

In the big data era, about 2.5 quintillion bytes of data are created every day [27]. Such massive amounts of data lead to the question: how can people harness the power of these data to gain knowledge, enhance quality of life, win competitive advantages, and improve productivity? To answer this question, the most fundamental challenge is to efficiently sift through the large volumes of data and extract useful knowledge to guide future actions. Accordingly, data mining, which aims to discover non-trivial and useful patterns from raw data, finds its applications under such a circumstance. For most small-scale data mining tasks, a single desktop computer or a server equipped with data mining software like Weka [26] is capable of fulfilling the goal of knowledge discovery. However, the deluge of aforementioned heterogeneous data calls for scalable solutions. In fact, the core principal of designing machine learning or data mining applications geared toward big data is to make them scale well to the extraordinarily large volume of data.

There has been some studies targeting at big data mining. Most of existing work is built upon parallel programming models like MapReduce [28] and utilizes parallel computing infrastructure. In [29], a general parallel computing framework for implementing the data mining algorithms which can fit the statistical query model was proposed. Utilizing the MapReduce programming paradigm, the authors implemented 10 commonly used machine learning algorithms, including *k*-means, logistic regression, support vector machine, and so on. Ranger et al. [30] introduced the Phonex framework in which they implemented the MapReduce model on multi-core and multi-processor systems. They also provided programming interfaces for easy software integration. Three data mining algorithms, which are *k*-means, principal component analysis, and linear regression were implemented. Papadimitrou and Sun [31] designed a distributed collaborative aggregation framework to solve the data co-clustering problem. Besides its popularity in academia, the MapReduce programming model also gets a lot of attention for its potential in solving middle-scale to large-scale data mining and machine learning problems in industry. In [32], two researchers from Twitter [32] gave an example to illustrate how machine learning applications are integrated into Twitter's Pig-centric analytics stack for large-scale predictive research using Hadoop software library [16]. Sumbaly et al. [33] introduced LinkedIn's Hadoop-based analytics stack, which helps data scientists and machine learning engineers to gain insights and develop products efficiently and effectively.

Besides aforementioned studies which focus on constructing novel frameworks, a lot of research work has been carried out to enhance the scalability of classic data analysis software by integrating them with software frameworks supporting parallel processing. In [34], in order to take advantages of both the rich functionality for data analysis of the R programming language and the powerful capability of handling petabytes of data of MapReduce-based systems, a scalable platform "Ricardo" was proposed to integrate them. This relatively successful integration makes it possible to re-use many software modules built in R to conduct large-scale data analysis, which saves a significant amount of time for implementing either statistical or data-management modules from scratch. Wegener et al. [35] achieved the successful integration of Weka, which is one of the most popular open-source machine learning and data mining software tools, and Hadoop. The authors claimed that their integrated software can handle up to 100-GB data on MapReduce clusters. Other similar projects include Radoop software package which combines RapidMiner and Hadoop [36] and the HadoopML project [37] which enables researchers to develop task-parallel and data-parallel machine learning algorithms easily under the language runtime environment.

2.1.2 Mining Multimedia Big Data

Multimedia data are big data not only because they are generated at an unprecedented rate but also because they have become valuable sources for insights and information. They provide the valuable information in multiple fields, ranging from scientific exploration to personal entertainment [17]. Accordingly, one of the most important applications of multimedia big data mining is to conduct content analysis for efficient data search and retrieval. In a recent paper published in ACM Transactions on Multimedia Computing, Communications and Applications [38], the author predicted that the emerging availability of big data offers a golden opportunity for multimedia data mining researchers. As described in the paper, the recent success in audio and video retrieval relies more on large-scale data storage and computing than advances in recognition models. Therefore, how to fully utilize the opportunity provided by the multimedia big data becomes an important research problem.

In order to address issues raised in large-scale multimedia analysis, some research efforts have been made in different steps of data mining, such as feature extraction, data annotation and classification, and content-based multimedia retrieval. Some representative studies are reviewed as follows.

Since multimedia data are unstructured and heterogeneous, the first task for data mining researchers working on these data is to convert them to well-formatted data matrices. Meaningful feature representations can increase the chance of success for a data mining task. However, computational complexity of extracting the state-of-the-art local features like SIFT [22] is usually very high [39]. In order to address this issue, Hsiang et al. [40] developed the affine scale invarant features (ASIFT) and capitalized on the MapReduce framework to speed up this process. Experimental results using a Hadoop computational cluster show the superior performance of their proposed feature extraction methods in terms of accuracy and efficiency. In [41], the researchers introduced a case study to use MapReduce programming model for constructing the bag of visual words using SIFT features. Two main steps, which are the centroids selection for starting points and iterative centroid recalculation, were implemented using the MapReduce programming model. In [42], the authors proposed the approach of extracting texture features from images in parallel using the MapReduce programming model for efficient object detection. Similar studies in this direction include [43], [44], and [45].

Besides the feature extraction step, another bottleneck of enhancing the efficiency of multimedia data mining framework lies in the model training step. For most supervised classification models, a substantial amount of computational efforts are made in obtaining model parameters by scanning the training data set. In order to expedite this process, some recent studies has been carried out to utilize parallel processing techniques in building classification models. Along this line, the frameworks introduced in Section 2.1.1, which focus on enhancing the efficiency of different frameworks can be utilized in the large-scale multimedia content analysis. Alham et al. [46] proposed the MRESVM system, in which they utilized the MapReduce programing model to implement the distributed SVM ensemble algorithm for image annotation. In their experiments, the training time was significantly reduced, while the performance of classification was not degraded. Zhao et al. [47] proposed an optimization approach based on proximal approximation and a parallel gradient descent method to categorize 1.2 million images from the ImageNet [48] into 1000 categories. Given the growing need of processing streaming videos, an online machine learning framework aimed to mine video content in a real time model was proposed in [49]. Specifically, the authors extended an existing machine learning framework "Jubatus" [50] by enhancing the clients and keeper nodes.

Content-based multimedia data retrieval is another important application of multimedia data mining. In this research field, the near-duplicate image/video detection has received a large amount of attention because of the expansion of duplicated multimedia data online. Wang et al. [44] developed a large-scale multimedia data mining framework for near-duplicate video detections using the MapReduce framework. Two classes of Hadoop, which are InputFormat and RecordedReader, were mortified specifically to represent video files as original streams. They also applied *k*-means clustering algorithm using the MapReduce model and used HDFS to store features of videos. In [51], the authors proposed a framework to integrate global and local features and developed a novel hashing technique under the MapReduce framework. Their proposed system discovered about 553.8 million duplicate images from 2 billion Internet images within 13 hours on a 2000 core cluster.

2.1.3 Benchmark Data Sets for Multimedia Big Data Mining

In order to facilitate the evaluation of different frameworks in the multimedia data mining community, some large-scale multimedia data benchmarks have been developed by different research groups. For example, the Sun database [52] developed in the MIT Computer Vision group contains 131,072 images with 908 annotated scene categories. The core idea of the Sun project to construct this database is to provide a comprehensive aggregation of human annotated images covering a broad range of scenes and places. In order to help researchers to build their own projects, some data processing and evaluation codes are also provided. Another famous and larger image database, the 80 million tiny images, contains 79,302,017 images and each of which has the dimension of 32 by 32 pixels. The total size of all the images is up to 400 Gb [53]. Each image is loosely annotated with one of the 75,062 non-abstract nouns in English. One challenge of using this data set is that only a very small portion of images are manually validated for correctness. However, the large collection provides a good simulation of the real-world scenario for image semantics mining. The ImageNet database [48] built at Stanford Vision Lab holds up to 14,197,122 images belong to 21,841 synsets (classes). One salient characteristic of this database is that all images are organized in a hierarchical structure according to categories, which facilitates the evaluation of the classification framework that addresses the hierarchical multi-class classification problem. In the biomedical image research domain, the *in situ* hybridization image database in the Berkeley *Drosophila* Gemome Project (BDGP) is a large-scale microscopic image database [12]. At the time of writing, it stores 115,647 images of different expression patterns of 7,686 genes in *Drosophila*. The expression patterns of each image are annotated by domain experts. This database becomes a testbed for large-scale biomedical image retrieval.

Compared with image databases, fewer comprehensive databases are available for video clips. The annual TREC video retrieval evaluation [54] competition provides a collection of videos from social media websites like YouTube, Dailymotion, etc. Up to 2013, the training data set consists of 800 hours of videos with durations between 10 seconds to 6 minutes and the total number of videos is up to 28,123. One keyframe is provided for each video shot for the convenience of the participants. The keyframes are partially labeled by a joint annotation effort among all the participants. This data set is suitable for video content analysis and content-based retrieval. Unfortunately, this data set is only available for the TRECVID community. Another relatively comprehensive data set is the UQ_VIDEO [55] developed at the University of Queensland. This data set contains the URLs for 169,925 videos in total from YouTube website. The authors further provided HSV and LBP features for all the keyframes. Compared with the TRECVID data set, this data set is more suitable for evaluating the frameworks for near-duplicate web video retrieval. Another issue with this data set is that users need to craw the raw videos by themselves if they want to extract more information from the raw data directly.

2.2 Utilizing Inter-Concept Correlations in Multimedia Concept Detection

Multimedia concept detection is a centric research task in multimedia domain [56][57] [58]. A lot of work efforts have been put on feature representation [59][60], multi-model fusion [61][62], and classification model enhancement [63][64]. Recently, the idea of utilizing the inter-concept relationship for improving the concept detection accuracy has gained popularity. The current approaches generally fall into four categories, which are the semantic model vector approaches, the probabilistic graphic model approaches, the concept ontology-based approaches, and the concept correlation-based approaches. In this section, a literature review is provided. The limitations of these approaches are also discussed.

In the computer vision domain, some researchers recognized the help from the context and improved the object detection accuracy by taking the context information into consideration. In [65], the conditional random field-based (CRF-based) framework was proposed to reduce the ambiguity in objects' visual appearance using the context information in order to improve the object categorization accuracy. The researchers compared the results of not using any context information, using the context information from the Google Set, and using the context information derived from the training data sets from the MSRC and PASCAL databases. Experimental results indicated that the average categorization accuracy increased by more than 10% using the semantic context provided by Google Sets and by over 20% using the training contexts. Based on the idea of that work, some other frameworks were also presented. In [66], based on the observation made in [67] that the proper spatial relations among objects decreased the error rates in the recognition of the individual objects, the authors proposed the so-called CoLA (Co-occurrence, Location, and Appearance) model to include not only the context information but also the information of relative locations of the objects and used the CRF formulation to maximize the contextual constraints over the object labels. Specifically, the authors modeled four spatial pairwise relationships: "above," "below," "inside," and "around," and built a CRF model based on such information. The experimental results indicated that the additional information improved the object detection rate compared with the results in [65]. In [68], a nonparametric map of spatial priors was learned for each pair of objects. Torralba [69] combined the boosting algorithm with the CRF to detect an easy object first and then used the contextual information to help detect more difficult ones. Other similar approaches include [70] and [71]

The most popular approach in multimedia retrieval domain is the model vector approach, which was first proposed in [23]. The general idea is to reuse the output scores from the binary classification model for each concept and train another classification model using those scores for each concept. The assumption is that the newly trained

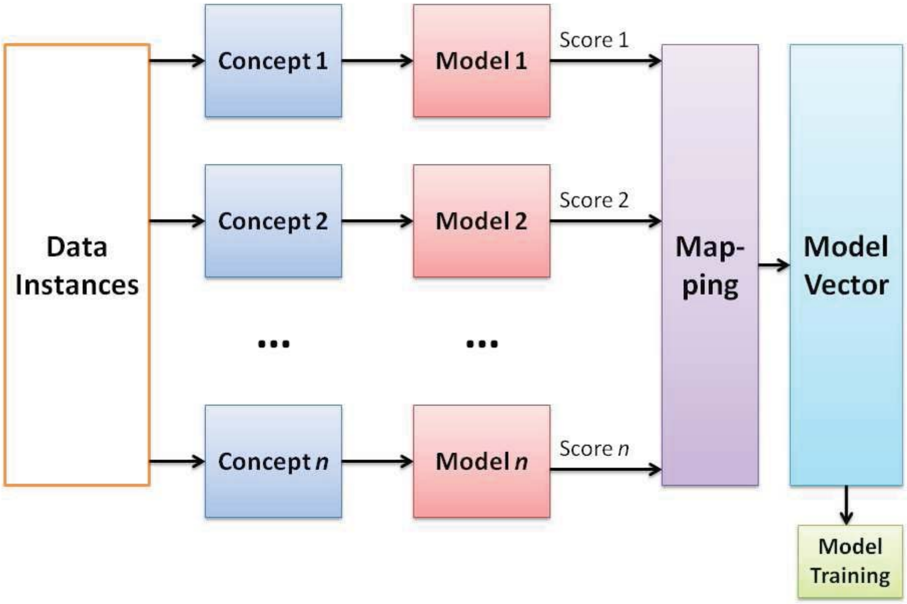


Figure 2.1: The framework of the model vector approach

classifier learns the correlation among labels by learning the structure of the output scores. In this way, the correlations among different concepts are modeled in the model vectors. Figure 2.1 shows a framework of such a model. n models output n scores, which are later mapped to a model vector.

For a data instance, such as a video clip or an image, n detection models are built for n high-level concepts. Each detection model outputs a confidence score for each concept. The n scores for the data instance are then mapped to a vector in χ dimensional space under certain criteria, such as minimizing the mean square error in the principal component analysis. Each vector in the χ dimensional space is named as a model vector and is utilized to construct another model, such as the k-nearest neighbor model or the support vector machine model for concept detection and retrieval. The experimental results shown in [23] indicated that the model vector-based approach improved the average precision of the four concepts “Building,” “Car,” “Sky,” and “Trees” of the TREVID video retrieval benchmark [54]. In [72], the researchers extended the work in [23] by training a support vector machine (SVM) model, which was named the context-based SVM model, using all the detection scores from all the other related primary detection models. Users were also involved in their approach to annotate some concepts in the testing videos, and the annotations were utilized to help the detection of other concepts. For a testing instance, assume $P_C(y_k = 1)$ is the output score from the context-based SVM model and $P_I(y_k = 1)$ is the output score from the primary detection model for concept k, the final detection score for concept k is given by the following equation. 2.1

$$P_F(y_k = 1) = w_k P_C(y_k = 1) + (1 - w_k) P_I(y_k = 1) \tag{2.1}$$

Here the w is trained using the cross-validation approach. In [73], the authors generated four SVM detection scores from four groups of features including bag of features, color histogram in the RGB color space, color moments in the YUV color space, and an edge direction histogram for each concept. In addition, they used 14 semantic concepts, including animals, city, desert, etc. in their framework. As a result, a 56-dimension vector named the prosemantic feature was used for building models for image retrieval. It could be seen that the prosemantic feature is in fact equivalent to the model vector. In a more recent work, the authors integrated the semantic model vectors with multiple low-level features into a composite framework. They argued that the semantic model vectors that comprised the semantic representation were able to mitigate the semantic gap. The experimental results indicated that the semantic model vector approach not only achieved the best performance but also was the most compact representation compared with all the other features in the experiment. The biggest issue about the model vector approach is the error propagation issue. Since each binary classification model is not perfect and could contain some classification errors, the output confidence scores from such models may contain noises. Accordingly, the model trained using the model vectors are not fully reliable. In summary, the errors of the binary models could propagate to the fusion step in the next round.

The second approach of modeling correlation is through probabilistic graphic or tree models. The general assumption is that multiple concepts are correlated with each other and the occurrence of a certain concept could affect that of the other related concepts. Such models usually take each high-level concept as the vertex and the correlation or the probabilistic relationships between two concepts as the edge. In this way, a graphic or tree model could be built to represent the inter-correlation of multiple concepts. For example, in [74], a Bayesian network that is a directed acyclic graph (DAG) was built to model the semantic contexts. In [75], the concept relationships were learned using the graph-based, semi-supervised learning for improving the concept annotation. Yan et al. [76] mined the relationship between the detection results of different concepts by a set of various probabilistic graphical models. In their paper, they compared directed graphic models such as the Bayesian network with undirected graphic models such as the Boltzmann machines, Markov random field, and the CRF. The authors performed experiments comparing two sorts of graphical models and showed that both models could improve the concept detection accuracy. In addition, the directed graphical model performed better. Jiang et al. [77] proposed the framework to combine the boosting approaches with the CRF to improve the detection results from independent detectors by taking into account the inter-correlation among concepts. The researchers further defined a criterion to predict which concepts might benefit from the contexture model. Choi et al. [78] learned a tree-structured graphical model to capture co-occurrence statistics of more than 100 object categories. Jiang et al. [79] identified the so-called domain change problem, in which the testing data are from a different domain of the training data. They proposed a domain adaptive semantic diffusion (DASD) framework to model the correlations among concepts using the Pearson product and built a weighted semantic graph for all the concepts. The graph was then applied in the semantic diffusion to recover the consistency of the annotation scores with respect to the concept affinities. This approach helps solve the domain change problem when a large amount of testing data is available. Another similar study was presented in [80].

In general, the graphical model aims at estimating the joint distribution of the probability of the co-occurrence of several concepts. Since the number of parameters increases exponentially with respect to the number of concepts, the biggest drawback of the graphic models remains their computational complexity, especially when the number of available concept detectors increases. In order to address this problem, some graphic models represent individual concept detection results as binary variables, such as in [81] [82]. Another study in this direction was presented in [83].

The third direction is the ontology-based frameworks, which model the inter-concept relationship utilizing the domain knowledge and combine the information from different classifiers. Wu et al. [84] connected 100 concept detectors to WordNet manually based on the similarity between concept definitions. They used the semantics to estimate the weights for the boosting linkage and for the confusion linkage, respectively. In their experiments, they showed that the ontology-based concept detector fusion algorithm performed better than the discriminative model fusion, which is in fact the model vector approach. In [85], the authors linked the multimedia semantics to the concept definitions provided by the large-scale concept ontology for multimedia. Their work might be the largest manual effort to connect concept detector definitions to an existing ontology so far. In [86], the authors defined two factors to either increase the concept detector probability when a hierarchical relation between detectors was present, such as trees and the vegetation, or decrease the probability when the ontology indicated that concepts cannot co-exist simultaneously, e.g., classroom and car. A further improvement of this idea was presented in [87], where the authors presented a comprehensive mathematical scheme for a video semantic ontology, which consists of concept lexicon, concept properties, and relations among concepts, and used the aforementioned ontology to refine concept detection of the SVM classifiers. Benmokhtar and Huet [24] incorporated an ontology model and integrated it with the neural network approach to assist concept detection. Wei et al. [88] constructed two semantic spaces, namely the Ontology-enriched Semantic Space (OOS) and the Ontology-enriched Orthogonal Semantic Space (OS2), to facilitate the selection and fusion of the concept detectors for video search . In [89], the authors combined the linguistic ontology from WordNet [90] and association rule mining to help video annotation and retrieval. One challenging problem is that the detectors are uncertain, while the reasoning using ontology relies on the symbolic facts. In order to address this issue, Elleuch et al. [91] used fuzzy logic to infer the correlations based on the ontology knowledge base.

The main disadvantage of these approaches lies in the dependence on the domain knowledge. In addition, as shown in [65], the performance of the frameworks based on prior knowledge may be worse than that of those based on the knowledge derived from the training data sets, which indicates that the robustness of the approaches using ontology and prior knowledge may not be good.

Although the aforementioned three directions cover most of the previous work in this area, there are some other approaches that utilize the concept co-occurrence relationship in the multimedia data set. In [92], the conditional probability was estimated based on the co-occurrence of each pair of concepts in the training data set and was used as the weight to fuse the detection scores output from each SVM model. In [93], the co-occurrence patterns were used to re-rank the initial results via the ranking functions. In order to generate the final score, a weighted combination of the original score and the re-ranked score was computed. They utilized a training framework to train the re-ranking algorithm on some concepts, and the re-ranking algorithm was applied to re-rank the remaining concepts. Liu et al. [94] mined association rules derived from the labeled training data to uncover hidden concept relations. The candidate rules were filtered using the support and confidence to re-compute individual concept detection scores. Experimental results showed that the proposed approaches improved the concept detection accuracy. In [95], the authors adopted a greedy algorithm, which partitions the training data into several hierarchical, concept-specific binary trees. The major problem of these approaches is that the performance of the framework is sensitive to selection of the concepts, and most of the work relies on human knowledge input for the related concepts selection.

2.3 Multi-Class Classification

In comparison to the multi-label classification problem, which could be solved by training a series of binary classifiers, the multi-class classification problem in which the labels are mutually exclusive is more challenging. The current approaches for the multi-class classification problem fall into two categories. The first category is the natural extension of the binary classification model; the second is to decompose the multi-class classification problem into a set of binary classification problems and fuse the classification results. They are introduced as follows.

2.3.1 Natural Extension of Binary Classification Model

Some classification algorithms for binary classification problem could be naturally extended to solve the multi-class classification problem. Some of the common algorithms include neural networks, decision trees, *k*-nearest neighbor, Naive Bayes, and SVM.

The multi-layer feedforward neural networks could be easily extended to solve the multi-class classification problem. Since the neurons of the output layer could be set by researchers, the output of all the neurons could be utilized to differentiate different classes. Generally, there are two major coding approaches: one-per-class coding and distributed output coding [96]. In the one-per-class coding framework, each output neuron is designed to identify a given class, while the instance is assigned to the class label, which corresponds to the neuron that gives the maximum output. In the distributed output coding framework, each class is assigned a unique binary code word from 0 to $2^N - 1$, where N is the number of output neurons. For a new instance, the output codeword is compared to the codewords for the K classes, and the nearest codeword is deemed to be the winning class.

The decision tree approach could naturally handle the multi-class classification problem. The idea of the decision tree is to infer a split of the training data based on the values of the available features in order to produce a good generalization. The split at each node is based on the feature that gives the maximum information gain. Each leaf node represents a class label. A testing instance will be classified by following the decision tree from the root node to the leaf node. Some common approaches of decision trees include classification and regression tree [97] and ID3/C4.5 [98]. Since the decision tree-based approach could suffer from the overfitting problem, a lot of pruning techniques were proposed to solve this issue [99].

The *k*-nearest neighbors approach [100] might be the most natural approach for performing classification. It is a classic non-parametric classification approach. The model keeps all the data in the training data set. When a testing instance comes, its class label is decided by its *k* nearest neighbors using some distance measurement such as the Euclidean distance or Hamming distance. One challenge of using this approach is to decide the parameter *k*, which could affect the performance to a large degree.

The Naive Bayes [101] approach tries to classify a testing instance based on the principle of Maximum A Posteriori (MAP). It could naturally solve the multi-class classification problem by assigning a label to an instance that could maximize the aposterior probability. In order to estimate the aposterior probability, the Naive Bayes approach makes the strong assumption that the features are independent given a certain class. Since this assumption does not hold in general, the performance of the Naive Bayes approach is usually poor.

Currently, the SVM is one of the most robust and successful classification algorithms [102] [103]. This approach is extended from the optimal margin classifier. The idea is to maximize the minimum distance from the separating hyper plane to the nearest example. The basic SVM supports only the binary classification, but several extensions [104][105][106][107] have been proposed to solve the multi-class classification case. In these extensions, additional parameters and constraints were added to the optimization problem to handle the separation of different classes. One issue about this approach is that the parameters increase exponentially with the increase of the number of classes, which increase the time complexity in the training phase.

2.3.2 Multi-Class Classification by Integrating Binary Classification Results

The other general direction in the research of multi-class classification problem is to decompose the problem into several binary classification tasks that could be solved efficiently using binary classifiers. After the classification results for each binary classifier are computed, the final decision is made by integrating these results together. The most widely used binary classifier is the SVM [103][102]. Specifically, there are four directions researchers usually follow to solve this problem.

The first approach is called the one-versus-all (OVA) approach, which decomposes the problem of classifying among K classes into K binary classification problems. Each of the binary problem discriminates a given class from the other $K-1$ classes. For this approach, the number of classes equals the number of classification models. The k-th classifier is trained with the positive examples belonging to class k and the negative examples belonging to the other $K-1$ classes. For a testing instance, the classifier that outputs the highest confidence value is considered to be the winner, and the corresponding label is assigned to that instance. Rafkin and Klautau [108] argued that this simple approach provided the accurate classification results as long as the underlying binary classifiers were well-tuned, regularized classifiers such as the SVM. One issue about this approach is that there is no bound on the generalization error [109].

The second approach is the all-versus-all (AVA) approach. In this approach, each class is compared to every other class. Therefore, a binary classifier is built to discriminate between each pair of classes. Therefore, this approach requires $\frac{K(K-1)}{2}$ binary classifiers. When a new example comes, the majority voting approach is used to decide which class wins. One problem of this approach is scalability, as the number of classifiers is proportional to the square of the number of classes. Another problem is that unless the individual classifiers are carefully regularized, the overall classifier system will tend to overfit [109].

The third approach is the error-correcting output coding (ECOC) [110]. It is a simple but powerful framework to deal with the multi-class categorization problem using the binary classifiers. It trains η binary classifiers to distinguish between the K different classes. Each class is given a codeword of length η according to a binary matrix ζ. Each row of ζ corresponds to a certain class. Each class is given a row of the matrix, and each column is used to train a distinct binary classifier. When testing an unseen example, the output codeword from the η classifiers is compared to the given K codewords, and the one with the minimum Hamming distance is considered the class label for that example. Diettrich and Bakiri [96] argued that the ECOC approach improved the generalization ability of the method above the OVA and AVA approaches.

Recently, the hierarchical classification approach has become increasingly popular. Yang et al. [111] proposed one approach to build the classification tree and utilized the relationship between the parent class and child class to help the classification. In that work, a set of binary classifiers formed a binary tree, and a testing instance was classified by traversing this tree from the root to the leaf. More recently, genetic programming was utilized, which brings relatively new solutions for this task [112]. However, there are two issues that remain to be solved. The first problem is how to generate the non-leaf class nodes so the relationships of the classes can be taken into account properly. The second problem is the error propagation issue that once the parent classifier makes an incorrect decision, there is no chance that the instance can be correctly classified. Consequently, there is lack of “cooperation” among different layers of classifiers. The genetic algorithm-based approach could provide cooperation to a certain degree.

2.4 Biomedical Image Classification

With the advancement in automatic imaging technologies, there is a rapid increase of biomedical imaging data sets in recent years, ranging from X-ray and CT for disease diagnosis to *in situ* hybridization (ISH) imaging for analyzing gene expression patterns. The number of bioimages is increasing on a scale comparable to that of the genomic revolution [113]. The huge biomedical image data sets present significant challenges for traditional analysis methods depending on manual annotation and human labeling. Therefore, utilizing computing technologies in automatic image processing and analysis has become a popular research topic. The need from the biological image research society motivates researchers in multimedia and data mining fields to develop solutions, which leads to a relatively new interdisciplinary research direction: bioimage informatics [114]. Many applications have been developed for automatic cell detection systems [115], bioimage segmentation systems [116][117], cell phenotype classification systems [118], etc.

Automatic categorization of the biological image is one of the important tasks in bioimage analysis. For example, it is a routine task in medical diagnosis to classify different kinds of tumors based on imaging information. In this section, the existing approaches for bioimage classification are reviewed. These approaches could be summarized into three categories. The first category is the generalized biological image classification system, which aims at building a comprehensive system that can be used in different classification tasks. In order to accommodate heterogeneous characteristics for different data sets, these applications usually encompass a large pool of features and carry the burden of huge computational complexity for feature extraction and feature selection. For example, Wndchrm [119], developed in the Laboratory of Genetics in

National Institute of Aging, contains 2,659 features from raw pixels and transforms of images, such as Gabor Filters and Chebyshev-Fourier features, and utilizes the nearest-neighbor algorithm to perform the classification. CellProfiler [120] and EnhancedCell-Classifier [121] apply the SVM with the radial basis function (RBF) kernel as the classifier and demonstrate relatively good performance, especially on cell images. For these two applications, the complex image processing pipelines can be mapped by linking different modules. Another well-known monolithic platform for biological image analysis is the ImageJ [122]. It has built a backbone of the image analysis workflow and accepts a number of plug-ins for different purposes. The Fiji [123] offers an extension of imageJ that is geared toward the needs of the bioimage community; it has special plug-ins for classification and many other applications. In terms of pattern recognition, KNIME [124] provides a user-friendly open source integration, processing, analysis, and exploration platform that is capable of processing large amounts of heterogeneous data sets. The platform also allows visual modeling of workflows.

The other category of research endeavor lies in developing specific applications for one or few data sets. There is a multitude of research on various biomedical imaging data such as histology images [125] [126] [127], endoscopic images [128], magnetic brain images [129], phase contrast CT images [130] [131], MRI images [132] [133], Thermogram analysis [134], etc. Just as in other machine learning and data mining research tasks, the main-stream researchers in this area focus on tackling two problems: first, how to extract the most significant features for a certain data set; second, how to improve the classification model and classification strategy.

In order to perform the classification tasks, many researchers focus on extracting significant features for a specific data set. Current commonly used features generally consist of texture descriptors, statistical distribution of pixel values, shape and edge features, and pixel transform-based features. In terms of the texture descriptors, the Scale-Invariant Feature Transform (SIFT) [135], histogram of Oriented Gradient (HOG) [21], and the Local Binary Pattern (LBP) [136] are the most widely used ones. In [137], the authors combined the SIFT features with density-based clustering to classify the biomedical images. Xu et al. [138] extracted HOG features from the anterior chamber angle region and performed the classification for the glaucoma subtype on the optical coherence tomography (OCT) images. Nanni et al. [139] utilized the LBP features for cell phenotype image classification and achieved relatively high accuracy. Utilizing the histogram to perform biological image classification was proposed in early years. Back in 2001, Soriano [140] proposed the approach of using the major color histograms for fluorescent image classification. Other similar studies using histograms include [141]. Compared with histogram-based features, shape and edge features are utilized more in biomedical image classification, as the objects in the image tends to have different shapes. In [142], the shape feature was utilized for X-ray image classification. In [143], shape features were combined with texture features to improve the X-ray image classification accuracy. In [144], the shape diffusion descriptor was developed for brain image classification. Other works that utilized shape and edge features include [145][146][147][148]. In terms of the transform-based features, the wavelet transform is the most common one. In [149], the authors extracted features using the adaptive discriminant wavelet packet transformation. In that paper, a full wavelet packet transform (FWPT) for each image in the texture sample was first acquired, and then the multi-resolution wavelet texture templates were used to calculate the pseudo-probability density estimates of a particular sub-band across all training samples of a particular class.

Buciu et al. [150] utilized Gabor wavelet-based features for mammographic image classification and got relatively high classification accuracy. Other works could be found in [151][152].

Some other studies perform segmentation before extracting features. For example, fractal geometry-based texture features were utilized to grade prostate carcinoma after human experts segmented the region of interest (ROI) in [153]. Another approach that applies the image segmentation technique before performing classification was presented in [154]. Their main idea was that by segmenting the ROI from the original image, the classification results could be improved. Their proposed approach achieved 100% accuracy by using a simple lazy learning algorithm on the testing data sets after segmenting the nuclei region from the original images and extracting features of chromatin patterns. A recent study showed that segmentation could help automated Gleason grading of the prostatic carcinoma tissue images [155].

Even though many of the features widely used in computer vision have been utilized in the biomedical image analysis domain and have shown good performance, domain experts such as the pathologists are reluctant to adopt the features for clinical diagnosis, as they lack clear biological interpretation. Therefore, the current research trend is to derive the features that not only provide good signatures for classification purpose but also have biological meanings and make sense for domain experts. Ali et al. [156] extracted the bio-inspired features (BIF) basically based on the computation of the local contrast in the cell. The authors claimed that this specially designed descriptor mimics the human visual system in categorizing the HEp-2 cells. In [125], the authors used the biological interpretable shape-based features for histological image classification. In that paper, the authors showed the shape features that are salient for each category of images and gave the biological interpretation. This type of features are quite useful in that specific classification task.

It is known that choosing an appropriate classification model for bioimage classification could affect the performance of the overall system. Many existing off-the-shelf pattern classification models have been applied in biomedical image categorization. Such algorithms include SVM [157][158][159], *k*-nearest neighbors [160], neural network [161][162], decision tree [163], logistic regression [164], etc. Recently, there are more research projects on the ensemble learning approaches that combine the results of multiple models. In [165], a hierarchical merging scheme was designed to perform the multi-class classification hierarchically. Kuncheva [166] proposed a random subspace ensemble method using the SVM for fMRI classification, and they showed that the ensemble classification strategy performed better than single classification models. Other similar studies include [167][141][168].

As biomedical images grow exponentially, classic analytical approaches face challenges. Therefore, some recent studies have been carried out on the large-scale biomedical image analysis. Said et al. [169] proposed a content-based image retrieval framework using the Hadoop distributed computing framework and the HDFS storage model. In [170], the authors presented three cases of applying the MapReduce model in biomedical image mining: feature extraction based on the three-dimensional wavelet analysis, parameter tuning for lung texture classification, and the content-based biomedical image indexing.

Chapter 3

Utilizing the AAN in Multimedia Concept Detection

The AAN framework, which essentially models correlations between different class labels, could be utilized in solving many problems. In this chapter, the application of the proposed AAN to assist the multimedia concept detection is introduced. The framework of using the proposed AAN to enhance the multimedia concept detection is introduced in Section 3.1. In Section 3.2, the enhancement of the framework, which incorporates two new models, is introduced. In Section 3.3, a negative correlation-based AAN is constructed and applied to enhance concept detection. Section 3.4 concludes this chapter and provides some insights based on experimental results.

3.1 Concept Detection Using the AAN

In this section, the general framework of the application of the AAN in concept detection is introduced. In Section 3.1.1, the high level description of the framework for using the AAN in multimedia concept detection and retrieval is given. Several important definitions and abbreviations used in this chapter are introduced in Section 3.1.2. The details of each component in the framework are introduced in Section 3.1.3. A detailed description of two data sets used in this chapter is given in Section 3.1.4. Experimental results and analysis are given in Section 3.1.5.

3.1.1 Overview of the Proposed Framework

The proposed framework is shown in Figure 3.1 (training stage) and Figure 3.2 (testing stage). As a specific application of the high-level framework proposed in Chapter 1, the overall framework consists of the “Multimedia Annotation,” “AAN Modeling,” and “Evaluation” components. In the training stage, the “Multimedia Annotation” component follows the architecture of the state-of-the-art multimedia concept mining system. Specifically, in a training data set, there are m instances (images or video shots) and n high-level concepts (such as “outdoor” and “sky”) to detect. The training instances are preprocessed and a set of features are extracted. Afterwards, n binary content-based classifiers (models) are trained for n concepts so that each model k $(1 \leq k \leq n)$ outputs m ranked scores for the k^{th} concept, represented by C_k in this chapter. This component is included here for the purpose of completeness, but is not the main focus of the overall framework.

In the “AAN Modeling” component, the detailed implementation of the “Node Link Modeling,” “Link Selection,” “Score Normalization,” and “Weight Computation” are shown. In the “Node Link Modeling,” each node represents one high-level concept, such as “sky,” “airplane,” etc.; each link represents the correlation between the two high-level concepts represented by the two nodes connected by that link. In order to prepare for link selection, all possible links are first generated using the label matrix, which is introduced in detail in Section 3.1.3. The “Link Selection” module uses the association rule mining (ARM) to discover the significant associations between concepts and to filter the insignificant ones. The training scores of different concepts are first input to the score normalization module so that scores from different concepts are normalized. The “Weight Computation” module receives the normalized training scores and computes

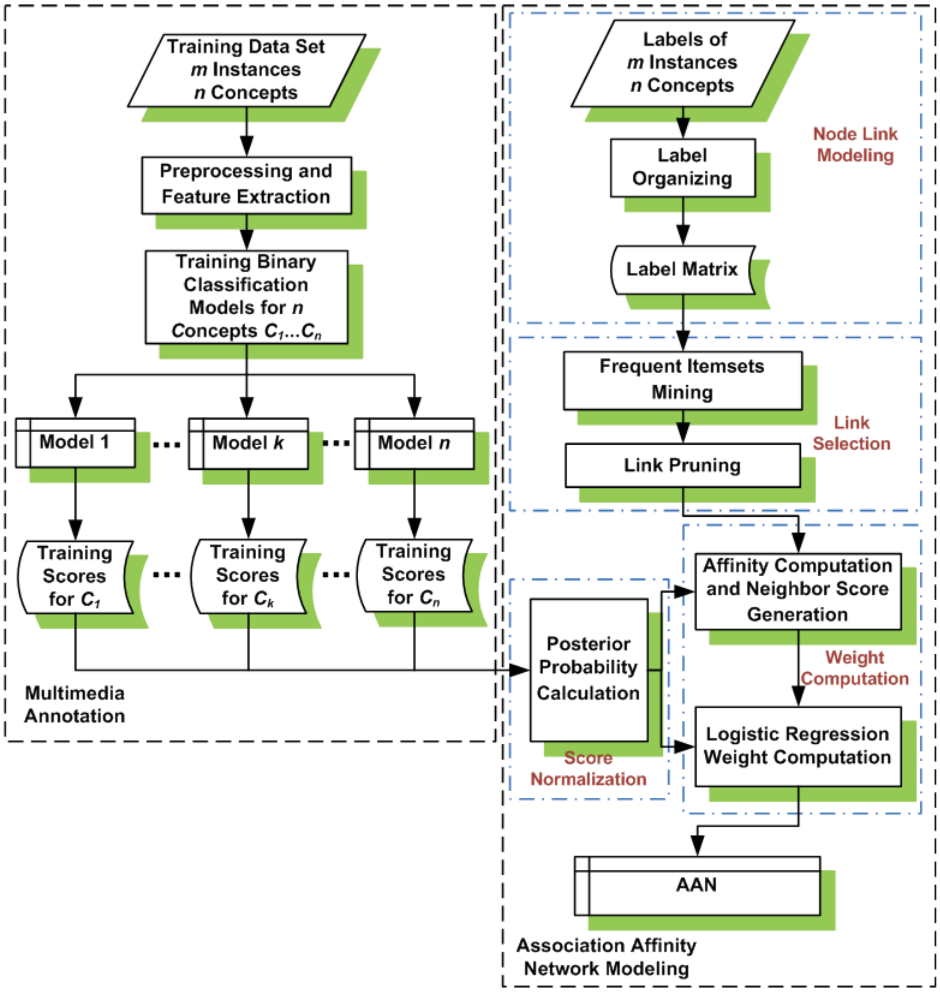

Training Data Set
m Instances
n Concepts
Preprocessing and Feature Extraction
Training Binary Classification Models for *n* Concepts $C_1...C_n$
Model 1
Model *k*
Model *n*
Training Scores for C_1
Training Scores for C_k
Training Scores for C_n
Multimedia Annotation
Labels of *m* Instances *n* Concepts
Label Organizing
Label Matrix
Node Link Modeling
Frequent Itemsets Mining
Link Pruning
Link Selection
Affinity Computation and Neighbor Score Generation
Weight Computation
Logistic Regression Weight Computation
Posterior Probability Calculation
Score Normalization
AAN
Association Affinity Network Modeling

Figure 3.1: The training stage of the proposed framework

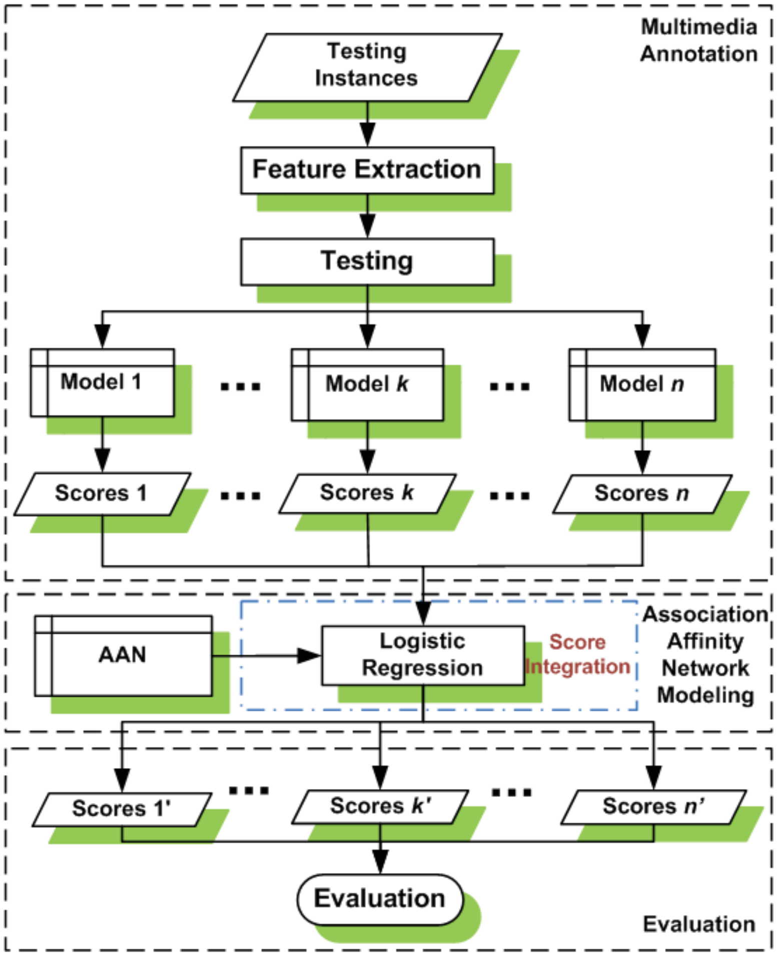


Figure 3.2: The testing stage of the proposed framework

the parameters and weights. The AAN is then constructed. It is important to point out that this component does not depend on models used in the "Multimedia Annotation" component.

In the testing stage, the same set of features as in the training stage is first extracted. Next, each testing instance receives one score from each content-based classifier in the "Multimedia Annotation" component. In the "AAN Modeling" component, a new score, which integrates information from related concepts using logistic regression, is generated as the final output. The output scores are input to the "Evaluation" component to evaluate the performance of the framework.

The detailed description of all modules are given in Section 3.1.2 and Section 3.1.3.

3.1.2 Definitions

Before introducing the important modules in the proposed framework, it is necessary to introduce several definitions used throughout this chapter.

Definition 1 (Data Instance and Label) *A* ***data instance*** *refers to a video shot, a keyframe, an image, or the features of one video shot/image/keyframe based on the context. The* ***label*** *in this paper is either 1 or 0, indicating whether the corresponding concept appears in the instance or not. If the label is 1, the data instance is named as a positive data instance for that concept, while if the label is 0, it is named as a negative data instance.*

Definition 2 (Correlation and Association) *The* ***correlation*** *and the* ***association*** *are used interchangeably and represent the relationship that the occurrence of one concept affects the probability of occurrence of another concept(s).*

Definition 3 (Positive Instance and Negative Instance) *A **positive instance** indicates that an instance contains a certain concept. A **negative instance** indicates that an instance does not contain a certain concept.*

Definition 4 (Concept-Class Pair) *A **concept-class pair** represents the label for the corresponding concept. In this paper, we denote the concept-class pair as C_k^ε, where k indicates the concept index and ε is the label. For example, C_5^1 indicates the event that a data instance is positive for concept 5; C_5^0 indicates the event that a data instance is negative for concept 5. When ε=1, the concept-class pair is a positive concept-class pair; when ε=0, the concept-class pair is a negative concept-class pair.*

Definition 5 (τ-itemset) *A **τ-itemset** is a set that consists of τ concept-class pairs. For example, $\{C_1^1, C_{100}^1\}$ is a 2-itemset. It is important to point out that a positive concept-class pair and a negative concept-class pair for the same concept should not appear in the same itemset, because one data instance is either a positive data instance or a negative data instance, but cannot be both at the same time.*

Definition 6 (Support) *The **support** value indicates the number of occurrences of the $\tau\text{-}itemset$ in the training data set. It is denoted as $sup(\tau\text{-}itemset)$.*

Definition 7 (Target Concept and Related Concept) *A **target concept** (TC) is defined as the concept to detect or the concept we are interested in. A **related concept** (RC) is a concept that is related to the TC. There could be more than one RC for one TC.*

Definition 8 (Target Score and Related Score) *A **target score** is a posterior probabilistic score of a data instance for a TC. A **related score** is a posterior probabilistic score of a data instance for a RC with respect to the TC.*

Definition 9 (Positive Rule and Negative Rule) *A **positive rule** indicates that the occurrence of one concept infers the occurrence of other concepts. A **negative rule** indicates that the occurrence of one concept infers non-occurrence of other concepts. For example, $C_{k1}^{1} \rightarrow C_{k2}^{1}$ is a positive rule. $C_{k1}^{1} \rightarrow C_{k2}^{0}$ is a negative rule. In this chapter, only the rules containing two concept-class pairs are considered.*

3.1.3 Detailed Description of Each Module

In this section, different modules in the AAN are introduced.

Node Link Modeling Module

The first step is to model the nodes and links in the application. Under the concept detection scenario, each node represents the concept and each link between two high-level concepts represents the correlation between them. The correlation could be mined from the concept-class pairs of all training instances. Therefore, all the labels of the training instances need to be organized in a more convenient form to be mined. In this work, the labels of all training data instances for all the concepts are reorganized into a label matrix. Table 3.1 shows an example of a label matrix. Assume there are m data instances and n concepts in the training data set; the rows in the matrix correspond to all the training data instances (1 to m) and the columns correspond to all the concepts (1 to n). The entry of the matrix is one concept-class pair, which indicates the instance in that row is positive or negative for the concept in that column. A case in point, Instance i is positive for C_1, negative for C_2, negative for C_k and positive for C_n, etc.

Link Selection

After the nodes and links are modeled, the next task is to select the significant links. The advantages of adding the link selection module are twofold: First, the time complexity

Table 3.1: A Label matrix

Instance	C_1	C_2	...	C_k	...	C_n
Instance 1	C_1^0	C_2^1	...	C_k^1	...	C_n^0
Instance 2	C_1^0	C_2^1	...	C_k^0	...	C_n^1
...	...	...	...	...	...	...
Instance i	C_1^1	C_2^0	...	C_k^0	...	C_n^1
...	...	...	...	...	...	...
Instance m	C_1^0	C_2^1	...	C_k^1	...	C_n^0

of computing the weights of the links is decreased. Second, as shown in the experiment part, filtering the casual and unreliable links could decrease the bias of models and improve the performance.

In general, significant links should have the following characteristics. First, the links need to be significant compared to other links. Second, the significance of the links depends on the *TC* to be detected. For example, in the IACC.1.A data set, the importance of the link between C_6 (animal) and C_{43} (dog) is different based on which concept to detect. In terms of C_6, the link between C_{43} and C_6 is very helpful because a correct positive label for “dog” ensures a correct label for “animal.” However, the link from C_6 to C_{43} is of less interest because an “animal” is not necessarily a “dog.” Taking this into consideration, an ARM-based association link generation algorithm is proposed. This link generation approach utilizes the Apriori [171] algorithm. The specific algorithm is given as follows.

ASSOCIATION LINK GENERATION

```
1 Search the label matrix to find all the 1-itemsets that contain positive concept-class pair.
2 Generate the candidate 2-itemsets by combining the 1-itemsets.
3 Filter candidate 2-itemsets to remove the 2-itemsets that have the support value less than one.
4 For a certain target concept $C_t$, select all 2-itemsets that contain $C_t$.
5 Generate the candidate positive rules for $C_t$ based on the selected 2-itemsets.
6 Select the significant rules from the candidate rules.
```

In sum, this algorithm consists of two processes. The first process is from Step 1 to Step 4. This process generates all candidate 2-itemsets that contain the target concept C_t. The second process is from Step 5 to Step 6. In this process, all candidate positive rules are generated for C_t with C_t as the target concept. Accordingly, the significantly related concepts are selected.

In order to select the most significant rules, two rule pruning modules are incorporated into the framework. Suppose that C_t is TC and C_r is RC, and one candidate rule is "$C_r^1 \rightarrow C_t^1$". The support ratio based rule pruning module addresses the significance of the rule from the perspective of TC and is modeled by the support ratio R_s defined in Equation (3.1).

$$R_s = \frac{sup(\{C_t^1, C_r^1\})}{sup(\{C_t^1\})}. \tag{3.1}$$

A threshold is used to select the rules with relatively high R_s values. Next, the interest ratio based rule pruning module is added to handle the significance from the RC's perspective. The interest ratio R_i is defined in Equation (3.2). Another threshold is used to select rules in a similar way. The thresholds are dynamically selected for each concept based on the training scores to maximize the average precision (AP) so that different preferences of the concept nodes are captured in the framework.

$$R_i = \frac{sup(\{C_t^1, C_r^1\})}{sup(\{C_t^1\}) \times sup(\{C_r^1\})}. \tag{3.2}$$

It should be pointed out that the selection procedure is in sequence. First, the support ratio based rule pruning module is applied. It focuses on the TC. Basically, it means how significant the co-occurrence of the RC and TC is compared with the occurrence of the TC alone. The higher the value is, the more valuable the RC is. Next, the interest ratio is used to make sure that the high co-occurrence of the RC and TC is not because the RC occurs frequently in the training instances.

In order to illustrate the idea clearly, a specific example is given here. Assuming that TC is C_3 (Airplane) and the support of 1-*itemset* $\{C_3^1\}$ is 216, there are four candidate 2-*itemsets* containing C_3, which are shown in Table 3.2. In the table, "$sup(2item)$" indicates the support of the corresponding 2-*itemset* and "$sup(rela.)$" indicates the support value of 1-*itemset* consisting of the related positive concept-class pair. If both thresholds are set to 50% and 50%, the support ratio based rule pruning module prunes Rule 3 and Rule 4. The interest ratio based rule pruning module prunes Rule 2. Rule 1 is retained and deemed to be the most significant rule. The parameters are dynamically selected for each concept based on the training scores to maximize the average precision (AP) so that different preferences of the concept nodes are captured in the framework.

Table 3.2: The candidate rules with $Concept_3$ (Airplane) at the conclusion side

ID	2-$itemset$	rule	$sup(2item)$	related concept	$sup(rela.)$	R_s	R_i
1	$\{C^1_3, C^1_4\}$	$C^1_4 \to C^1_3$	104	C_4:Airplane_Flying	104	0.481	0.004629
2	$\{C^1_3, C^1_{87}\}$	$C^1_{87} \to C^1_3$	113	C_{87}:Outdoor	19631	0.523	0.000027
3	$\{C^1_3, C^1_{108}\}$	$C^1_{108} \to C^1_3$	67	C_{108}:Sky	5053	0.310	0.000061
4	$\{C^1_3, C^1_{98}\}$	$C^1_{98} \to C^1_3$	9	C_{98}:Road	43303	0.0417	9.6×10^{-7}

The selected links are the significant connections among different nodes and form the core of an AAN.

Score Normalization

As the concept detectors could be distinct, the scores from different models need to be normalized to make them comparable to each other. In addition, it would be better if the credibility of the scores is also modeled. Instead of assigning a positive or negative label, modern multimedia retrieval systems usually output ranking scores in a descending order to indicate the relevance of the retrieved results to a user's query. Therefore, traditional F_1 score, precision, and recall are not suitable to be used here. Given these factors, a Bayesian posterior probability based score normalization approach is proposed.

Assuming for a data instance i, let the output score for concept k be S^i_k; $C_k = 1$ indicates that a data instance is positive for concept k; $C_k = 0$ indicates that a data instance is negative for concept k. The output score after conversion is S'^i_k. Equation (3.3) shows the formula to calculate S'^i_k.

$$S'^i_k = \frac{p(S = S^i_k | C_k = 1) p(C_k = 1)}{\sum_{q=0}^{1} p(S = S^i_k | C_k = q) p(C_k = q)} \tag{3.3}$$

Here, $p(C_k = 1)$ and $p(C_k = 0)$ indicate the prior probabilities that the data instance is positive or negative for concept k, respectively. $p(S = S^i_k | C_k = 1)$ and $p(S = S^i_k | C_k = 0)$ are the values of the two conditional probability density functions (pdf) $p(S | C_k = 1)$

and $p(S|C_k = 0)$ evaluated at S_k^i. The conditional probability density functions are estimated from the training data using the Parzen-Window approach [172]. The kernel function used in this study is the standard normal distribution $\mathscr{N}(0,1)$, and the Parzen-Window estimation of the pdf for one dimensional random variable χ is given by Equation (3.4).

$$p(\chi) = \frac{1}{\nu} \sum_{\mu=1}^{\nu} \frac{1}{\sqrt{2\pi}} \exp(-\frac{(\chi_\mu - \chi)^2}{2}). \tag{3.4}$$

Here, χ_μ are the training instances and ν is the total number of observed values. Using this equation, $p(S|C_k = 1)$ and $p(S|C_k = 0)$ could be estimated from the positive and negative training instances, respectively.

The output $S_k'^i$ is defined as the posterior probabilistic score in this study. As is shown in Equation (3.3), the proposed approach has two advantages. First, since the concept might prefer different models and the output scores from distinct models could fall into distinct ranges, this step unifies all scores and makes them comparable. This conversion simplifies the following mathematical computation in other modules. Second, the posterior probabilistic score incorporates the prior knowledge about the likelihood that a concept occurs in training data. This domain knowledge can sometimes be very helpful.

Weight Computation and Score Integration

After association links are selected, the skeleton of the AAN is completed. The next step is to integrate information from different sources. Specifically, two questions need to be answered. The first question is how to compute the weight between the TC node and RC nodes quantitatively. The second question is how to integrate the contribution of the target score and related scores. The link between two concept nodes is modeled as a bi-directional link in this study. This approach is adopted based on the observation of the asymmetrical relationship between semantic concepts in the real world. As far as the weight is concerned, inspired by the confidence measurement in ARM, we define the affinity of a link in Equation (3.5). Here, C_t is the TC, and $A_{r,t}$ indicates the affinity of the link from the RC C_r to C_t.

$$A_{r,t} = \frac{sup(\{C_t^1, C_r^1\})}{sup(\{C_r^1\})}. \tag{3.5}$$

Hence, for one TC node C_t, the affinities of all the links from the RC nodes could be computed. Assume they form a set E, for one data instance i, the integrated information from all RC nodes is summarized in the integrated related score $O_N(t,i)$, defined in Equation (3.6).

$$O_N(t,i) = \frac{\sum_{e \in E} A_{e,t} \cdot S_e'^i}{\sum_{e \in E} A_{e,t}}, \tag{3.6}$$

where $S_e'^i$ is computed for C_e and instance i using Equation (3.3).

While Equation (3.6) summarizes the related scores, the information of the target score is still missing. It is important to weigh the two types of scores properly. In this study, the logistic regression algorithm is applied to determine the necessary weights. Specifically, for a target concept C_t, let $\boldsymbol{x}^i$ be the column vector $[1 \; O_N(t,i) \; S_t'^i]^T$ and a parameter vector $\boldsymbol{\theta} = [\theta_0 \; \theta_1 \; \theta_2]^T$. The probability that an instance i is positive is given by the logistic function in Equation (3.7).

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}^i) = \frac{1}{1 + \exp(-h_{\boldsymbol{\theta}}(\boldsymbol{x}^i))}; \tag{3.7}$$

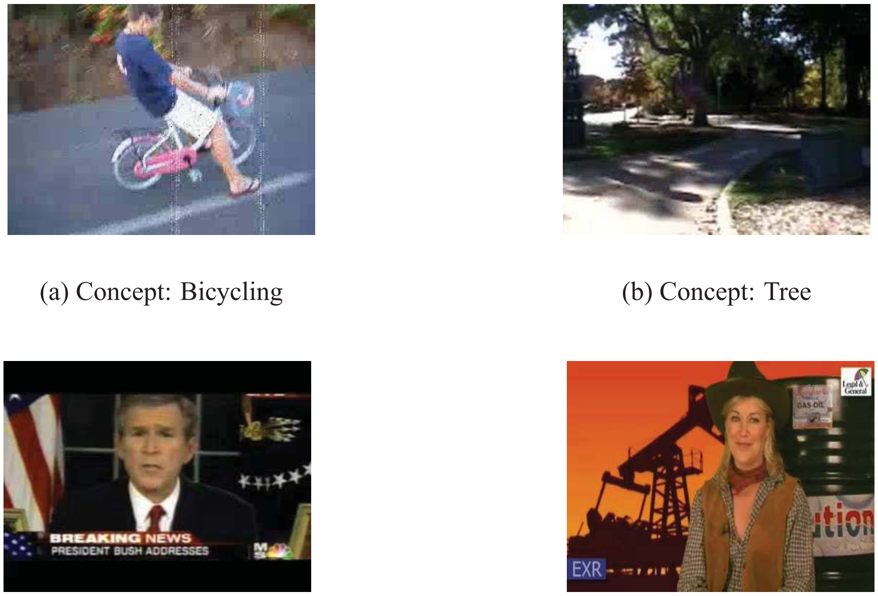
$$h_{\boldsymbol{\theta}}(\boldsymbol{x}^i) = \boldsymbol{\theta}^T \boldsymbol{x}^i. \tag{3.8}$$

Here, $\boldsymbol{\theta}$ could be learned by minimizing the cost function in Equation (3.9) using the gradient descent algorithm. The updating rule for $\theta_\kappa (0 \leq \kappa \leq 2)$ is shown in Equation (3.10), where δ is the learning rate determined by empirical study based on the training data set, m is the number of training instances, and y^i is either 1 or 0, indicating whether the instance is positive or negative. For a testing instance, the final score is computed using Equation (3.7) and the trained $\boldsymbol{\theta}$.

$$\begin{aligned} J(\boldsymbol{\theta}) &= -\frac{1}{m}[\sum_{i=1}^{m} y^i \log g_{\boldsymbol{\theta}}(\boldsymbol{x}^i) + (1-y^i)\log(1-g_{\boldsymbol{\theta}}(\boldsymbol{x}^i))]; && (3.9) \\ \boldsymbol{\theta}_\kappa &\leftarrow \boldsymbol{\theta}_\kappa - \delta \frac{\partial}{\partial \boldsymbol{\theta}_\kappa} J(\boldsymbol{\theta}). && (3.10) \end{aligned}$$

3.1.4 Data Sets

In this section, both the IACC.1.A and the IACC.1.B data sets used in this study from the TRECVID benchmark are introduced [54]. Each data set contains 200 hours of videos with durations between 10 seconds and 3.5 minutes. These videos are collected from the Internet and are diversified in terms of creator, content, style, production qualities and original collection devices. The videos are segmented into a number of shots and each shot is represented by a keyframe. The shot boundary and keyframes are also given. The labels are provided by collaborative annotation organized by National Institute of Standards and Technology(NIST). In this study, each keyframe is treated as a data instance. Figure 3.3 shows four sample keyframes with labeled concepts. Some basic statistics of these two data sets are given in Table 3.3. As shown in the table, one striking characteristic of the data sets is the data imbalance problem. On average, the positive instances are less than 1%. This poses great challenges in data retrieval in practice.



(a) Concept: Bicycling

(b) Concept: Tree

(c) Concept: Politics

(d) Concept: Face

Figure 3.3: Sample keyframes with annotated concepts in the TRECVID database

Table 3.3: Data statistics

Data Set	TRECVID Year	No. Concepts	No. Instances	Average Pos No.	Average P/N Ratio
IACC.1.A	2010	130	144774	865.42	0.0062
IACC.1.B	2011	346	137327	408.32	0.0030

From the data mining point of view, the concept detection problem is a multi-label classification problem. Figure 3.4 shows the distribution of number of concepts per instance has for the IACC.1.A data set. It can be seen that around 45.81% instances contain more than one concept. If the 40.26% instances that do not contain any concepts are eliminated, 76.69% instances contain more than one concepts. These data statistics indicate that rich information is contained in the inter-concept correlation and should be utilized. A similar distribution is observed in the IACC.1.B data set. The detailed explanations of the concepts of both data sets are introduced in [54].

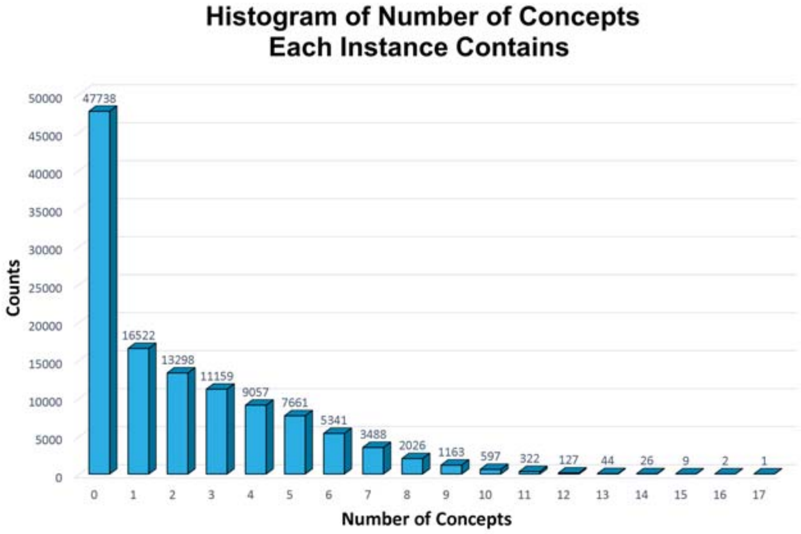


Figure 3.4: The histogram of the number of concepts in each data instance

In terms of scores, the detection scores of all concepts for the IACC.1.A data set were downloaded from the DVMM Lab of Columbia University [173]. The detection scores of all shots for the IACC.1.B data set were kindly provided by the Shinoda Lab at Deparment of Computer Science at Tokyo Institute of Technology [174].

3.1.5 Experiments and Results

We used the three-fold cross validation to evaluate the framework. In each fold, the data were divided into a training data set (2/3 of the total instances) and a testing data set (1/3 of the total instances). The training data set was further divided into (i) a developing set containing 70% of the instances in the training data and (ii) a cross-validation set containing 30% of the data instances in the training data. The model was trained on the developing set and several important parameters were tuned using the cross-validation set. After the optimized parameters were gained, the test data were input into the model trained on the training data with the optimized parameters, and the Mean Average Precision (MAP) value was calculated. Finally, the MAP values of the three folds were averaged to get the final evaluation of the model. Three rounds of three-fold cross validation were performed and the average MAP of the three rounds is reported here.

To better evaluate the performance of the proposed framework, the approaches in [92] and [24] were implemented. Compared to our proposed framework, the frameworks in [92] do not select the associations among concepts. The contribution of a RC to the TC is modeled using the conditional probability, which is inferred from the training labels. The parameter matrix used in the implementation was computed using the least square method. In [24], a pipeline was proposed from the video segmentation to final multi-model fusion. Since our work focuses on the model fusion part, only the ontology-based evidential algorithm in that work was implemented. The ontology was built based on the LSCOM-lite ontology framework [85]. The RCs were selected for each TC in the ontology-based graphic model. The similarity between a RC and a TC was computed based on the entropy values and was used as the weight between them.

In the IACC.1.A data set, there are 16,770 possible links for 130 concepts if one bi-directional link between two concepts is counted as two different links. After the association link generation, only about $1\% - 2\%$ of the links are retained and proved to be significant. For example, in one fold of the cross validation, the retained links totaled 291 and each target concept node has 2.24 links on average, which indicates the selected links are sparse. In addition, different concept nodes have distinct preferences for the number of selected links pointing to it. The concept node corresponding to the maximum number of links are C_{112} (Stadium) and C_{127} (Walking), which both have 12 links pointing to it. On the other hand, there are 42 concept nodes that do not have any

Table 3.4: The MAP values of 130 concepts in the IACC.1.A data set for different numbers of retrieved instances

Retrieved Instances	Top10	Top20	Top40	Top60	Top80	Top100	Top500	Top1000	Overall
Baseline	0.5218	0.4898	0.4481	0.4212	0.3999	0.3845	0.2807	0.2393	0.1382
Aytar	0.4600	0.4304	0.4075	0.3925	0.3806	0.3693	0.2754	0.2374	0.1363
BH	0.5316	0.5011	0.4570	0.4308	0.4082	0.3927	0.2853	0.2440	0.1416
Proposed	0.5491	0.5143	0.4709	0.4428	0.4211	0.4051	0.2944	0.2515	0.1452
Impr.R1	**5.23%**	**5.00%**	**5.09%**	**5.13%**	**5.30%**	**5.36%**	**4.88%**	**5.10%**	**5.07%**
Impr.R2	**19.37%**	**19.49%**	**15.56%**	**12.82%**	**10.64%**	**9.69%**	**6.90%**	**5.94%**	**6.53%**
Impr.R3	**3.29%**	**2.63%**	**3.04%**	**2.79%**	**3.16%**	**3.16%**	**3.19%**	**3.07%**	**2.54%**

link pointing to them at all, which indicates that they do not benefit from the network in the training set. Some concept nodes show consistent preferences for relatively large number of assistant links, such as C_{112} (Stadium), C_{102} (Science_Technology), and C_{103} (Scientists), while some concepts show preferences for no associations, such as C_5 (Anchorperson), C_7 (Asian_People), and C_{124} (US_Flags). Figure 3.5 shows the constructed AAN. Some hot spots, which have a relatively large number of connections (both out and in links), are marked using red and orange. Such nodes include C_{127} (Walking), C_{112} (Stadium), C_{89} (People_Marching), and C_{63} (Highway). The top 50 selected correlations are in Appendix A.

For the IACC.1.B data set, the same procedure is applied and 493 links are selected. The selected correlations are visualized in 3.6. The top 50 selected correlations are in Appendix B. In this section, only the IACC.1.A data set is used to evaluate the framework. In the next section, the proposed framework is further enhanced and both data sets are utilized for more detailed evaluations.

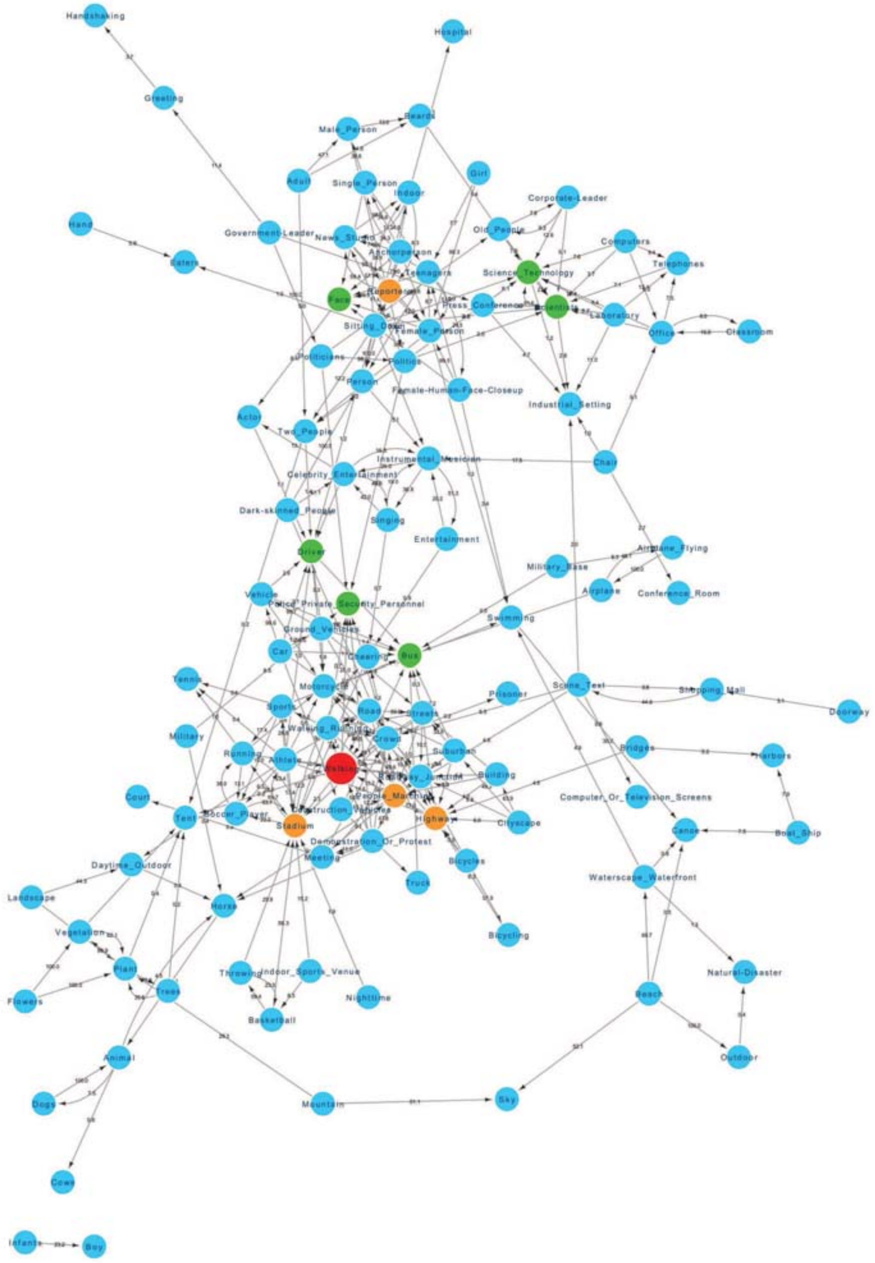


Figure 3.5: The AAN for positive correlations for the IACC.1.A data set

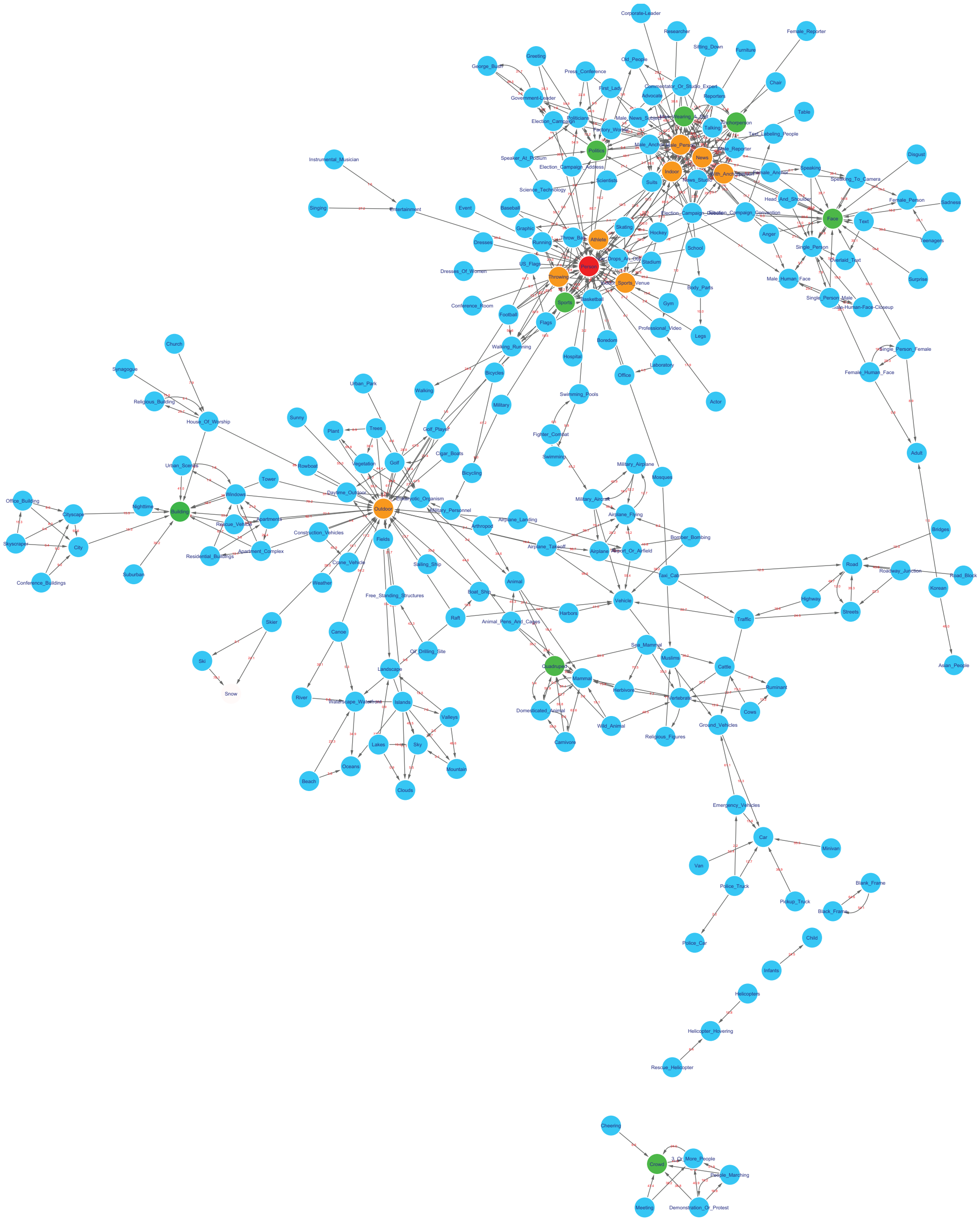


Figure 3.6: The AAN for positive correlations for the IACC.1.B data set

Table 3.5: The overall MAP of rare concepts in the IACC.1.A data set

Rare Concepts	Top5Rare	Top10Rare	Top15Rare	Top20Rare	Top25Rare	Top30Rare
Mean Positive Number	4.2	10.6	15.87	20.8	25.96	30.6
Baseline	0.0010	0.0256	0.0245	0.0238	0.0215	0.0228
Aytar	0.0003	0.0040	0.0053	0.0056	0.0067	0.0072
BH	0.0011	0.0286	0.0265	0.0254	0.0227	0.0239
Proposed	0.0012	0.0309	0.0289	0.0274	0.0245	0.0260
Impr.R1	**20.00%**	**20.70%**	**17.96%**	**15.13%**	**13.95%**	**14.04%**
Impr.R2	**300.00%**	**672.50%**	**445.28%**	**389.29%**	**265.67%**	**261.11%**
Impr.R3	**9.09%**	**8.04%**	**9.06%**	**7.87%**	**7.93%**	**8.79%**

Table 3.4 shows the MAP values for all the 130 concepts in the IACC.1.A data set for different numbers of retrieved instances. For example, “Top10” indicates the MAP value of the top 10 retrieved instances for all concepts, and the last column is the MAP value if all instances are retrieved. Each value in the table is the average of the three rounds of three-fold cross validation. “Baseline” corresponds to the MAP value of the raw scores, “Aytar” indicates the approach in [92], “BH” indicates the method in [24], and “Proposed” indicates the proposed framework. “Impr.R1,” “Impr.R2,” and “Impr.R3” represent the relative improvement rates of the proposed framework compared with the “Baseline,” “Aytar,” and “BH,” correspondingly. It could be seen that our proposed method outperforms “Baseline,” “Aytar,” and “BH” in terms of the MAP values. The experimental results show that the “Aytar” method is worse than the “Baseline.” One possible reason is that the noisy inputs from the insignificant links actually confuse the models. Interestingly, the performance of the “Aytar” method approaches baseline as the number of retrieved instances increases. This was also observed in [92] and one possible reason is that the scores of all models become smaller with the increase of the retrieved instances, so the effect of integrating scores turns trivial. The “BH” approach performs better than the “Baseline” but worse than the proposed framework. One possible reason is that their framework models the link between two concepts as a

symmetric edge, which does not match the asymmetrical relationship between semantic concepts in the real world.

In terms of rare concepts, the concepts are sorted in the ascending order according to the number of positive instances in the developing data set. Table 3.5 shows the MAP values for the top-ranked rare concepts. For example, "Top5Rare" means the concepts that have the 5 fewest positive instances in the developing data set. "Mean Positive Number" means the mean value of the number of positive instances of the rare concepts in the corresponding column. It can be observed that the total number of instances in the developing set is 96,516. Therefore, the top 5 rare concepts only have 4.2 positive instances for training. The following rows show the performance comparison in a similar way as in Table 3.4. Here, it shows that our proposed framework improves the MAP values for the rare concepts consistently when compared with the "Baseline," "Aytar," and "BH." The relative improvement of the MAP values for the rare concepts is higher than that of the whole 130 concepts. It indicates that the proposed framework is more effective for the rare concepts. The two possible reasons are as follows. First, compared with the classification model built on the target rare concept, the models of the RCs that have more training instances perform better, and the scores output from them are more reliable. By integrating such scores properly, the final output scores for rare concepts are improved. Second, generally speaking, the baseline MAP of the rare concepts is worse than that of the overall 130 concepts. Therefore, the relative improvement of MAP for rare concepts compared to that of all the concepts is larger. More future studies will be performed to shed light on the rationales in depth.

To further evaluate our proposed framework, a detailed analysis is performed to evaluate the contribution of each important module. The results are shown in Table 3.6.

Table 3.6: The contribution of each module

Component Change	Overall MAP	Performance Drop
No Change	0.1452	0
Remove Link Selection	0.1245	0.0207
Remove Post. Proba. Calc.	0.1411	0.0041
Remove Logistic Regression	0.1435	0.0017
Remove Affinity Computation	0.1424	0.0028

The first column indicates the changing of the module. “No Change” denotes the original framework. “Remove Link Selection” indicates removing the link selection module to keep all the links between concepts. “Post. Proba. Calc.” indicates using the raw scores directly without computing the posterior probabilities. “Remove Logistic Regression” indicates adding the self score and related score directly without using the logistic regression algorithm to compute the proper weights. “Remove Affinity Computation” indicates the affinities in Equation (3.5) are set to 1 for all the related concepts. The second column is MAP for all 130 concepts. The third column indicates the drop of the performance compared with the original framework. For example, if the “Link Selection” module is removed, the overall MAP drops to 0.1245 and the absolute value of drop is $0.1452 - 0.1245 = 0.0207$. It could be seen that the link selection module contributes the most to the performance gain. This observation again shows the importance of selecting the significant links. Besides the aforementioned module, the posterior probability calculation module, which converts different scores to the same scale, also makes relatively large contribution to the final performance gain.

3.2 Utilizing the Multi-Model Collaboration for Score Integration

3.2.1 Overview of the Enhancement of the Proposed Framework

In the aforementioned framework, the scores from the RC models are integrated into one score and that score is used to help improve the TC detection. There are two potential issues existing in that framework. First, the accuracy of the model, which determines its credibility, is not taken into consideration when RC is selected. Second, training labels for RCs are not fully utilized. It would be nice if the training labels, which contain valuable information given by the annotation of the users, could be reused. In order to address these issues, two new components are added to enhance the proposed framework.

In order to address the first issue, a Pearson product moment correlation based RC filtering approach is added to the link selection module. This approach takes into consideration of the accuracy of the related models and the harmony between the related scores and target labels. The details of this module are introduced in Section 3.2.2.

In order to address the second issue, the label and the score should play their roles simultaneously. In this thesis, inspired by the collaborative filtering approach, a multi-model collaboration framework that incorporates both the information from scores and labels is used to implement the weight computation module.

The enhanced framework is shown in Figure 3.7 and Figure 3.8. The enhanced modules are marked in red. “PPMC RC Filtering” indicates the Pearson product moment correlation based RC filtering approach. It could be seen that some modules such as the “Node Link Modeling” and “Score Normalization” are reused in this enhanced framework. So the specific details regarding them are already introduced in Section 3.1.3.

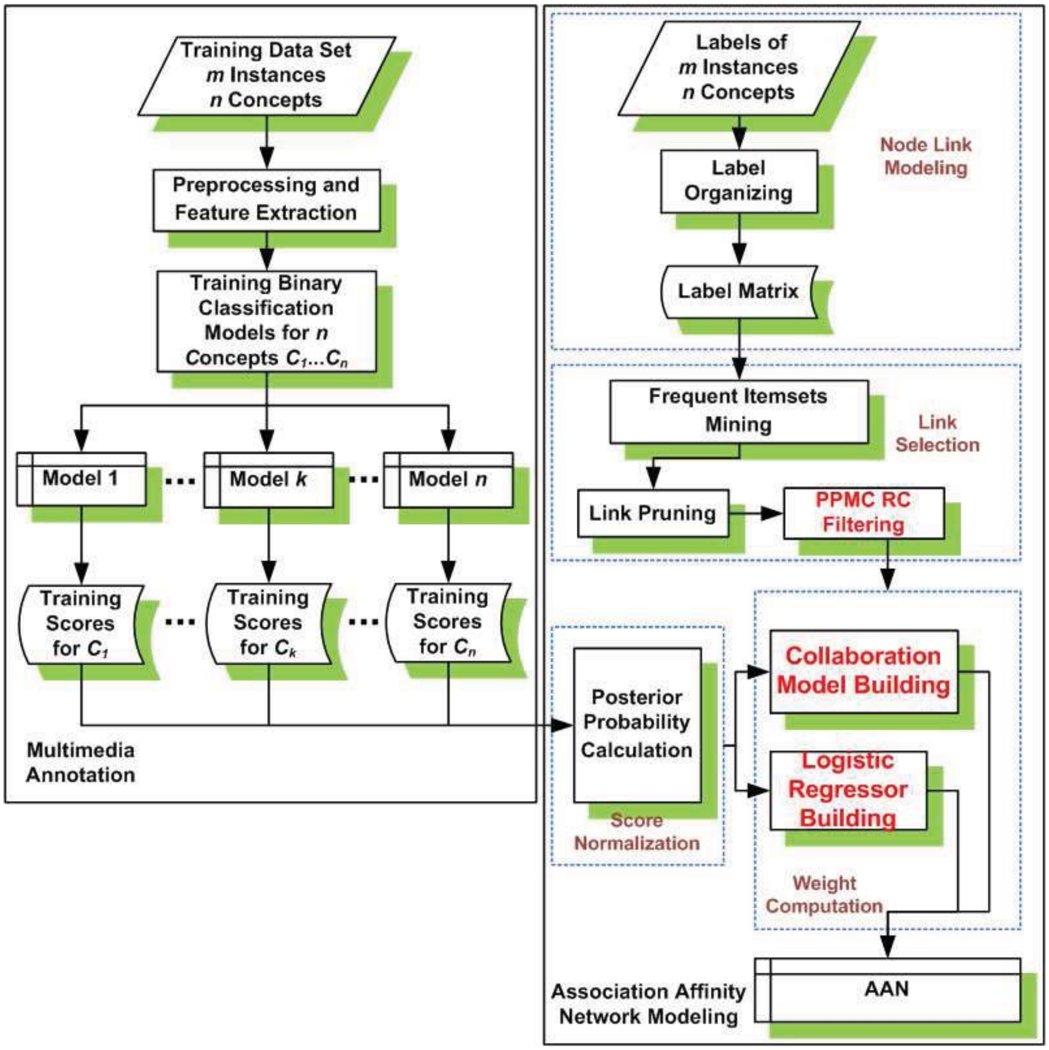

Training Data Set
m Instances
n Concepts
Preprocessing and Feature Extraction
Training Binary Classification Models for *n* Concepts C_1...C_n
Model 1
Model *k*
Model *n*
Training Scores for C_1
Training Scores for C_k
Training Scores for C_n
Multimedia Annotation
Labels of *m* Instances *n* Concepts
Label Organizing
Label Matrix
Node Link Modeling
Frequent Itemsets Mining
Link Selection
Link Pruning
PPMC RC Filtering
Posterior Probability Calculation
Score Normalization
Collaboration Model Building
Logistic Regressor Building
Weight Computation
AAN
Association Affinity Network Modeling

Figure 3.7: The multi-model collaboration training stage

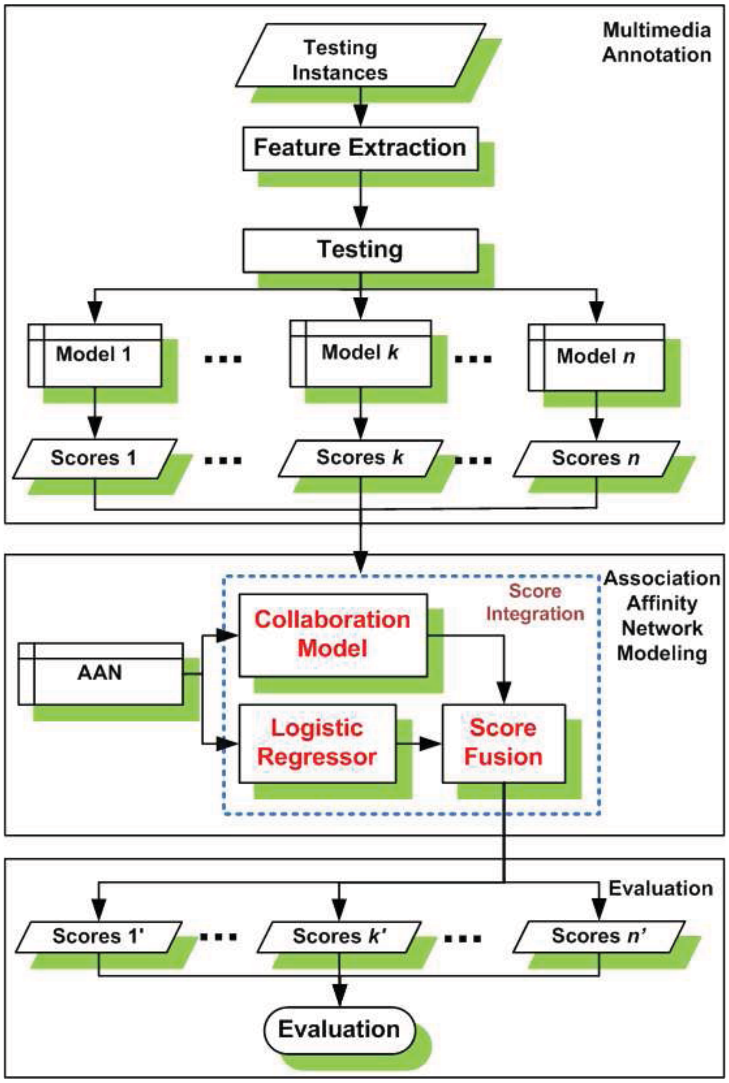


Figure 3.8: The multi-model collaboration testing stage

3.2.2 Detailed Description of Enhanced Modules

The specific improvements of all modules are given in this subsection.

Pearson Product Moment Correlation Based RC Filtering

The link selection in Section 3.1.3 does not consider the accuracy of the related models explicitly. This could cause some problem if the selected models do not perform well. Using the preceding example in Table 3.2, even though the selected concept "airplane_flying" is a strong RC, if the probabilistic scores output from the model built on the "airplane_flying" are relatively noisy, it could adversely affect the model of the concept "airplane." Therefore, the selected associations need to be filtered based on the quality of the concept model. Given this factor, the Pearson product moment correlation based RC selection is formulated and proposed in this work.

Formally, let y_t (0 or 1) represent the label for the TC, S_r' represent the posterior probabilistic score for the RC, MO_{TC} and MO_{RC} represent the binary classification models for TC and RC, and $Ac(MO)$ represent the accuracy of the model MO. The utility of the related model (UT), which measures how well the RC model could help the TC model, is defined in Equation (3.11).

$$UT = \int_0^1 HA(S_r', y_t) Ac(MO_{RC}) \, \mathrm{d}S_r' \tag{3.11}$$

Here, $HA(S_r', y_t)$ is the measurement of the harmony between S_r' and y_t. The assumptions here are (i) the higher the harmony between the probabilistic scores of the RC and the label of TC, the better the RC model is for the TC model; (ii) the higher the accuracy of RC model, the better RC model is for the TC model. Therefore, the RC models which have relatively high UT are preferred and selected.

The harmony could be modeled differently. In this dissertation, we use the Pearson product moment correlation coefficient, which captures the linear correlation between the two variables as the measurement of harmony. The Pearson product moment correlation coefficient for the two random variables $X1$ and $Y1$ is defined in Equation (3.12).

$$PCC = \frac{E[(X_1 - \overline{X_1})(Y_1 - \overline{Y_1})]}{\sigma_{X_1}\sigma_{Y_1}} \tag{3.12}$$

Here, $\overline{X_1}$ and $\overline{Y_1}$ are the mean values of X_1 and Y_1. σ_{X_1} and σ_{Y_1} are the standard deviations of X_1 and Y_1. When PCC equals 1, the two variables show perfect linear positive correlations. On the other hand, when PCC equals -1, the two variables are negatively correlated perfectly. Since we utilize the positive-positive correlations between the TC and RC, the Pearson product moment correlation coefficient serves as a good mathematical model for evaluating the harmony. The point-biserial correlation coefficients are also tested and the selection results are similar.Equation (3.11) gives a guideline for measuring the utility of RC. However, it is difficult to be evaluated directly. In order to approximate the utility function in the discrete case, the integral is approximated using summation. In addition, there are m instances in the training data set. The new value UT', which is the utility, is computed in Equation 3.13.

$$UT' = \sum_{i=1}^{m} \Gamma(S_{RC}^{\prime i}) \tag{3.13}$$

THE ALGORITHM TO COMPUTE $\Gamma(S_{RC}'^{i})$

```
1   Find the $\Theta$-nearest neighbors for $S_{RC}'^{i}$, and they are
    $S_{RC}'^{1}, \ldots S_{RC}'^{\Theta}$.
2   Find the instances $I_1, I_2, \ldots, I_\Theta$.
3   Find the corresponding labels of TC $y_{TC}'^{1}, \ldots y_{TC}'^{\Theta}$ for the
    $\Theta$ instances.
4   Evaluate the Average Precision of $MO_{RC}$
    using $S_{RC}'^{1}, \ldots S_{RC}'^{\Theta}$ and $y_{RC}^{1}, \ldots y_{RC}^{\Theta}$.
5   Evaluate the Pearson product moment correlation coefficient using
    $S_{RC}'^{1}, \ldots S_{RC}'^{\Theta}$ and $y_{TC}'^{1}, \ldots y_{TC}'^{\Theta}$
6   Calculate $\Gamma(S_{RC}'^{i})$, which is the product of results from 4 and 5.
```

The $\Gamma(S_{RC}'^{i})$ value is computed using the aforementioned algorithm. Here, the $Ac(MO_{RC})$ is estimated using Mean Average Precision. For the special case that all $y_{TC}^{1}, \ldots y_{TC}^{\Theta}$ are 0 or 1, the UT' is set to zero for that instance. The final results of UT' are used to filter the selected RC models to keep the most helpful models. A threshold is adjusted based on each concept in the cross-validation process.

Multi-Model Collaboration

In this section, the multi-model collaboration framework is introduced. This framework performs the functions of the "Weight Computation" module and the "Score Integration" module. In general, the main problem of the overall framework is to solve the re-ranking problem based on the scores from different models so that the overall classification accuracy is improved. This procedure could be formalized as the following mathematical problem. Assume for one target concept C_t and the i-th data instance, $S_t'^{i}$

is the posterior probabilistic score computed using Equation (3.3), Ψ is the set of IDs for the positive data instances for concept *k* and *m* represents the total number of training instances, and $\{S_t^{\prime j}\}$ $(j \neq i)$ denotes the set of posterior probabilistic scores of the $(m-1)$ data instances in the training data set by removing the data instance *i*. If we sort all the scores of concept *t* in descending order, the ranking number for data instance *i* is a function of $S_t^{\prime i}$ and $\{S_t^{\prime j}\}$, which could be represented as $\Lambda(S_t^{\prime i}, \{S_t^{\prime j}\})$. On the other hand, the scores for all RCs of data instance *i* for concept *j* form a set represented as $\{S_r^{\prime i}\}$ $(r \neq t)$, then $Q_{r,t}^i$ (i.e., the score after re-ranking for data instance *i*) could be expressed using a function *f* in Equation (3.14).

$$Q_{r,t}^i = f(S_t^{\prime i}, \{S_r^{\prime i}\}). \tag{3.14}$$

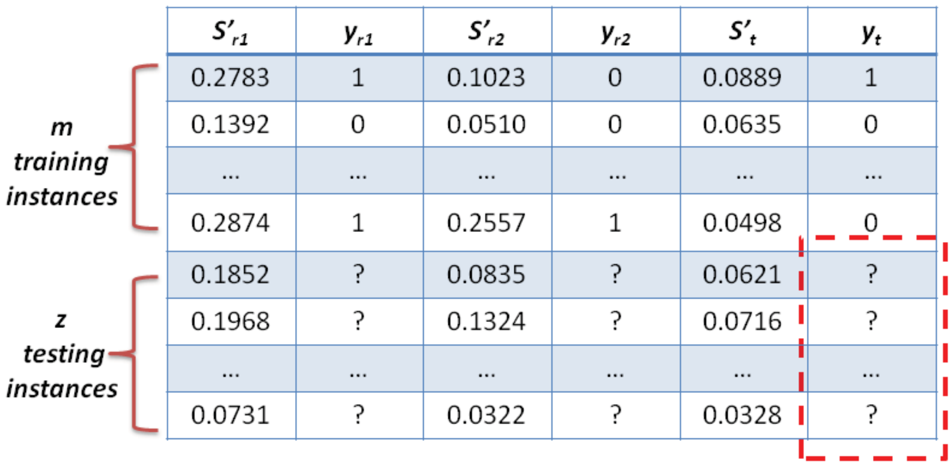
Similarly, $\{Q_{r,t}^j\})$ $(j \neq i)$ denotes the set of scores after re-ranking of the $(m-1)$ data instances in the training data set by removing the *i*-th data instance. If all the re-ranking scores are sorted in descending order, the new ranking number of instance *i* is given by $\Lambda(Q_{r,t}^i, \{Q_{r,t}^j\})$. Therefore, the re-ranking process is converted to solve the optimization problem in Equation (3.14) and Equation (3.15). It is noticed that the minimization is done with respect to f. This minimization problem is non-linear and very difficult to be solved directly. Therefore, we propose the logistic regression based model and the collaboration model to get the approximation results.

$$f = \underset{f}{\operatorname{argmin}} \sum_{i \in \Psi} \Lambda(Q_{r,t}^i, \{Q_{r,t}^j\}) - \Lambda(S_t^{\prime i}, \{S_t^{\prime j}\}). \tag{3.15}$$

The logistic regression model utilizes the sigmoid function to represent the logarithm likelihood and to maximize logarithm likelihoods of all the training instances. In our proposed framework, since the positive label is 1 and the negative label is 0, the re-ranking procedure represented by Equation (3.15) could be viewed as maximizing posterior probabilities based on the scores of RCs and TC.

Formally, for the *i*-th data instance and the target concept C_t, let $CA = |\{S_r'^i\}|$ be the cardinality of $\{S_r'^i\}$, r be the concept ID for one RC, $S_t'^i$ be the score of that target concept, and $[S_r'^i]$ be the vector of which each element is a member of $\{S_r'^i\}$. The concatenated vector $\boldsymbol{x}^i = [1, S_t'^i, [S_r'^i]]^T$ is the column vector of dimension $(CA+2)$ by 1, and the corresponding parameter is $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{(CA+2)}]^T$. The final score is given by Equation (3.7). The parameters can be estimated by minimizing Equation (3.9). An important improvement in this enhanced framework is that the $\boldsymbol{x}^i$ consists of all the related scores and target score. In Section 3.1, the $\boldsymbol{x}^i$ consists of three fixed elements, which are the target score, the integrated related score, and the bias unit.

While the logistic regressor captures the relationship among training scores of related concepts and target concepts, the information of training labels is missing. In this dissertation, the idea of collaborative filtering is adopted to solve this problem. Collaborative filtering (CF) [175] is an algorithm used in the recommendation systems. One of its applications is to predict users' ratings for an item, such as a movie, according to (i) the user's previous ratings for other items and (ii) the ratings for the same item from other users. The fundamental assumption of CF is that if two users of the system have similar behaviors, then most likely they will act similarly on the other items. This collaboration inspires us to study whether it could be used as a fusion strategy for integrating information from multiple related models to help the target concept detection. Using this approach, both training scores and training labels could be fully utilized in the framework.



	S'_{r1}	y_{r1}	S'_{r2}	y_{r2}	S'_{t}	y_{t}
m training instances	0.2783	1	0.1023	0	0.0889	1
	0.1392	0	0.0510	0	0.0635	0
	…	…	…	…	…	…
	0.2874	1	0.2557	1	0.0498	0
z testing instances	0.1852	?	0.0835	?	0.0621	?
	0.1968	?	0.1324	?	0.0716	?
	…	…	…	…	…	…
	0.0731	?	0.0322	?	0.0328	?

Figure 3.9: A sample collaboration matrix

The general steps are as follows. First, the labels and the scores of TC and RCs are put into a collaboration matrix. Figure 3.9 shows an example of a collaboration matrix. In this example, for the target concept C_t, there are two related concepts, which are C_{r1} and C_{r2}. In addition, the total number of training instances is m and the total number of testing instances is z. y_t, y_{r1}, and y_{r2} are the columns of labels for C_t, C_{r1}, and C_{r2}. S'_t, S'_{r1}, and S'_{r2} are the columns of posterior probabilistic scores for C_t, C_{r1}, and C_{r2}. Since testing labels for all the testing data instances are the targets we want to predict, they are represented as the question marks in the figure.

From the collaborative filtering point of view, both the labels and the scores in the collaboration matrix could be viewed as ratings given by different “users.” The ground truth labels y_t, y_{r1}, and y_{r2} are given by the “users” who give high-quality ratings to the training instances but do not give any ratings to the testing instances. Although the scores S'_t, S'_{r1}, and S'_{r2} are given by the “users” who assign ratings to both the training and testing instances, the quality of the ratings might be low. Therefore, the target of our algorithm is to predict the ratings given by the “user” corresponding to y_t for the testing instances, which are indicated using the red rectangle in Figure 3.9. In order to solve this problem, the regression-based collaborative filtering approach is adopted. This approach learns a new set of "features" based on the ratings given by the "users" and uses these "features" to predict the ratings given by a certain user. Specifically, let $\boldsymbol{G}$ stand for the collaboration matrix and $G(u,v)$ be the element at the row u and column v in G, where $1 \leq u \leq U$, $1 \leq v \leq V$; U and V are the total number of data instances ($U = m + z$) and the total number of models in the collaboration matrix. $\Upsilon(u,v) = 1$ indicates that $G(u,v)$ is known and $\Upsilon(u,v) = 0$ indicates the $G(u,v)$ is unknown. $\boldsymbol{\omega}$ is the "feature matrix," which is U by o. $\boldsymbol{\phi}$ is the "model coefficient matrix," which is o by V. o is a parameter that is determined in the cross-validation process. Therefore, a new matrix $\boldsymbol{G'}$ could be computed as in Equation (3.16). This matrix represents the information integrated from the two related matrices.

$$\boldsymbol{G'} = \boldsymbol{\omega}\boldsymbol{\phi}. \tag{3.16}$$

Specifically, $(\boldsymbol{\omega}^{(u)})^T$ is the u-th row of matrix $\boldsymbol{\omega}$ and $\boldsymbol{\phi}^{(v)}$ is the v-th column of matrix $\boldsymbol{\phi}$. One element $G'(u,v)$ could be computed as follows.

$$G'(u,v) = (\boldsymbol{\omega}^{(u)})^T \boldsymbol{\phi}^{(v)}. \tag{3.17}$$

Since we want to use $\boldsymbol{G'}$ to approximate $\boldsymbol{G}$, the problem is converted to solving the optimization problem to minimize the mean square error between matrix $\boldsymbol{G}$ and matrix $\boldsymbol{G'}$. Accordingly, a cost function defined in Equation (3.18), where $\boldsymbol{\phi}_l^{(v)}$ indicates the l-th element of vector $\boldsymbol{\phi}^{(v)}$, $\boldsymbol{\omega}_l^{(u)}$ indicates the l-th element of vector $\boldsymbol{\omega}^{(u)}$, $1 \leq l \leq o$, and λ_1 stands for the regularization parameter to address the overfitting issue. The gradient descent algorithm is used here and the updating rule for each parameter is given in Equation (3.19) and Equation (3.20). Here, ψ is the learning rate.

After the two matrices are computed, the predictions for the target labels could be computed by multiplying the corresponding row in matrix $\boldsymbol{\omega}$ and the corresponding column in matrix $\boldsymbol{\phi}$.

For a testing data instance, after the prediction scores from the logistic regressor and the collaboration model are computed, another question is how to combine the two scores. We carried out an empirical study and the logistic regression model was chosen to combine the two scores. The results of empirical study are given in Table 3.10.

$$\begin{aligned}
&J'(\boldsymbol{\omega}^{(1)},...,\boldsymbol{\omega}^{(U)},\boldsymbol{\phi}^{(1)},...,\boldsymbol{\phi}^{(V)}) = \\
&\frac{1}{2}\sum_{(u,v)}((\boldsymbol{\omega}^{(u)})^T\boldsymbol{\phi}^{(v)} - G(u,v))^2 \\
&+\frac{\lambda_1}{2}\sum_{u=1}^{U}\sum_{l=1}^{o}(\boldsymbol{\omega}_l^{(u)})^2 + \frac{\lambda_1}{2}\sum_{v=1}^{V}\sum_{l=1}^{o}(\boldsymbol{\phi}_l^{(v)})^2 \qquad (3.18)\\
&\text{ under the condition: } \Upsilon(u,v) = 1.
\end{aligned}$$

$$\boldsymbol{\omega}_l^{(u)} \leftarrow \boldsymbol{\omega}_l^{(u)} - \psi\frac{\partial J'}{\partial\boldsymbol{\omega}_l^{(u)}} \qquad (3.19)$$

$$\frac{\partial J'}{\partial\boldsymbol{\omega}_l^{(u)}} = \sum_v((\boldsymbol{\omega}^{(u)})^T\boldsymbol{\phi}^{(v)} - G(u,v))\boldsymbol{\phi}_l^{(v)} + \lambda_1\boldsymbol{\omega}_l^{(u)}$$

under the condition: $\Upsilon(u,v) = 1$.

$$\boldsymbol{\phi}_l^{(v)} \leftarrow \boldsymbol{\phi}_l^{(v)} - \psi\frac{\partial J'}{\partial\boldsymbol{\phi}_l^{(v)}} \qquad (3.20)$$

$$\frac{\partial J'}{\partial\boldsymbol{\phi}_l^{(v)}} = \sum_u((\boldsymbol{\omega}^{(u)})^T\boldsymbol{\phi}^{(v)} - G(u,v))\boldsymbol{\omega}_l^{(u)} + \lambda_1\boldsymbol{\phi}_l^{(v)}$$

under the condition: $\Upsilon(u,v) = 1$.

Table 3.7: The MAP values of the IACC.1.A data set

Retrieved Instances	Top10	Top20	Top40	Top60	Top80	Top100	Top500	Top1000	Overall
Baseline	0.5218	0.4898	0.4481	0.4212	0.3999	0.3845	0.2807	0.2393	0.1382
Aytar	0.4600	0.4304	0.4075	0.3925	0.3806	0.3693	0.2754	0.2374	0.1363
BH	0.5316	0.5011	0.4570	0.4308	0.4082	0.3927	0.2853	0.2440	0.1416
DASD	0.4637	0.4561	0.4240	0.4063	0.3903	0.3743	0.2886	0.2491	0.1397
Previous	0.5491	0.5143	0.4709	0.4428	0.4211	0.4051	0.2944	0.2515	0.1452
Proposed	0.5738	0.5488	0.5015	0.4788	0.4472	0.4304	0.3191	0.2779	0.1548
Impr.R1	**9.93%**	**12.05%**	**11.92%**	**13.68%**	**11.83%**	**11.94%**	**13.68%**	**16.13%**	**12.01%**
Impr.R2	**24.70%**	**27.51%**	**23.07%**	**21.99%**	**17.50%**	**16.54%**	**15.87%**	**17.06%**	**13.57%**
Impr.R3	**7.90%**	**9.52%**	**9.74%**	**11.14%**	**9.55%**	**9.60%**	**11.85%**	**13.89%**	**9.32%**
Impr.R4	**23.70%**	**20.32%**	**18.28%**	**17.84%**	**14.58%**	**14.99%**	**10.57%**	**11.56%**	**10.81%**
Impr.R5	**4.46%**	**6.71%**	**6.50%**	**8.13%**	**6.20%**	**6.25%**	**8.39%**	**10.50%**	**6.61%**

Table 3.8: The MAP values of the IACC.1.B data set

Retrieved Instances	Top10	Top20	Top40	Top60	Top80	Top100	Top500	Top1000	Overall
Baseline	0.6470	0.5926	0.5285	0.4749	0.4418	0.4206	0.4184	0.4311	0.4449
Aytar	0.5333	0.4743	0.4559	0.4440	0.3794	0.3591	0.3410	0.3941	0.3978
BH	0.6799	0.6236	0.5488	0.4864	0.4597	0.4286	0.4225	0.4455	0.4491
DASD	0.6376	0.5852	0.5181	0.4547	0.4170	0.3959	0.3881	0.4008	0.4149
Previous	0.7516	0.6861	0.6155	0.5369	0.4912	0.4584	0.4531	0.4722	0.4759
Proposed	0.7902	0.7219	0.6424	0.5655	0.5228	0.4872	0.4839	0.5043	0.5060
Impr.R1	**22.13%**	**21.83%**	**21.55%**	**19.07%**	**18.33%**	**15.82%**	**15.65%**	**16.99%**	**13.74%**
Impr.R2	**26.84%**	**27.27%**	**24.98%**	**20.40%**	**21.35%**	**18.53%**	**19.20%**	**18.58%**	**15.36%**
Impr.R3	**16.22%**	**15.76%**	**17.06%**	**16.27%**	**13.72%**	**13.68%**	**14.55%**	**13.21%**	**12.68%**
Impr.R4	**23.94%**	**23.35%**	**23.99%**	**24.37%**	**25.38%**	**23.06%**	**24.68%**	**25.83%**	**21.95%**
Impr.R5	**5.14%**	**5.22%**	**4.37%**	**5.33%**	**6.43%**	**6.28%**	**6.80%**	**6.80%**	**6.32%**

3.2.3 Experiments and Results

Both the IACC.1.A and the IACC.1.B data sets are used to evaluate the enhanced framework. Besides the two comparison approaches introduced in Section 3.1.5, the DASD approach proposed in [25] was added as a comparison approach. Compared with our proposed framework, the frameworks in [92] and [25] do not select the associations among concepts. The correlations between concepts were modeled as symmetric links with the weight gained from the label matrix only, and the authors used the graphic model to address the domain change problem.

Table 3.7 and Table 3.8 show the average of the MAP values for the three rounds of three-fold cross-validation results for the two data sets, respectively. Table 3.7 shows the results for the 130 concepts selected in the TRECVID 2010 semantic indexing task and Table 3.8 shows the results for the 50 concepts selected in the TRECVID 2011 semantic indexing task. The columns indicate the number of retrieved data instances for evaluations. “Baseline” indicates using the raw scores without applying any re-ranking framework; “Aytar,” “DASD,” and “BH” are the frameworks in [92], [25], and [24]; “Previous” indicates the framework proposed in Section 3.1; and “Proposed” is our proposed framework in this section with the enhanced modules. The rows of “Impr.R1,” “Impr.R2,” “Impr.R3,” “Impr.R4,” and “Impr.R5” correspond to the relative improvements of our proposed framework with respect to the performance of the “Baseline,” “Aytar,” “DASD,” “BH,” and “Previous” frameworks. According to TRECVID official evaluation results for scores provided by Shinoda Lab [174][176], the mean inferred average precision is 0.1875079. The overall MAP value reported for “Baseline” in Table 3.8 is different because experimental settings are different. From both tables, it can be seen that our proposed framework outperforms the “Aytar” framework, the “DASD” framework, the “BH” framework, and the “Previous” framework consistently for different numbers of retrieved results. In both tables, both the “Aytar” approach and the “DASD” framework perform worse than baseline results. One possible reason is that the selection of significant associations plays an important role. In [92] and [25], the numbers of concepts are 39 and 20, respectively, which are far fewer than current number of concepts. For the “BH” approach, the relationships of different concepts are derived from ontology so the significant links are selected based on domain knowledge. It shows that the “BH” framework performs relatively well in the three comparison meth-

Table 3.9: The contribution of each module in performance boosting in the IACC.1.A data set

Component Change	Overall MAP	Performance Drop
No Change	0.1548	0
Remove PPMC RC Filtering	0.1510	0.0038
Replace the Model Collaboration Module	0.1489	0.0059

ods, which shows that link selection based on domain knowledge is helpful. However, the proposed framework performs better than the "BH" algorithm since the correlations discovered from training data fits better into the testing data than the general prior probabilities.

The proposed framework outperforms the model proposed in the previous section. It indicates that the enhancement for the framework boosts performance. In our previous work, the affinities that are computed purely based on the label matrix are used as the weights to combine scores. In this enhanced model, both labels and scores are considered in modeling the associations through the multi-model collaboration. It shows that the second strategy gives better performance. On the other hand, the accuracy of the related concept models is also considered using the Pearson product moment correlation based filtering module to filter the related models that are less accurate. In order to determine the contribution of each module, the results of the ceiling analysis for the IACC.1.A data set are shown in Table 3.9. It could be observed that the multi-model collaboration module plays a more important role.

In order to determine what strategies to use to combine the logistic regression model and the collaboration model, an empirical study was carried out using 50 randomly sampled concepts in the IACC.1.A data set. The results are shown in Table 3.10. The first column indicates different frameworks. The second and third columns are AUC and

Table 3.10: The comparison of different score fusion strategies

Strategy	AUC	MAP
Logistic Regression Model	0.4950	0.4981
Multimodal Collaboration Model	0.5089	0.5117
Linear Combine, weight = 0.1	0.4977	0.5011
Linear Combine, weight = 0.3	0.5072	0.5103
Linear Combine, weight = 0.5	0.5159	0.5188
Linear Combine, weight = 0.7	0.5190	0.5217
Linear Combine, weight = 0.9	0.5141	0.5169
Logistic Regression	0.5294	0.5324
Naive Bayesian	0.0024	0.0025

the overall MAP. The rows indicate different frameworks. Assuming scores from the logistic regression model and the multimodal collaboration model are represented as S_1 and S_2, "Linear Combine" indicates that the scores are linearly added, which is formulated as $S_L = (1-w)S_1 + wS_2$. "Logistic Regression" indicates the results of fitting a logistic regression model to combine the two scores. "Naive Bayesian" uses the Naive Bayesian approach to integrate two scores. As shown in the table, the logistic regression model achieves the best performance and therefore is utilized in our work. One interesting observation is that by setting the weight carefully, the weighted linear combination strategy can achieve better performance than each individual model. The best performance is achieved by setting the weight to 0.7 for the multimodal collaboration model which gives better performance than the logistic regression model. This observation indicates that the proper combination of two frameworks can take advantage of both frameworks and give better performance.

In summary, the proposed framework conducts concept selection via association rule mining and Pearson product moment correlation based RC filtering. In addition, the collaboration model makes use of the training scores, testing scores, and training labels directly to generate the final outputs. The experimental results show that the proposed framework improves the concept detection accuracy and performs better than other four approaches in comparison.

3.3 Exploring Negative Correlations in the AAN to Enhance Concept Detection

In essence, there are two sorts of correlations among concepts. The first kind of correlation is the positive correlation, which describes the case that the occurrence of one concept increases the chance of appearance of another one, such as the example of “sky” and “cloud.” The second correlation is the negative correlation, which indicates that the existence of one concept decreases the probability of occurrence of another one. In the previous two sections, the positive inter-concept correlations are capitalized on to enhance concept detection. A natural question is whether negative correlations are also helpful. Following this research direction, we propose an AAN, which captures negative associations and utilizes them to enhance the multimedia concept detection framework. In addition, the weights between concept nodes are also enhanced to incorporate features, which give a finer mathematical model. The proposed framework and detailed description of each module are given in Section 3.3.1 and Section 3.3.2. Experimental analysis indicates that negative correlations, if modeled properly, help improve concept detection accuracy.

3.3.1 Overview of the Proposed Framework

Figure 3.10 and Figure 3.11 illustrate an overview of the proposed framework, which consists of a training stage and a testing stage. Similar to the framework proposed in Section 3.1.1, the training stage consists of a “Multimedia Annotation” component and an “AAN Modeling” component. In the former component, for *m* data instances (e.g., images/video shots) and *n* concepts, *n* concept detection models are trained such that for each instance, the model *k* outputs a score measuring the likelihood that concept *k* exists in that data instance. The later component is the main contribution of the proposed framework. First, all the class labels are organized into a label matrix so that each row contains the labels of different concepts for one data instance. Next, significant negative correlations are selected using this label matrix in the negative correlation selection module. A set of features is extracted from the original training data set to train the MCA-based negative weight estimation model. The weights generated from this model together with output scores from “Multimedia Annotation” component are first normalized and then used to train the regression-based score integration model. The selected negative correlations and regression models form the core of trained AAN. The corresponding modules such as “Node Link Modeling” and “Link Selection” are marked in red color the original figure.

In the testing stage, the testing data instances are plugged into all concept detection models to generate testing scores. The same features are extracted from the data instances to get MCA-based weights. After scores and weights are normalized, they are plugged into the regression-based score integration model in the trained AAN to generate a new set of re-ranked scores. Finally, the new output scores are evaluated.

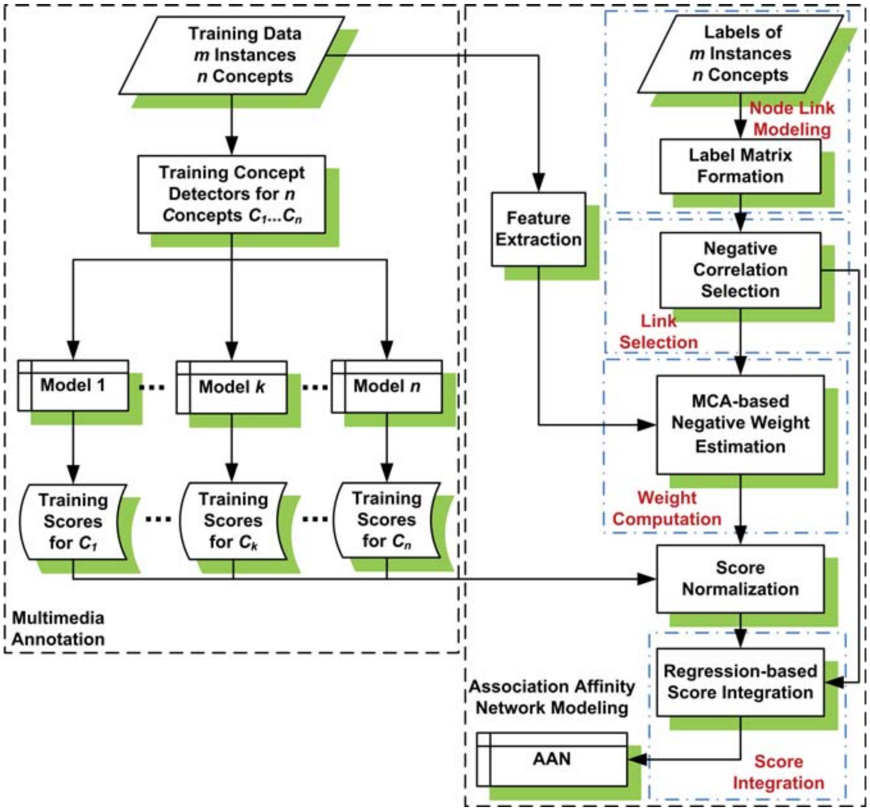

Training Data
m Instances
n Concepts
Training Concept Detectors for *n* Concepts C_1...C_n
Model 1
Model *k*
Model *n*
Training Scores for C_1
Training Scores for C_k
Training Scores for C_n
Multimedia Annotation
Feature Extraction
Labels of *m* Instances *n* Concepts
Node Link Modeling
Label Matrix Formation
Negative Correlation Selection
Link Selection
MCA-based Negative Weight Estimation
Weight Computation
Score Normalization
Regression-based Score Integration
Association Affinity Network Modeling
AAN
Score Integration

Figure 3.10: The proposed framework for the negative AAN (training)

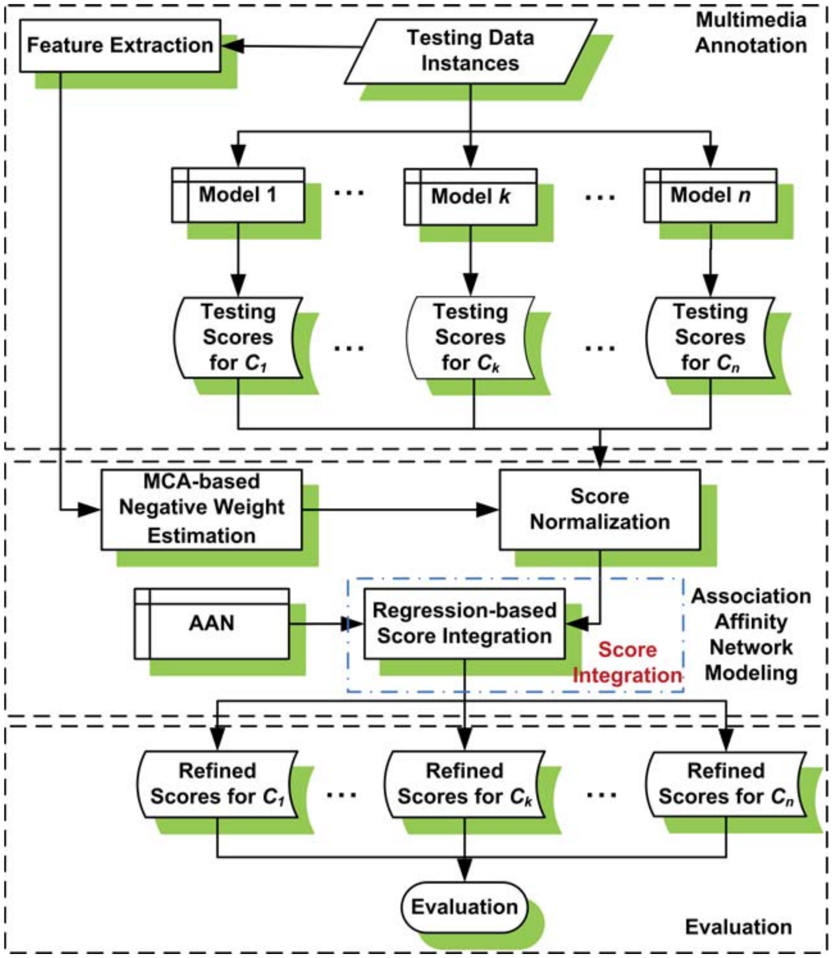


Figure 3.11: The proposed framework for the negative AAN (testing)

3.3.2 Detailed Description of Each Module

A detailed description of each important individual module is given as follows.

Negative Correlation Selection

The initial step of the overall framework is to select significant negative correlations from labels to form the AAN. While the co-occurrence of two concepts in one video shot/image increases the probability that they are positively correlated, the fact that one concept does not occur while the other appears does not indicate they are negatively correlated. For example, given an image that depicts a cat with no human face does not mean the appearance of a cat will suppress the appearance of a face. In addition, in many large-scale multimedia data sets, compared with positive instances, a significant number of negative instances are inferred rather than manually labeled. For instance, the concept pair "Indoor" and "Outdoor" should show perfect negative correlations. However, in our data set, $104,054$ out of $115,806$ instances have negative labels for both concepts. Therefore, a two-step hierarchical selection strategy combining coarse filtering and fine filtering is proposed.

First, a conditional probability-based coarse filtering algorithm is applied. The purpose of this step is to eliminate irrelevant correlations in an efficient way. Following definitions in Section 3.1.2, for a target concept C_t, C_t^1, and C_t^0 represent the events that a data instance is positive or negative for C_t. Likewise, for a related concept, C_r^1 and C_r^0 represent the events that a data instance is positive or negative for C_r. If C_t and C_r are negatively correlated, the following conditions must hold.

$$\frac{P(C_t^0|C_r^1)}{P(C_t^0)} > 1 \tag{3.21}$$

$$\frac{P(C_t^1|C_r^0)}{P(C_t^1)} > 1 \tag{3.22}$$

Here, $P(E)$ indicates the probability of event E. The thresholds of 1, indicated in the inequalities above, are not selected arbitrarily, but are necessary conditions for negative correlations. The first inequality indicates that the probability of C_t appearing decreases if C_r appears. On the other hand, the second inequality indicates that the probability of C_t appearing increases if C_r does not appear. From the association rule mining point of view, these two values are related to the conviction measurement introduced in [177].

The second step is to filter the selected concepts more rigorously. This step addresses the issue of noisy labels in negative instances. Using the aforementioned concept pair "Indoor" and "Outdoor" as an example, $P(C_{Outdoor}^1|C_{Indoor}^0)$ is only 0.0786, which severely deviates from 1. The reason is that $104,054$ out of $115,806$ instances have negative labels for both concepts. Generally speaking, the problem could not be solved by discarding data instances that have negative labels for both concepts simply, as this will introduce the bias that the two concepts are negatively correlated.

In order to tackle this difficulty, a novel strategy is proposed here. It is based on two assumptions. First, if two concepts are negatively correlated, their correlations would not be affected by the existence of the third concept, which is named as a control concept in this thesis. Second, the positive labels are manually annotated and therefore are more reliable. To formulate this problem, an integrated correlation factor (ICF) between a target concept and a related concept is defined in Equation (3.23).

$$ICF(t,r) = \frac{1}{|\Omega|-2} \sum_{\eta \in \Omega, \eta \neq t, \eta \neq r} \rho(C_r, C_t|C_\eta^1) \tag{3.23}$$

Table 3.11: The values to set under special conditions

C_t	C_r	Value To Set
All C_t^1	All C_r^1	1
All C_t^0	All C_r^0	0
All C_t^1	All C_r^0	Average value of negative Pearson correlation coefficients
All C_t^0	All C_r^1	Same as above
Both C_t^0 and C_t^1 appear	All C_r^0	Same as above
All C_t^0	Both C_r^1 and C_r^0 appear	Same as above
All C_t^1	Both C_r^1 and C_r^0 appear	0.5
Both C_t^0 and C_t^1	All C_r^1	0.5

In this equation, Ω indicates the set of all concepts and $|\Omega|$ indicates the total number of concepts. C_η represents the control concept. C_η^1 is the condition that a data instance is positive for C_η, and $\rho(C_t, C_r|C_\eta^1)$ indicates the Pearson product-moment correlation coefficient between the labels of C_t and C_r given C_η^1. The reasons for adding the control concept C_η are as follows. First, the control concept is annotated as positive, which indicates that this data instance is viewed by a human curator. This increases the credibility of other labels for that data instance. Second, ICF represents an average quantitative measurement of correlations under different conditions. For special cases where $\rho(C_t, C_r|C_\eta^1)$ are not defined, default values are assigned as follows in Table 3.11.

In this table, "All C_t^1" indicates that all instances are positive for C_t and "All C_t^0" indicates that all instances are negative for C_t. Correspondingly, "All C_r^1" and "All C_r^0" have similar meanings. All the special cases happen because instances have unique labels for either C_t or C_r, in which case the Pearson correlation coefficients are not defined. As shown in the table, as long as C_t and C_r co-occur once, the value is set to a positive value, which imposes a relatively large penalty on that concept pair. After sorting all the ICF values in an ascending order, the histogram was plotted. The Box-Cox transformation [178] is applied to transform the data to a normal distribution. Different thresholds depending on significance levels such as 95% and 67% could be set to select the significant negative correlations. As the Pearson product-moment correlation coefficients are symmetric, the concept pairs are selected. Therefore, the concept whose corresponding detector is less accurate is chosen as the target concept and the other one as the related concept. The distribution and selected concept pairs are given in Section 3.3.3.

MCA-based Negative Weight Estimation

Weights quantify the impact of related concepts on target concepts. Most previous studies utilized labels in the training data to estimate the weights, which did not consider the observed feature values for each data instance. In this paper, we treat weights as a function of feature values. Formally, for a target concept C_t and a related concept C_r, let F_i represent the visual features for data instance i; the probability that this data instance i is negative given F_i is represented as $P(C_t^0|F_i)$. Associating this probability with the related concept, it could be expanded as follows.

$$\begin{aligned} P(C_t^0|F_i) &= P(C_t^0|C_r^1,F_i)P(C_r^1|F_i)+P(C_t^0|C_r^0,F_i)P(C_r^0|F_i) \\ &= P(C_t^0,C_r^1|F_i)+P(C_t^0,C_r^0|F_i) \end{aligned} \tag{3.24}$$

This equation shows that $P(C_t^0|F_i)$ relies on both $P(C_t^0,C_r^1|F_i)$ and $P(C_t^0,C_r^0|F_i)$. From the correlation point of view, the summation of two conditional probabilities quantifies the impact of the related concept to the target concept given the observed feature values.

In order to estimate these probabilities, the MCA-based model is applied here. MCA is an extension of the standard correspondence analysis to more than two variables. It demonstrates the robustness and relatively high accuracy in modeling the posterior probability distribution [179][180]. Here, we use $P(C_t^0, C_r^1|F_i)$ as an example to show the algorithm to compute these weights. First, the training data instances that are negative for C_t and positive for C_r are selected and labeled as Type I instances. Second, other training data instances are labeled as Type II instances. Third, a MCA model is trained using features, Type I labels, and Type II labels. Fourth, for each testing data instance, the features are plugged into the trained MCA model and a transaction weight is computed. This transaction weight represents the likelihood that the testing data instance belongs to Type I and is used to model the weights. The details of training the MCA model and calculating the transaction weights can be referred in [180]. $P(C_t^0, C_r^0|F_i)$ is estimated in a similar way.

Score Normalization

The reasons for adding a normalization module in the proposed framework are as follows. First, the transaction weights from the MCA-based weight estimation module and those from the original concept detectors need to be made compatible. Second, it is desirable that all scores are converted to well-calibrated probabilities. Given these requirements, the Bayesian conditional probability-based score normalization approach proposed in Section 3.3.2 is also applied here. The details are skipped to avoid duplication.

Regression-based Score Integration

After scores are normalized, the next question is how to integrate the outputs from target concept detectors, related concept detectors, and MCA-based weight estimation models. Specifically, two scenarios, which correspond to a single related concept and multiple related concepts, need to be considered. In the first case, a constrained optimization problem is formulated. Assume for one data instance i, the normalized score from the target concept detector, the related concept detector, and MCA-based weight estimation module are represented by S_t^i, S_r^i, and S_w^i, respectively. Let $\boldsymbol{S}$ be the matrix such that each row $\boldsymbol{S}^i$ is a row vector $[1, S_t^i, S_r^i, S_w^i]$, and $\boldsymbol{\theta}$ is the column vector $[\theta_0, \theta_1, \theta_2, \theta_3]^T$. If the label of a data instance is represented by y^i, it indicates that it is positive ($y^i = 1$) or negative ($y^i = 0$). m is the total number of data instances. Assuming all training data instances are independent and identically distributed, a likelihood function is formulated as follows.

$$
\begin{aligned}
L(\boldsymbol{S};\boldsymbol{\theta}) &= \prod_{i=1}^{m} (g(\boldsymbol{S}^i\boldsymbol{\theta}))^{y^i} \cdot (1 - g(\boldsymbol{S}^i\boldsymbol{\theta}))^{(1-y^i)}, && (3.25)\\
\text{where } g(x) &= \frac{1}{1+e^{-x}} && (3.26)
\end{aligned}
$$

Accordingly, a cost function is defined in Equation (3.27).

$$
J(\boldsymbol{S};\boldsymbol{\theta}) = -\log L(\boldsymbol{S};\boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||_2, \qquad (3.27)
$$

$$
\text{subject to } \theta_1 \geq 0, \theta_2 \leq 0, \theta_3 \leq 0
$$

Here, λ is the regularization parameter to handle the overfitting problem, and $||\boldsymbol{\theta}||_2$ indicates the L_2 norm of $\boldsymbol{\theta}$. This is a constrained optimization problem and could be solved using the active set approach, which is a numerical algorithm to solve the constrained optimization problem. The gradient of $\boldsymbol{\theta}$ is computed in Equation (3.28).

$$\nabla_{\boldsymbol{\theta}} = \boldsymbol{S}^T(\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{S}\boldsymbol{\theta})) - 2\lambda \boldsymbol{I}\boldsymbol{\theta}, \tag{3.28}$$

Here, $\boldsymbol{S}$ has a dimension of m by 4, $\boldsymbol{y}$ is a column vector $[y^1, y^2, ..., y^m]^T$, which has a dimension of m by 1, $\boldsymbol{g}(\boldsymbol{U})$ indicates applying Equation 2.29 on each element in $\boldsymbol{U}$ and has a dimension of m by 1, and $\boldsymbol{I}$ is a 4 by 4 identity matrix.

For a testing data instance f, the vector $\boldsymbol{S}^f$ can be formed in the same way as in the training section. Assume that the estimated parameter $\boldsymbol{\theta}$ is represented as $\hat{\boldsymbol{\theta}}$, the final output score S_F is computed as follows.

$$S_F = g(\boldsymbol{S}^f \hat{\boldsymbol{\theta}}) \tag{3.29}$$

In the second case, the multi-score collaboration approach is utilized. For each pair of the related concept and the target concept, an integrated score is computed using Equation (3.29), and then the integrated scores are further fused using logistic regression. In this case, the score integration module could be viewed as a two-layer neural network. This process is illustrated in Figure 3.12.

3.3.3 Experiments and Results

In this study, the IACC.1.A and the IACC.1.B data sets from TRECVID 2010 and 2011 semantic indexing tasks [54] were used as the benchmark data sets to compare different frameworks. The detailed descriptions of these data sets are given in Section 3.1.4.

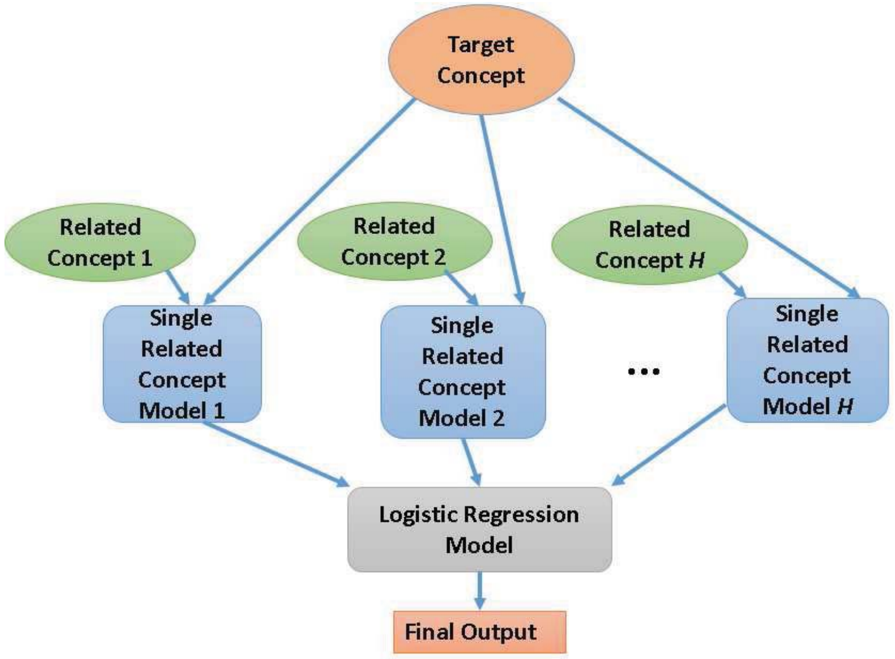

Target Concept
Related Concept 1
Related Concept 2
Related Concept *H*
Single Related Concept Model 1
Single Related Concept Model 2
...
Single Related Concept Model *H*
Logistic Regression Model
Final Output

Figure 3.12: The fusion strategy of multiple related concepts

Negative Correlation Selection Results

In the IACC.1.A data set, there are 130 concepts for evaluation and all pair-wise associations are $8,385$. In the IACC.1.B data set, there are 346 concepts for evaluation and all pair-wise associations come up to $59,685$. The top 10 selected associations from the conditional probability-based selection and the ICF-based selection for the IACC.1.A and the IACC.1.B data sets are shown in Table 3.12 and Table 3.13. For the conditional probability-based selection, the concept pairs are selected by adding the two probability ratios on the left sides in Equation (3.21) and Equation (3.22).

It can be seen that the proposed ICF-based selection approach selects more significant negative associations compared with the conditional probability-based approach. This indicates that the proposed ICF approach is effective. It should be pointed out that some negative correlations are caused by the definitions of concepts. For example,

Table 3.12: The comparison of negative association selection in the IACC.1.A data set

Rank	Conditional Probability-based	ICF-based
1	Entertainment, Building	Indoor, Outdoor
2	Infants, Industrial	Daytime-Outdoor, Indoor
3	Person, Helicopter-Hovering	Indoor, Vegetation
4	Person, Natural-Disaster	Two-People, Single-Person
5	Person, Airplane-Flying	Male-Person, Female-Human-Face-Closeup
6	Canoe, Bus	Trees, Indoor
7	Telephones, Swimming	Indoor, Building
8	Cats, Person	Suburban, Indoor
9	Canoe, Car-Racing	Indoor, Plant
10	Person, Birds	Road, Waterscape-Waterfront

Table 3.13: The comparison of negative association selection in the IACC.1.B data set

Rank	Conditional Probability-based	ICF-based
1	Person, Junk-Frame	Indoor, Outdoor
2	Person, Blank-Frame	Daytime-Outdoor, Indoor
3	Person, Black-Frame	Single-Person, 3-Or-More-People
4	Skyscraper, Ski	Female-Person, Single-Person-Male
5	Birds, Bicycling	Two-People, Single-Person
6	Person, Oil-Drilling-Site	Amateur-Video, Professional-Video
7	Person, Airplane-Takeoff	Male-Person, Single-Person-Female
8	Person, Fighter-Combat	Room, Outdoor
9	Child, Politics	Indoor, Building
10	Person, Flood	Sky, Indoor

the concept “Two people” indicates that there must be exactly two people in the video shot so the “Single_Person” concept does not occur, and the concept “Building” means the shots of an exterior of a building so it has the negative correlation with “Indoor.” The full explanations of all concepts can be found in [54]. However, the conditional probability-based selection module is necessary from the computational point of view. Assume that the number of data instances is m and the number of concepts is n; the time complexity of the conditional probability-based selection module is $O(n^2m)$. Since there is no threshold tuning and each unit computation is a simple summation, this step is relatively efficient. However, the ICF-based approach has a complexity of $O(n^3m)$. For the IACC.1.A data set, in one round of the experiments (on MacBook Pro 2.6GHz Intel Core i7, 8GB RAM), it takes $12,902$ seconds to run all $8,385$ pair-wise associations for $234,387$ data instances using the ICF-based approach directly. However, the conditional probability-based selection module only takes 73.7 seconds. After running the conditional probability-based module, $2,682$ pairs are filtered, reducing the running time of ICF-based selection by $4,126$ seconds.

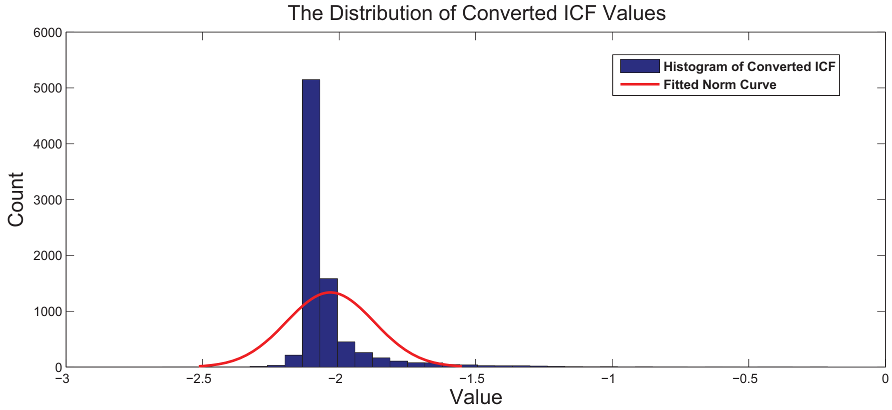


Figure 3.13: The histogram of ICF values for the IACC.1.A data set

As introduced in Section 3.3.2, we generated a histogram and tried to fit the probability density function to all the data. Figure 3.13 and Figure 3.14 show the histogram and probability density function of the fitted norm curve. The selected negative correlations for the IACC.1.A and the IACC.1.B data sets are shown in Appendix C and Appendix D. They are also visualized in Figure 3.15 and Figure 3.16.

Performance of the Proposed Framework

In order to evaluate the effectiveness of the proposed framework, it was compared with the following four frameworks. First, no modifications were made on the raw scores. Second, an intuitive solution that subtracts the scores of a related concept from those of a target concept was applied. Third, related concepts were selected randomly and the same framework was applied. Fourth, the domain adaptive semantic diffusion (DASD) framework [25] was applied. The MAP values for the selected concepts at different numbers of the retrieved data instances, the average precision recall curve, and average AUC are reported. All results are averaged over three-fold cross-validations over selected concepts. In order to compare the effects of thresholds to the performance, two thresholds ($Th1$ and $Th2$) are set to mean NIC minus two standard deviations and one

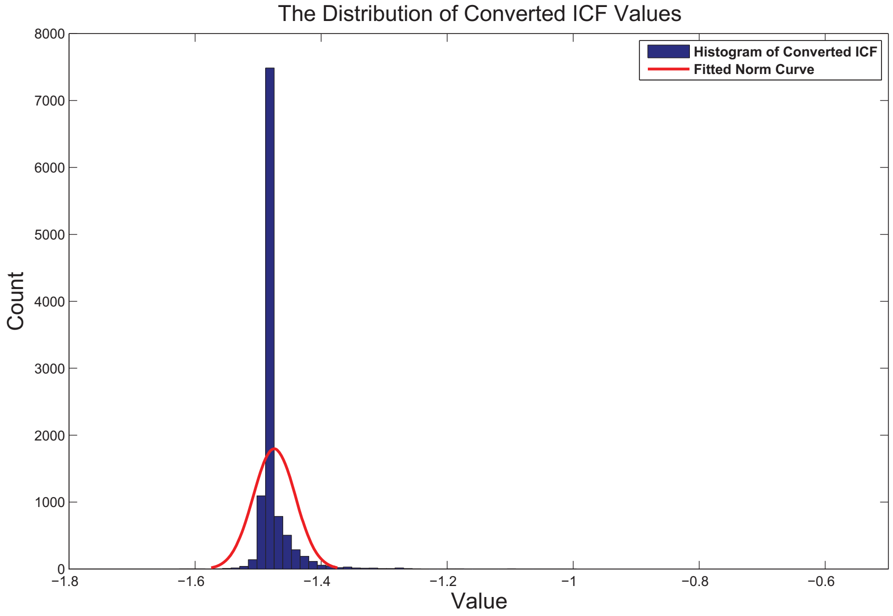


Figure 3.14: The histogram of ICF values for the IACC.1.B data set

standard deviation, respectively. Table 3.14 and Table 3.15 show the MAP comparison results using $Th1$ and $Th2$ for the IACC.1.A data sets. The corresponding average precision-recall curve and AUC are presented in Figure 3.17 and Figure 3.18.

The comparison of different frameworks gives the insights about negative correlations. First, the intuitive solution of the "Subtraction" framework performs worse than the framework using the raw scores only. It indicates that the integration of negative correlations to enhance concept detection is a non-trivial research task. However, some improvements can be observed, such as "MAP@10" in Table 3.14, which indicates that negative correlations can provide some helpful information if they are used properly. Second, the "Random Selection" framework decreases the accuracy tremendously, which indicates that the selection of negative correlations is important in our framework. Third, the "DASD" framework, which utilizes the graph diffusion algorithm,

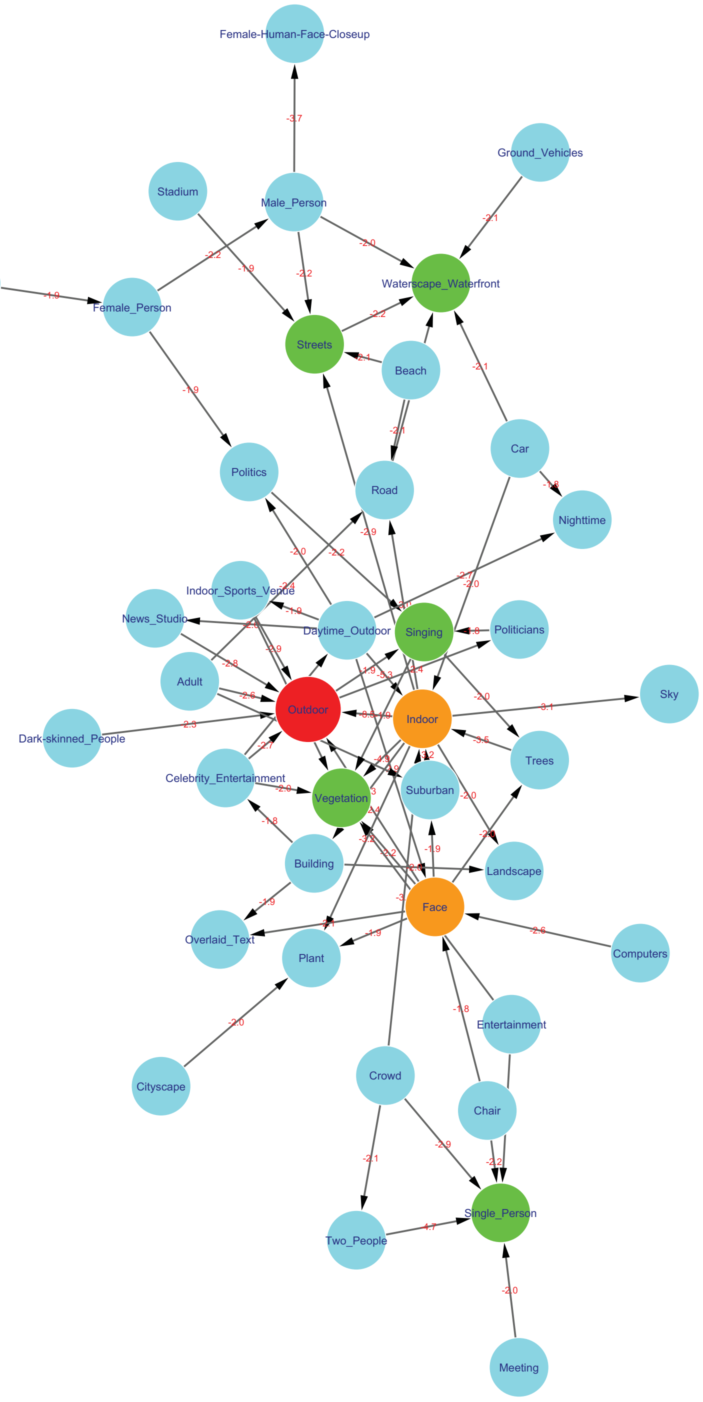


Figure 3.15: The AAN for negative correlations selected for the IACC.1.A data set

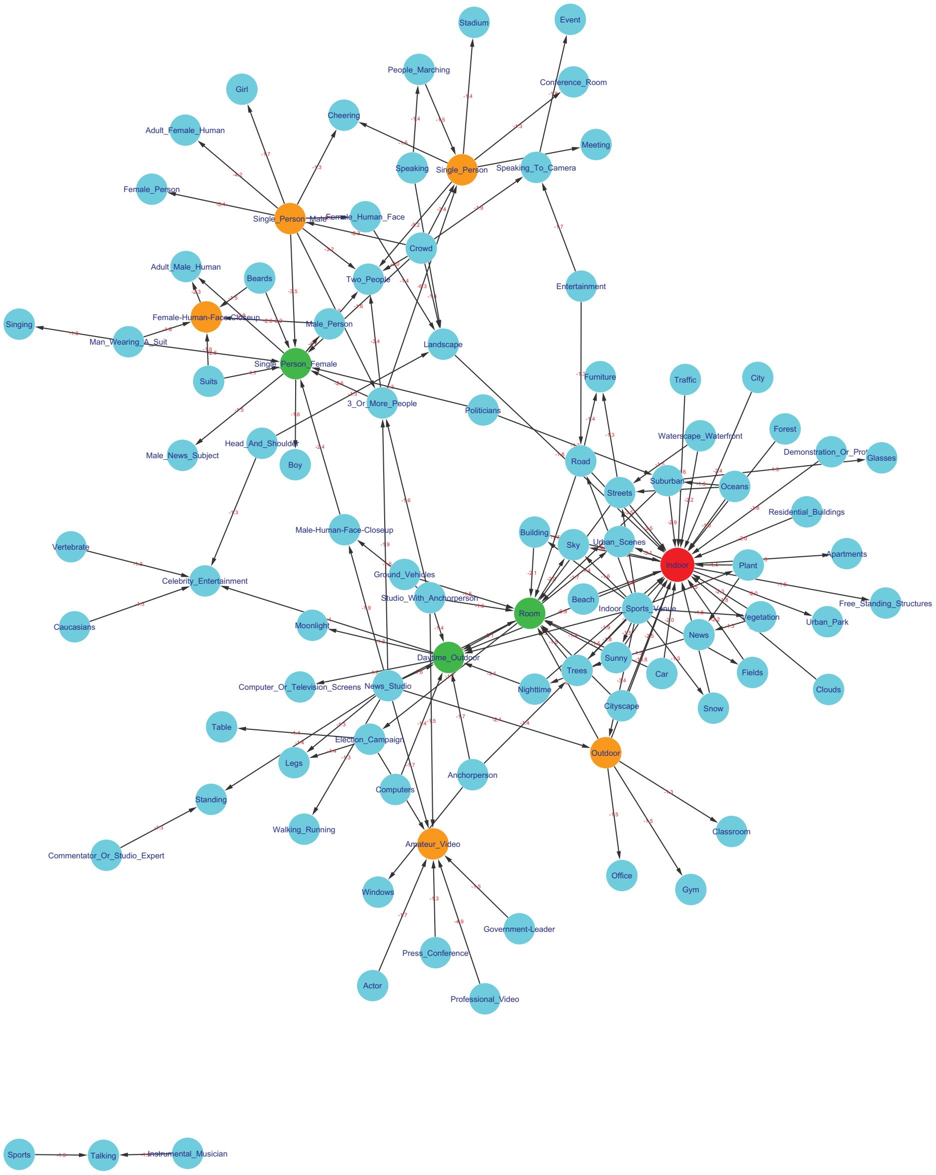


Figure 3.16: The AAN for negative correlations selected for the IACC.1.B data set

shows roughly the same performance as original raw scores. The results match original paper [25], in which the authors claimed that the negative associations hardly improve the performance. On the other hand, our proposed algorithm gives the best performance in all frameworks. The possible reasons are two-fold. First, the proposed negative correlation selection module is able to capture significant negative associations such as “Indoor” vs. “Outdoor,” given the challenging condition that many negative labels are inferred rather than manually annotated. Second, the MCA-based weight estimation model, which estimates two conditional probabilities, $P(C_T^-, C_R^+|F_i)$ and $P(C_T^-, C_R^-|F_i)$, gives a better model than the one utilizing the probability estimated from labels only, such as in [25]. In conclusion, the experimental results show that the negative correlations could help concept detection if modeled properly.

Comparing the results of Table 3.14 and Table 3.15, it can be seen that the improvements of the negative relationships are more significant when using a relatively strict selection criteria. As the condition of negative correlation selection is relaxed, more insignificant negative correlations are involved and the assistance of negative correlations decreases.

Table 3.16 and Table 3.17 show experimental results for the IACC.1.B data sets. The corresponding precision-recall curves and AUC values are given in Figure 3.19 and Figure 3.20. The corresponding mean inferred average precision based on the official evaluation results are 0.100683 and 0.157503. These results are different from those in the tables because of different experimental settings.

3.4 Conclusion

In this chapter, the framework of utilizing the AAN in multimedia concept detection is introduced. Both the positive and negative correlations are utilized. From the ex-

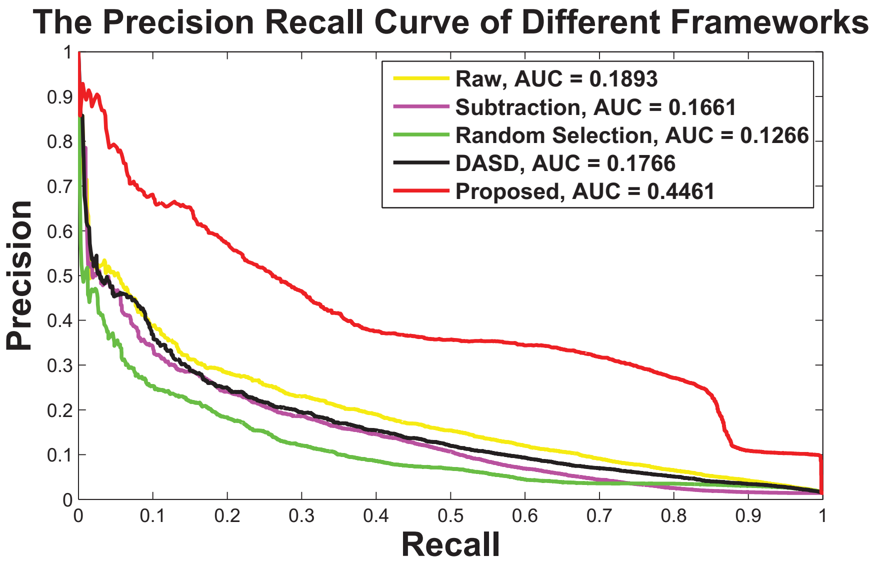


Figure 3.17: The precision recall curve for the IACC.1.A data set (Threshold=$Th1$)

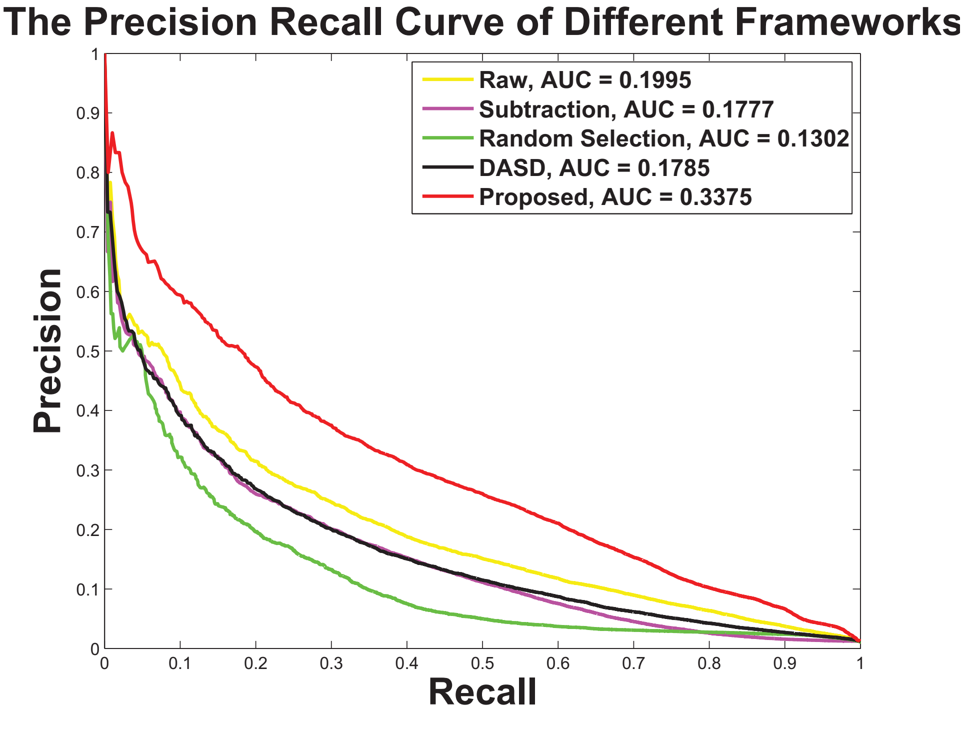


Figure 3.18: The precision recall curve for the IACC.1.A data set (Threshold=$Th2$)

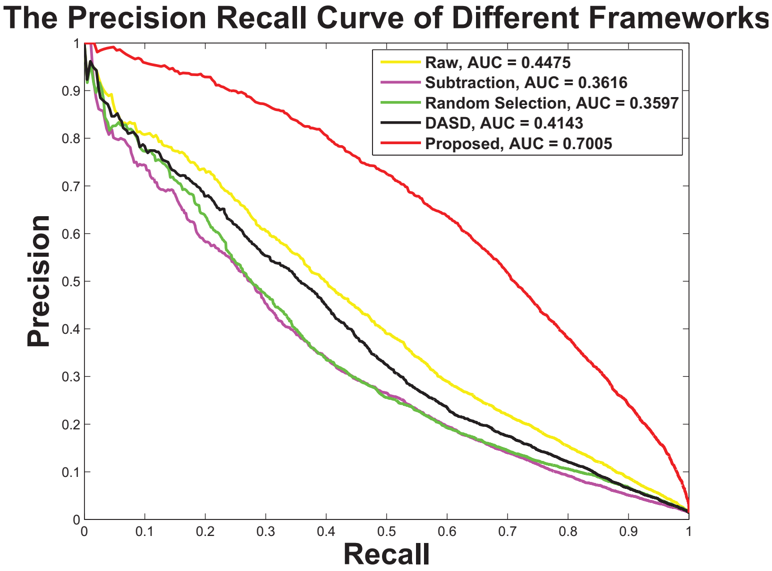


Figure 3.19: The precision recall curve for the IACC.1.B data set (Threshold=$Th1$)

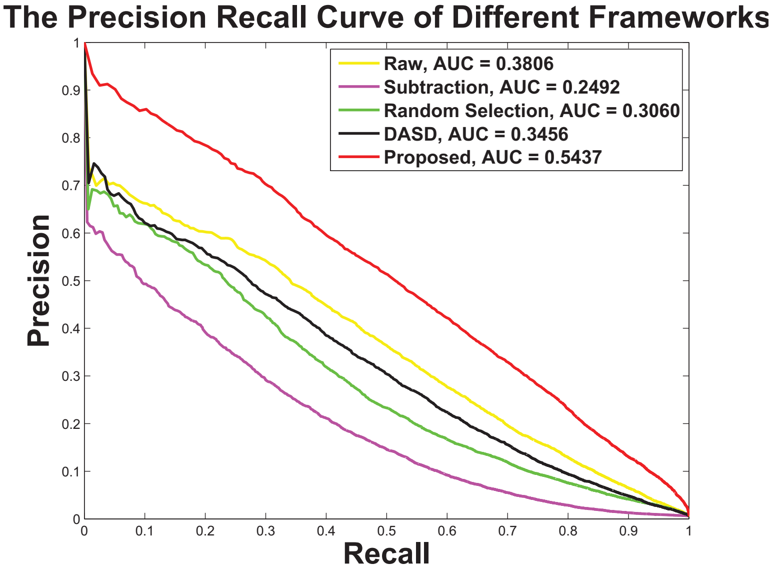


Figure 3.20: The precision recall curve for the IACC.1.B data set (Threshold=$Th2$)

Table 3.14: MAP values at different number of instances retrieved for the IACC.1.A data set (Threshold=$Th1$, No. of TC: 7, No. of Selected Links: 10)

Frameworks	Top10	Top20	Top40	Top60	Top80	Top100	Top500	Top1000	Overall
Raw	0.4508	0.4084	0.3576	0.3137	0.2738	0.2441	0.1305	0.1442	0.1910
Subtraction	0.4729	0.3997	0.3391	0.2872	0.2537	0.2227	0.1155	0.1281	0.1673
Random Selection	0.3601	0.3156	0.2317	0.1844	0.1587	0.1397	0.0845	0.0929	0.1280
DASD	0.4827	0.4020	0.3340	0.3113	0.2786	0.2431	0.1222	0.1339	0.1778
Proposed	0.8626	0.7355	0.6054	0.5588	0.5105	0.4729	0.3397	0.4062	0.4478

Table 3.15: MAP values at different number of instances retrieved for the IACC.1.A data set (Threshold=$Th2$, No. of TC: 30, No. of Selected Links: 67)

Frameworks	Top10	Top20	Top40	Top60	Top80	Top100	Top500	Top1000	Overall
Raw	0.4782	0.4060	0.3397	0.2911	0.2577	0.2346	0.1555	0.1676	0.2019
Subtraction	0.4472	0.3798	0.3169	0.2734	0.2424	0.2197	0.1400	0.1506	0.1797
Random Selection	0.3990	0.3548	0.2406	0.1974	0.1745	0.1541	0.1010	0.1056	0.1323
DASD	0.4719	0.3890	0.3095	0.2682	0.2399	0.2154	0.1377	0.1489	0.1806
Proposed	0.6472	0.5485	0.4614	0.4157	0.3827	0.3539	0.2868	0.3044	0.3396

perimental results, it can be observed that the correlations among different concepts, which are modeled in the AAN, could help improve the concept detection accuracy. In addition, the proper selection of the related concepts for each target concept plays an important role in this process.

Table 3.16: MAP values at different number of instances retrieved for the IACC.1.B data set (Threshold=$Th1$, No. of TC: 13, No. of Selected Links: 44)

Frameworks	Top10	Top20	Top40	Top60	Top80	Top100	Top500	Top1000	Overall
Raw	0.8585	0.7887	0.7078	0.6497	0.5967	0.5470	0.4093	0.4190	0.4491
Subtraction	0.7710	0.7195	0.6072	0.5435	0.4873	0.4524	0.3217	0.3313	0.3629
Random Selection	0.7823	0.7433	0.6261	0.5414	0.4904	0.4452	0.3283	0.3337	0.3615
DASD	0.8353	0.7699	0.6873	0.6224	0.5731	0.5246	0.3756	0.3836	0.4158
Proposed	0.9781	0.9433	0.9011	0.8464	0.8111	0.7762	0.6699	0.6851	0.7012

Table 3.17: MAP values at different number of instances retrieved for the IACC.1.B data set (Threshold=$Th2$, No. of TC: 61, No. of Selected Links: 151)

Frameworks	Top10	Top20	Top40	Top60	Top80	Top100	Top500	Top1000	Overall
Raw	0.6133	0.5502	0.4783	0.4302	0.3928	0.3661	0.3598	0.3723	0.3846
Subtraction	0.4730	0.4240	0.3449	0.3022	0.2704	0.2479	0.2290	0.2388	0.2524
Random Selection	0.5581	0.5026	0.4173	0.3592	0.3251	0.2979	0.2850	0.2965	0.3101
DASD	0.5934	0.5189	0.4493	0.3962	0.3603	0.3359	0.3243	0.3361	0.3495
Proposed	0.8042	0.7329	0.6341	0.5611	0.5287	0.5086	0.5243	0.5372	0.5467

Chapter 4

Utilizing the AAN in Multi-Class Classification for Biomedical Images

In this chapter, the AAN is adapted to solve the multi-class classification problem in biomedical images. Specifically, this chapter focuses on the temporal stage annotation of biological images. In Section 4.1, the problem of biological temporal image classification is introduced. In Section 4.2, the proposed framework of the application of the AAN in solving this problem is introduced and each component is described in detail. In Section 4.3, experimental results and analysis are provided. Section 4.4 concludes this chapter and gives some insights based on experimental results.

4.1 Biological Temporal Stage Annotation

A biological process usually consists of several temporal stages. For example, the mitosis process consists of the stages of prophase, prometaphase, metaphase, anaphase, and telophase. In each stage, certain steps are taken to promote the progress of the mitosis. In the process of *Drosophila* embryogenesis, the correct sequence of gene expressions and interactions ensures the normal development of an individual organism from embryo to adult. Recently, owing to the rapid advances of the high-throughput microscopic imaging technology, middle to large-scale biological image repositories have been built and the number of available images is increasing exponentially. A case in point, the BDGP database [181], which is a public database constructed by the University of California at Berkeley, currently contains 115,006 ISH images spanning six developmental stage ranges of *Drosophila*, and the total size of these images is already over 100GB [12]. Figure 4.1 shows the *Drosophila* embryos in six developmental stage ranges.

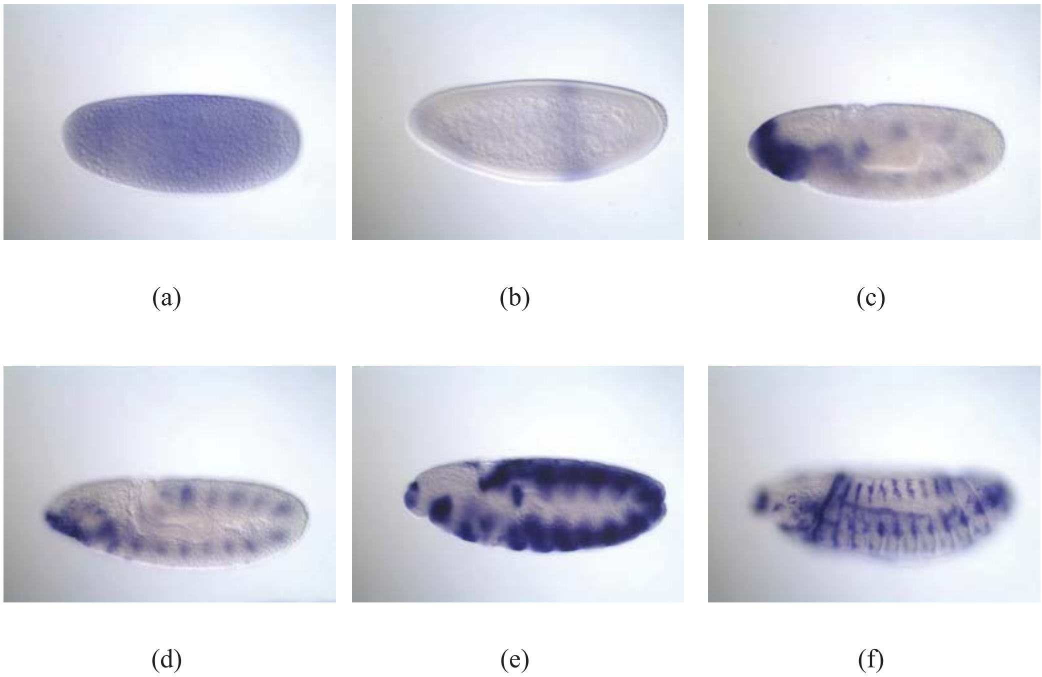

(a) (b) (c) (d) (e) (f)

Figure 4.1: Sample images of six developmental stage ranges of *Drosophila* embryo (a) Stage 1-3 (b) Stage 4-6 (c) Stage 7-8 (d) Stage 9-10 (e) Stage 11-12 (f) Stage 13-16

In this chapter, the AAN is extended to help multi-class classification. Our assumption is that the relatively complicated multi-class classification could be improved by integrating classification results from relatively easy classification tasks. Therefore, some middle layer assisting classes are created to help improve the final classification results. In order to achieve this, three essential questions need to be answered. The first question is how to generate the middle layer assisting classes so the relationship of different classes can be considered properly. The second question is how to model the weights between the different classes to help improve the final classification results. The third question is how to address the error propagation issue. In the following sections, the general framework and detailed solutions to these problems are presented.

4.2 The Proposed Framework

Our proposed framework is shown in Figure 4.2 and Figure 4.3. It consists of the training stage and the testing stage. In the training stage, the general framework contains the "Multimedia Annotation" component and the "AAN Modeling" component. In the "Multimedia Annotation" component, the training images are first preprocessed, if necessary. A set of visual descriptors including color, texture, edge, etc., is extracted from each image. The feature selection step selects the most significant features for the following processing steps. Next, two kinds of classifiers, the middle layer classifiers and the end layer classifiers, are trained and their output scores are used in the "AAN Modeling" component. Here, the end layer classifier performs the initial classification task and the corresponding classes are named as end classes. On the other hand, the middle layer classifier performs the classification for the classes which are created in the "Node Link Modeling" module and these classes are named middle classes. The "AAN Modeling" component consists of the "Node Link Modeling," "Score Normalization," and "Link Selection & Weight Computation" modules. In the "Node Link Modeling," the middle classes are created and both the end classes and middle classes are represented as nodes in the AAN. The "Score Normalization" module performs similar function as described in the previous chapter. In this work, the "Multi-Layer Model Collaboration" framework is proposed to perform the function of "Link Selection" and "Weight Computation" at the same time. The "Link Selection" follows the soft selection ap-

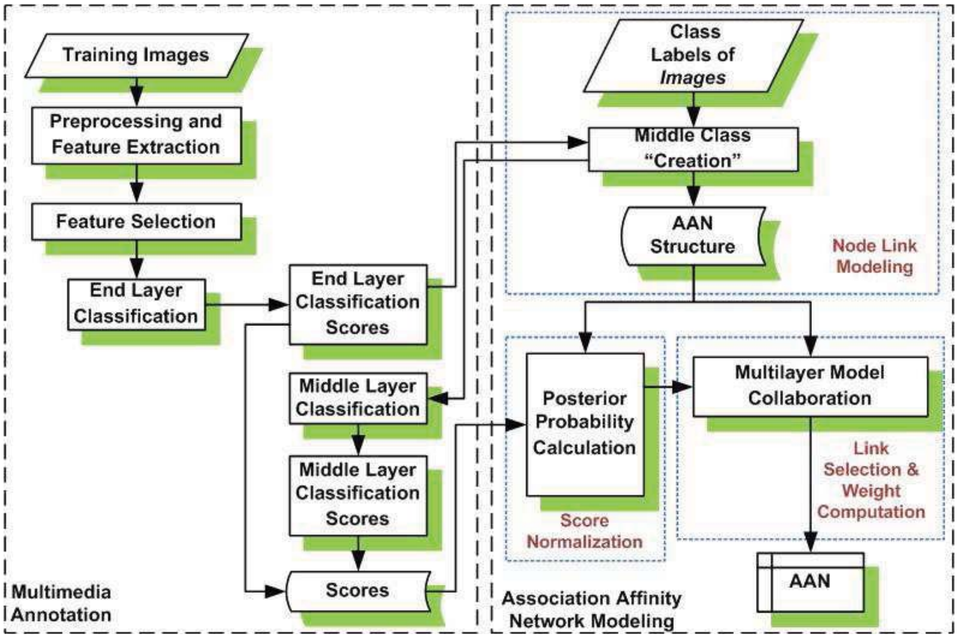

Training Images
Preprocessing and Feature Extraction
Feature Selection
End Layer Classification
End Layer Classification Scores
Middle Layer Classification
Middle Layer Classification Scores
Scores
Multimedia Annotation
Class Labels of *Images*
Middle Class "Creation"
AAN Structure
Node Link Modeling
Posterior Probability Calculation
Score Normalization
Multilayer Model Collaboration
Link Selection & Weight Computation
AAN
Association Affinity Network Modeling

Figure 4.2: The training stage of the proposed framework

proach. In the testing stage, the testing images are preprocessed in the same way as in the training stage. The same set of features is extracted and the features corresponding to the selected features in the training stage are retained. Next, the AAN is applied so the scores from the end layer classifier and the middle layer classifier are integrated to give the final classification results. The classification accuracy is used as the metric for evaluation.

4.2.1 Image Preprocessing

Since the proposed framework is geared towards the classification of a wide range of biological images, the images could be of different characteristics. As a result, the input raw images are preprocessed in a sequence of steps such as the histogram equalization and normalization if necessary. Particularly, for those bioimages in which the objects are clearly discernible from the background, the object segmentation is performed be-

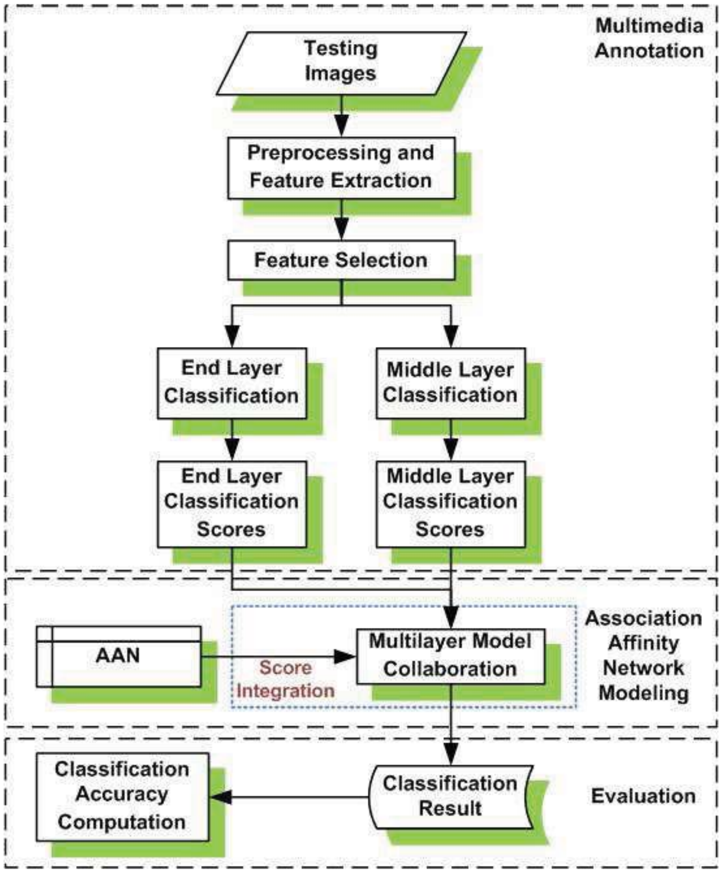


Figure 4.3: The testing stage of the proposed framework

fore further processing steps so the irrelevant background information is eliminated from the beginning. Figure 4.4(a) shows an example where the embryo is clearly differentiable from the background, so a segmentation process is performed. The detailed procedure of this segmentation is shown in Section 4.3.1; Figure 4.4(b) shows an example of the raw image being used directly without segmentation. It is important to point out that this step is the regular image processing step but not the focus of this work.

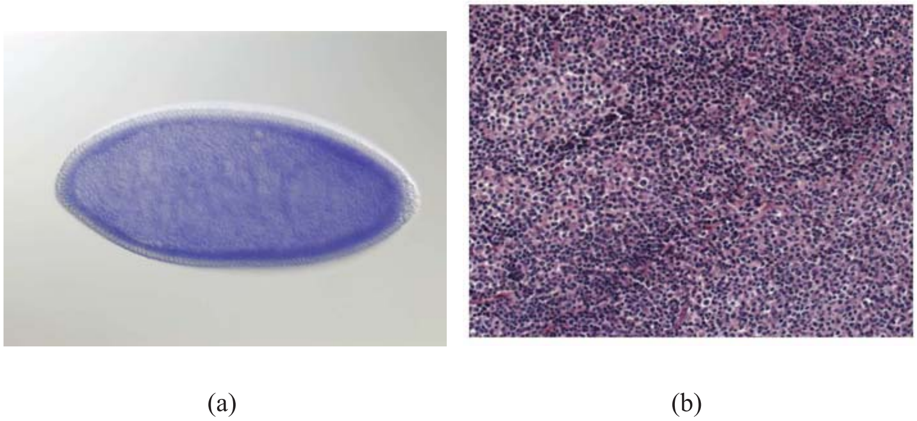


Figure 4.4: The comparison of one bioimage with a clear background and another with an unclear background (a) One sample image from the BDGP database (b) One sample image from the follicular lymphoma microscopic histology database

4.2.2 Feature Extraction and Feature Selection

In order to apply the machine learning based approach, the first step is to represent the image using a set of descriptors or features. Such representation should be able to cover most of the information contained in the images. In fact, good feature representation of the object/target is an active research area itself and it never stops developing. Therefore, the proposed framework is designed to be flexible so it could accept the input of any feature descriptors provided by the users. If there are no features specified by the user, we provide a default feature set consisting of 640 visual descriptors. Those features generally fall into the following categories: color features such as color dominance, color histogram and color moment, edge features such as edge histogram, texture features such as wavelet, and Gabor and histogram of oriented gradients (HOG) [182]. Based on our empirical testing on a variety of biological images, this relatively comprehensive feature set covers most of the necessary information and is adopted here.

With all the features available, the next important task is to decide which subset of features to retain in order to decrease the computational cost as much as possible on one hand, but to still capture the characteristics of data on the other hand. In this work, we first use the chi-square feature selection, which evaluates the features by ranking the chi-square statistics of each feature with respect to the class to get the ranks of all features and perform an empirical study to decide the number of features to retain. The details of the feature selection process are introduced in Section 4.3.2.

4.2.3 Subspace-Based Classifier and Posterior Probability Calculation

The AAN requires the multi-class classification model to output confidence scores for each class. In this work, the subspace-based classifier extended from the C-RSPM model [183] is used as the classification model. The general idea is to build an array of Principal Component Classifiers (PCOCs), each of which is trained to learn the similarities among the data instances from a particular class. Specifically, for $PCOC_k$, which is the subspace classifier for $Class_k$, suppose the training data instances from $Class_k (1 \leq k \leq K)$ (K is total number of classes) form a matrix $\boldsymbol{X}_k = \{x_{t\varphi}\}$, t = 1, 2, ..., m and φ = 1, 2, ..., Φ, where m indicates the total number of training data instances from $Class_k$ and Φ is the total number of selected features. Next, the outliers are removed from matrix $\boldsymbol{X}_k$ by calculating the Mahalanobis distance. Let $\boldsymbol{x}_t$=$[x_{t1}, x_{t2}, \ldots, x_{t\Phi}]^T$ be the column vector representing the feature vector for the t-th instance. The Mahalanobis distance is calculated using Equation (4.1), where a parameter trimming percentage ($\gamma\%$) is utilized so that the number of data instances to be retained is $(1-\gamma\%)\cdot m$.

$$d_t^2 = (\boldsymbol{x}_t - \overline{\boldsymbol{x}})^T \boldsymbol{S}^{-1} (\boldsymbol{x}_t - \overline{\boldsymbol{x}}) \tag{4.1}$$

$$\text{where } \overline{\boldsymbol{x}} = \frac{1}{m}\sum_{t=1}^{m} \boldsymbol{x}_t, \text{ and}$$

$$\boldsymbol{S} = \frac{1}{m-1}\sum_{t=1}^{m} (\boldsymbol{x}_t - \overline{\boldsymbol{x}})(\boldsymbol{x}_t - \overline{\boldsymbol{x}})^T$$

The rest of the data instances form a matrix $\boldsymbol{B}_k = \{b_{w\varphi}\}$ where w = 1, 2,..., W, φ = 1, 2, ..., Φ, which has W data instances ($W \leq m$). The principal component analysis is applied on matrix $\boldsymbol{B}_k$ and only those principal components whose corresponding eigenvalues are greater than the threshold are kept. Next, $\boldsymbol{B}_k$ is projected to this subspace and a new data matrix $\boldsymbol{R}_k = \{r_{wf}\}$ in which the row vector $\boldsymbol{r}_w = [r_{w1}, r_{w2}, ...r_{wf}, ..., r_{wF}]$ ($F \leq \Phi$) is the projection of the instance $\boldsymbol{b}_w = [b_{w1}, b_{w2}, ..., b_{w\Phi}]$ in matrix $\boldsymbol{B}_k$. For each instance w in $\boldsymbol{R}_k$, a distance value named $Cd_k^{(w)}$ is computed using Equation (4.2).

$$Cd_k^w = \sum_{f=1}^{F} \frac{r_{wf}^2}{\lambda_f}, \tag{4.2}$$

where λ_f is the eigenvalue corresponding of the f-th principal component.

In order to get the normalized score, the posterior probability score is calculated as follows. The generated Cd values for all the training instances in $Class_k$ could be computed and its probability density function (pdf) $P(Cd_k|C_k = 1)$ ($C_k = 1$ represents $Class_k$ is the ground truth label for the instance) is estimated using the Parzen window approach [172], which is a non-parameter probability density estimation approach. On the other hand, the negative instances for $Class_k$ (all the training instances which are not from $Class_k$) could be projected to the same subspace and the corresponding Cd_k values can be computed. So $P(Cd_k|C_k = 0)$ for the negative instances could be estimated using Parzen Window approach as well. For a testing instance *i*, the Cd_k^i is computed and the final probability of testing instance *i* from $Class_k$ is computed using Bayes' rule shown in Equation (4.3).

$$P(C_k = 1|Cd_k^i) = \frac{P(Cd_k^i|C_k = 1) \cdot P(C_k = 1)}{\sum_{q=0}^{1} P(Cd_k^i|C_k = q) \cdot P(C_k = q)} \tag{4.3}$$

$P(C_k = 1)$ is the prior probability that an instance belongs to $Class_k$, which could be estimated from the training instances. $P(C_k = 0)$ equals to $1 - P(C_k = 1)$. Therefore, for one instance, each $PCOC_k$ outputs a posterior probability indicating the likelihood that a testing instance belongs to $Class_k$. For a multi-class classification problem, the final classification decision is made by choosing the class that corresponds to the highest posterior probability. If there is a tie, the decision will be determined by prior probabilities.

4.2.4 Middle Class Creation

The preceding classification framework aims at solving the problem directly and tries to differentiate all classes at once. This relatively strict requirement presents challenges to the model. On the other hand, when facing a difficult problem, humans usually adopt the strategy of "divide and conquer" to break the problem into small pieces, solve them first, and integrate the solutions to solve the difficult one. Inspired by this strategy, some middle-level classes which are not our targets but are helpful in solving the target problem, are created. In order to eliminate ambiguity, we define the classes created as **middle classes** and the corresponding classifiers as the **middle layer classifiers**, while the classes for the initial classification task are defined as **end classes** and the corresponding classifiers as the **end layer classifiers**. Figure 4.5 shows an example of the structure of a two-layer framework. In this figure, without loss of generality, C_1, C_2, ... C_k ..., C_K represent K end classes, and C'_1, C'_2, ..., C'_h, ..., C'_H represent H middle classes. TM_1, TM_2, ..., TM_k, ..., TM_K represent K end layer classifiers corresponding to K end classes, while MM_1, MM_2, ..., MM_h, ..., MM_H represent H middle layer classifiers corresponding to H middle classes. For example, C_1, C_2, and C_3 form a middle class C'_1 in the figure and the corresponding middle layer classifier is MM_1. In a special case, the middle class could just have one end class and the middle layer classifier is the same as the end layer classifier. For example, C'_h in the figure only contains C_k and $MM_h = TM_k$ in this case. It should be pointed out that Figure 4.5 is just a simple example. The number of layers can be extended for more complicated tasks and the formation of the middle classes could be changed.

From the AAN point of view, the addition of middle classes changes the structure of the network. Figure 4.6 shows the structure of classifiers before adding middle classes. The features of a data instance are input into to end layer classifiers directly and each classifier outputs a score. The final decision is made by integrating these scores. Figure 4.7 shows the network structure after adding middle classes. In this structure, another layer of classification models is added. The final score for a certain class is computed by integrating information from an end layer classifier and all middle layer classifiers. It is important to point out that all classifiers are trained using original features. This is an important difference between the proposed framework and a neural network model.

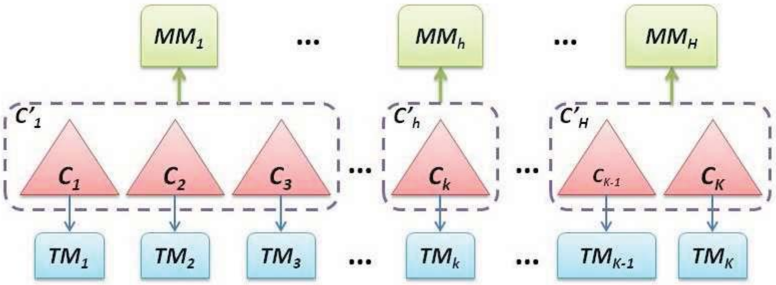


Figure 4.5: The structure of the two-layer classifiers. The dashed rectangles indicate the middle classes which consist of different number of end classes

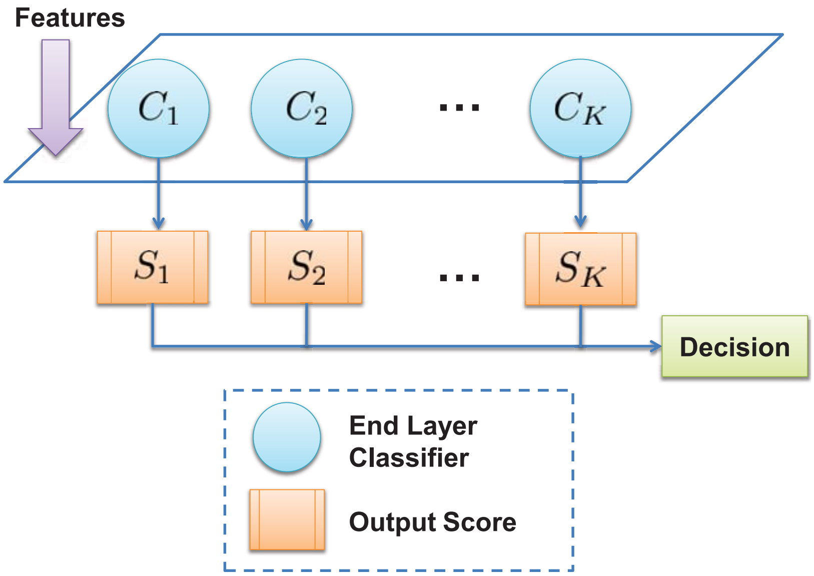


Figure 4.6: The structure of the original AAN

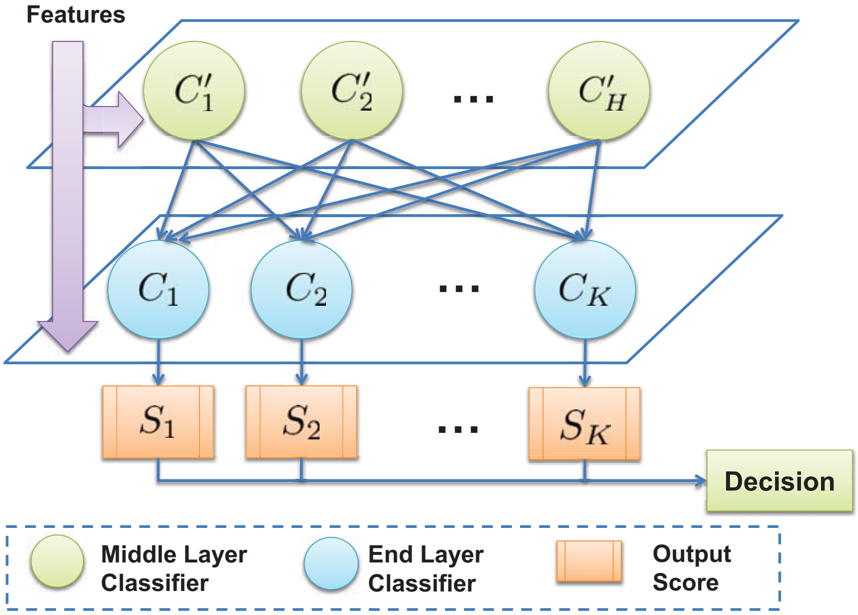


Figure 4.7: The structure of the AAN after adding middle classes

In terms of creating the middle classes, one way is to consult the domain expert. Like cases in other data mining applications, the domain expert could provide critical information that helps decrease the searching space. If such information is unavailable, two data-driven determination strategies are proposed here. The first approach is the supervised approach. The main idea of this approach is that the middle classes should help each end class classification and also match the domain knowledge. The whole procedure of the supervised approach is shown in Figure 4.8.

The end layer classifier is first trained using the training instances and the classification results are gained on the validation data set. Based on the classification results, the confusion matrix is computed so the row of the matrix is sorted according to the temporal sequence of the classes. Next, the candidate division points are found for each end class by identifying the class that corresponds to the largest classification error. The final grouping strategy is decided by combining the candidate division points of each

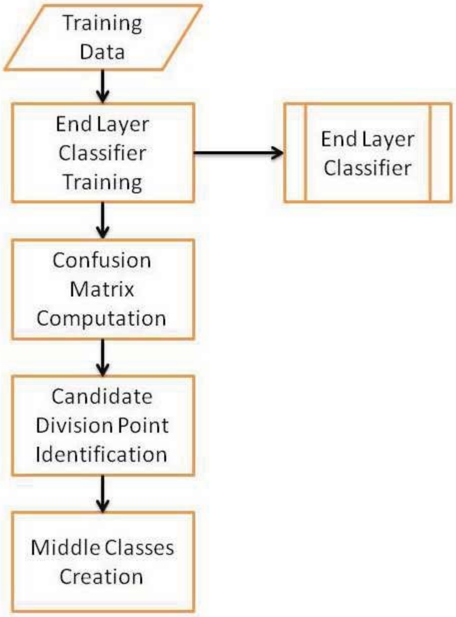
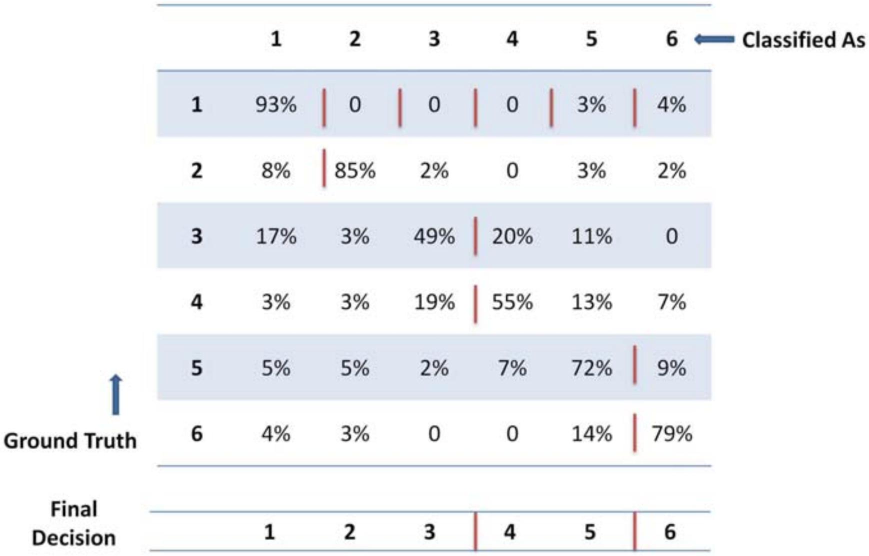

Training Data
End Layer Classifier Training
End Layer Classifier
Confusion Matrix Computation
Candidate Division Point Identification
Middle Classes Creation

Figure 4.8: The flow chart of creating middle classes using the supervised approach

end class using the majority voting strategy with the constraint that the total number of middle classes is less than or equal to half of the total number of end classes and greater than or equal to two. Here, a specific example with six end classes is given. Figure 4.9 shows a confusion matrix for the classification results based on the validation data set. In this matrix, the rows indicate the ground truth class labels and the columns indicate the predicated class labels. The class 1, 2, ..., 6 represent the six consecutive temporal stages. Each element in the row represents the ratio of the number of instances predicted as a certain class, so all the diagonal elements are correct while all the off-diagonal elements are incorrect. For example, for all the instances from $Class_1$, 93% is predicted as $Class_1$, 3% as $Class_5$, and 4% as $Class_6$. Therefore, the class that corresponds to the largest error is the $Class_6$ and the candidate division points are between $Class_1$ and $Class_6$ shown as the vertical bar in the figure. Afterwards, the final decision is made using the majority voting strategy. In this example, given the constraint that


Classified As
Ground Truth
Final Decision

	1	2	3	4	5	6
1	93%	│ 0	│ 0	│ 0	│ 3%	│ 4%
2	8%	│ 85%	2%	0	3%	2%
3	17%	3%	49%	│ 20%	11%	0
4	3%	3%	19%	│ 55%	13%	7%
5	5%	5%	2%	7%	72%	│ 9%
6	4%	3%	0	0	14%	│ 79%
Final Decision	1	2	3	│ 4	5	│ 6

Figure 4.9: A numerical example of a confusion matrix and the division scheme (the candidate division points are labeled as the vertical bar in the confusion matrix; the final decision of the division points is illustrated at the bottom of the figure)

the number of middle classes should be less than or equal to three, two division points are chosen and three middle classes are created. For the special case that the overall accuracy is 100%, which indicates the end layer classifiers are perfect for the testing data, two middle classes are formed so that the total number of positive instances for each middle class are as close as possible. It is noticed that the temporal sequence of different classes is not changed so that only the consecutive stages can be grouped together. In this way, the internal connections among classes are properly integrated into the proposed classification model.

In some applications, the supervised approach is not suitable for two reasons. First, the combinations of classes can be complicated because there is no sequential relationship between multiple classes in general. Second, in order to compute the confusion matrix, more data need to be held in the validation data set. This could face challenges

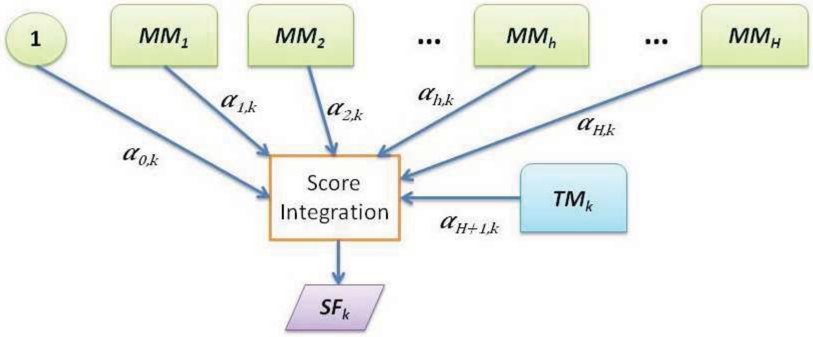


Figure 4.10: The strategy for combining scores from middle layer classifiers and end layer classifiers for the $Class_k$

when the amount of data is limited. In order to address the aforementioned two problems, an unsupervised solution is proposed. The algorithm consists of two steps. The first step is to carry out a clustering step for both training and testing data so each data instance is assigned to a group number. Each cluster is treated as a middle class. Next, for each instance, each middle class outputs 0 or 1, which indicates whether that instance belongs to that cluster or not. The outputs from all middle classes are plugged into the muli-layer model collaboration module introduced in the following section.

4.2.5 Multi-Layer Model Collaboration

After the middle classes are created, the middle classes and the end classes form the node of the AAN. The next question is how to perform link selection and weight computation. In this work, we propose the multi-layer model collaboration strategy to perform these functions. This framework has two characteristics. First, it does not significantly increase computational cost. Second, it relies on a closed-form cost function that can be minimized numerically in order to calculate the affinities of the links. The models are formulated as follows: Let C_1, C_2, ..., C_k, ..., C_K represents K end classes; H is the total number of middle classes, which are represented as C'_1, C'_2, ..., C'_h, ..., C'_H; TM_1,

TM_2, ..., TM_k, ..., TM_K are K end layer classifiers corresponding to end classes; and MM_1, MM_2, ..., MM_h, ..., MM_H are H middle layer classifiers corresponding to middle classes. For an instance i ($1 \leq i \leq m$), where m is the total number of training instances, the output score of instance i from the middle layer classifier MM_h is represented as SM_h^i and the output score from the end layer classifier TM_k is represented as ST_k^i. In addition, a weight matrix $\boldsymbol{\alpha}$ with a dimension of $(H+2)$ by K is defined as:

$$\begin{bmatrix} \alpha_0^1 & \alpha_0^2 & \cdots & \alpha_0^k & \cdots & \alpha_0^K \\ \alpha_1^1 & \alpha_1^2 & \cdots & \alpha_1^k & \cdots & \alpha_1^K \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_h^1 & \alpha_h^2 & \cdots & \alpha_h^k & \cdots & \alpha_h^K \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_H^1 & \alpha_H^2 & \cdots & \alpha_H^k & \cdots & \alpha_H^K \\ \alpha_{(H+1)}^1 & \alpha_{(H+1)}^2 & \cdots & \alpha_{(H+1)}^k & \cdots & \alpha_{(H+1)}^K \end{bmatrix}$$

Each column represents weights for one end class C_k. The final score of instance i for that end class C_k is represented as SF_k^i, which is defined using Equation (4.4).

$$\begin{aligned} SF_k^i &= g(1 \cdot \alpha_0^k + \sum_{h=1}^{H} SM_h^i \cdot \alpha_h^k + ST_k^i \cdot \alpha_{(H+1)}^k), \qquad (4.4) \\ \text{where } g(x) &= \frac{1}{1+e^{-x}} \end{aligned}$$

Figure 4.10 illustrates the strategy of combining output probabilities for C_k, where “1” is the bias unit. In this strategy, the linear model is utilized to save computational cost.

In order to estimate parameter matrix $\boldsymbol{\alpha}$, let $y(i)$ represent the ground truth label for instance i, where $1 \leq y(i) \leq K$. The cost function is defined in Equation (4.5).

$$J = \frac{1}{2m} \cdot \sum_{i=1}^{m} [(SF^i_{y(i)} - 1)^2 + \sum_{k=1, k \neq y(i)}^{K} (SF^i_k)^2] + \frac{1}{2}\lambda \sum_{k'=1}^{K} \sum_{h=0}^{H+1} (\alpha^{k'}_h)^2 \tag{4.5}$$

From the equation above, the cost function measures the Euclidean distance between the prediction and the ground truth labels. λ is the regularization parameter and is determined using the cross-validation approach. This optimization problem can be solved using the gradient descent approach. The corresponding derivatives with respect to elements of $\boldsymbol{\alpha}$ are defined in Equation (4.6), Equation (4.7), and Equation (4.8).

$$\frac{\partial J}{\partial \alpha^k_0} = \frac{1}{m} \sum_{i=1}^{T} \Delta^i_k + \lambda \alpha^k_0 \tag{4.6}$$

$$where\ \ \Delta^i_k = \begin{cases} -(SF^i_{y(i)} - 1)^2 SF^i_{y(i)} & if\ \ k = y(i) \\ (SF^i_k)^2 (1 - SF^i_k) & if\ \ k \neq y(i) \end{cases}$$

$$\frac{\partial J}{\partial \alpha^k_h} = \frac{1}{m} \sum_{i=1}^{m} \Omega^i_{h,k} + \lambda \alpha^k_h (1 \leq h \leq H) \tag{4.7}$$

$$where\ \Omega^i_{h,k} = \begin{cases} -(SF^i_{y(i)} - 1)^2 \cdot SF^i_{y(i)} \cdot SM^i_h & if\ \ k = y(i) \\ (SF^i_k)^2 \cdot (1 - SF^i_k) \cdot SM^i_h & if\ \ k \neq y(i) \end{cases}$$

$$\frac{\partial J}{\partial \alpha_{H+1}^{k}} = \frac{1}{m}\sum_{i=1}^{m}\Pi_k^i + \lambda \alpha_{(H+1)}^{k} \tag{4.8}$$

$$where\ \ \Pi_k^i = \begin{cases} -(SF_{y(i)}^i - 1)^2 \cdot SF_{y(i)}^i \cdot ST_{y(i)}^i & if\ k = y(i) \\ (SF_k^i)^2 \cdot (1 - SF_k^i) \cdot ST_k^i & if\ k \neq y(i) \end{cases}$$

It should be pointed out that the proposed multi-layer regression algorithm is different from the neural network classifier. In the neural network algorithm, the classification results are based on the neurons of the output layers. Those neurons receive the probability or classification scores from the neurons in the hidden layer. Therefore, they could not “see” the features originally input to the neural network. One of problems for neural network approach is the error propagation issue. In this work, all the classifiers are trained using the original features so there is no error propagation issue involved. The proposed work is also different from most of the boosting or bagging algorithms since all the classifiers are participating in the decision process and the proposed framework does not re-train models for those misclassified instances.

4.3 Experiments and Results

4.3.1 Data Sets and Data Preprocessing

In order to evaluate our proposed framework experimentally, three data sets were selected from the the BDGP ISH image database [12] and IICBU database [184]. We randomly selected 1,475 lateral view images and 2,000 dorsal view images from the BDGP database to form the Lateral View data set (D1) and Dorsal View data set (D2).

In addition, the terminal aging data set (T1) was selected from the IICBU database, as it proves to be the most challenging problem among all the classification tasks. The distribution of the data instances corresponding to class labels are shown in Table 4.1 and Table 4.2. The evaluation criterion used in this work was the classification accuracy, which is the ratio of the correctly classified instances to the total number of instances. Three-fold cross validation was adopted to evaluate the performance of the framework. The sample images from D1, D2, and T1 data sets are shown in Figure 4.11.

Table 4.1: The distribution of instances among classes in D1 and D2

Class Label	**Meaning**	**No. of Images (D1)**	**No. of Images (D2)**
Class 1	Stages 1-3	187	9
Class 2	Stages 4-6	270	142
Class 3	Stages 7-8	146	197
Class 4	Stages 9-10	145	263
Class 5	Stages 11-12	380	422
Class 6	Stages 13-16	347	967
Total		1475	2000

Table 4.2: The distribution of instances among classes in T1

Class Label	**Meaning**	**No. of Images (T1)**
Class 1	day 0	106
Class 2	day 2	218
Class 3	day 4	159
Class 4	day 6	176
Class 5	day 8	195
Class 6	day 10	62
Class 7	day 12	54
Total		970

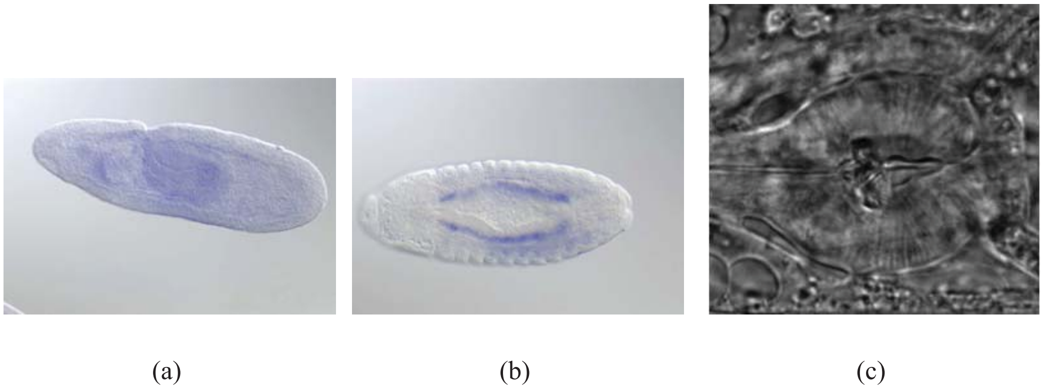


Figure 4.11: Sample images from the three data sets used in experiments (a) A sample image from D1 (b) A sample image from D2 (c) A sample image from T1

As introduced in Section 4.2.1, the images that contain clear objects were preprocessed first. As shown in the sample images, such was the case for the images in D1 and D2. Therefore, a normalization and segmentation operation for D1 and D2 were carried out. We first used the Otus method [185] to compute the threshold for segmenting the embryo from the background. In addition, principal component analysis (PCA) was applied on the binary images generated based on the embryo region so that the main axis was identified. In addition, we unified the orientation of the embryos by rotating the main axis to the horizontal position and made the anterior side on the left and the posterior side on the right. A minimum bounding rectangle was then computed for each embryo image and the region in that bounding rectangle was segmented. In order to ensure the quality, each image was manually checked after segmentation. All the segmented images were resized to 1200x480 pixels. Figure 4.12 shows an example of the image after preprocessing compared with the raw image. For the images in T1, no preprocessing operation was conducted.

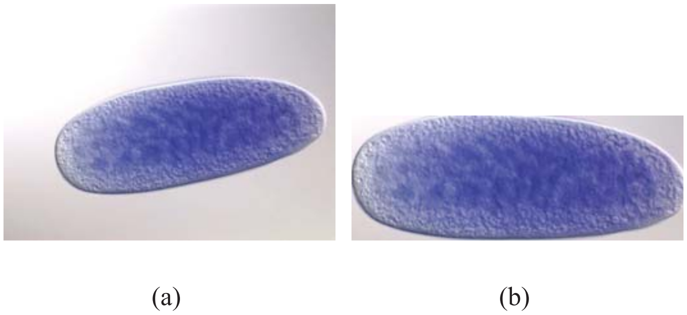


Figure 4.12: The comparison between a raw image and the corresponding segmented object (a) The raw image (b) The image of the object after segmentation

4.3.2 Features and Feature Selection Results

Since there are no visual features provided for D1 and D2, we extracted 640 features, which are introduced in Section 4.2.2, for each image in D1 and D2 data sets. In order to compare the T1 data set with the existing work of the Wndchrm framework [119], the same set of 2,873 features were extracted using the feature extraction facility provided by Wndchrm applications.

Feature selection is an important step in a classification framework, as it determines the overall classification performance. In practice, the number of features to retain is usually difficult to decide. In this dissertation, we performed an empirical study which provides some insights into feature selection. As is mentioned in Section 4.2.2, the chi-square feature ranking algorithm [26] was applied to get the ranking list of the features according to their chi-square statistic values with respect to the class with the most significant features at the top. Next, starting from the empty set, one feature was added to the selected feature set sequentially each time, according to its rank in the aforementioned ranking list and the three-fold cross-validation classification result was generated based on the current selected feature set. Figure 4.13 shows the plot of classification accuracy according to the number of features in the selected feature set using D1. A circle represents a data point and indicates the average classification accuracy of three-fold cross-validation using a certain number of features and the red curve is drawn by fitting each data point using a polynomial of degree 4. It can be seen from the figure that with more and more features added to the selected feature set, the classification accuracy increases while the increasing rate is decreasing, since the features added later have relatively low ranking scores compared with the features added before. In other words, the marginal profit of adding more features is actually decreasing with the increase of the number of features in the selected feature set. When the number of features passes around 200, the gain of adding more features becomes less and less, which indicates the top ranked one-third of all features play more than 90% of the role in classification. This empirical study provides some insights for real-world applications. For example, for a real-time classification system that is under strict time constraints, a similar study could be carried out off-line to decide which features to keep in order to strike the best balance between processing time and accuracy. In this dissertation, 570, 586, and 1,602 features corresponding to the highest classification accuracies were selected for D1, D2, and T1, respectively.

4.3.3 Significance of Adding Middle Classes

The main contribution of this work is to create the middle classes, which are relatively easy to categorize, and to integrate outputs from different layers of classifiers. Therefore, our main assumption is that the middle classes are usually easy to classify. To evaluate the assumption, the classification accuracies of middle classes are computed and compared with those of the end classes. In this section, the supervised middle cre-

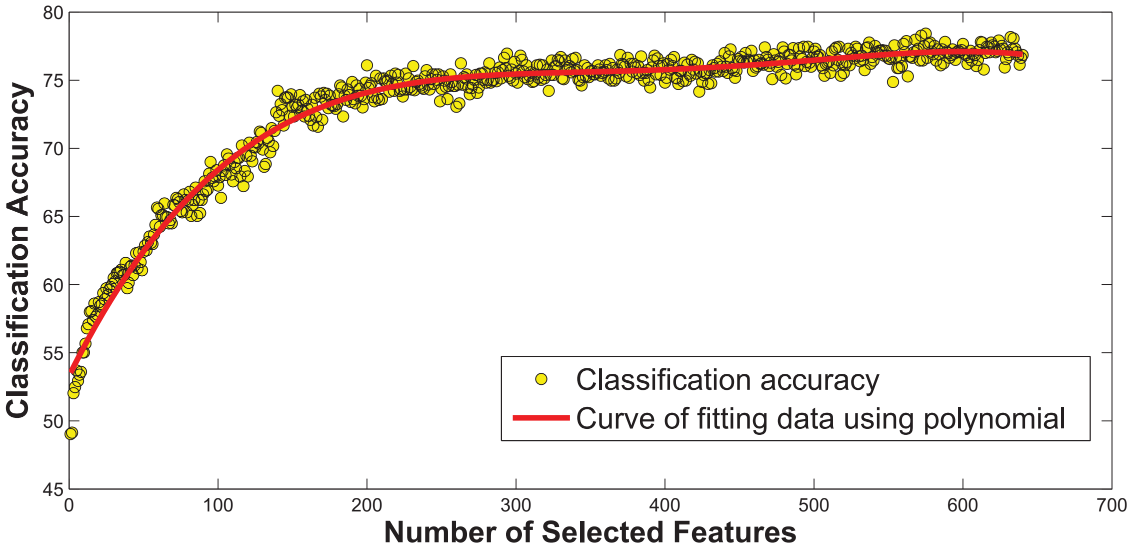


Figure 4.13: The classification accuracy vs. the number of selected features

ation algorithm is applied. Table 4.3, Table 4.4, and Table 4.5 show the comparison of accuracies of the end class classification and middle class classification in D1, D2, and T1 correspondingly. In all the tables, the column "ID" is the identification number for the middle class creation scheme, which is used later in the chapter. The column "Group" shows the grouping of the end classes into the middle classes. Specifically, the end class labels in one pair of curly brackets are grouped into one middle class. The "M_Accuracy" column shows the classification accuracy of middle classes. The "T_Accuracy" column shows the classification accuracy of end classes without integrating any information from middle layer classifiers. It should be pointed out that "D1_5," "D2_5," and "T1_5" are generated using the algorithm introduced in Section 4.2.3. Others are generated arbitrarily. It is also noticed that the sequence of the end classes is not changed in the grouping process in order to match the temporal relationship of end classes in this application. It could be seen from both tables that the accuracy of the binary or ternary middle class classification task is higher than that of the end class classification task, which matches our assumption.

Table 4.3: The comparison of middle class classification and end class classification of D1

ID	Group	M_Accuracy	T_Accuracy
D1_1	{1,2,3};{4,5,6}	90.17%	77.32%
D1_2	{1,2};{3,4,5,6}	89.49%	77.32%
D1_3	{1,2,3,4};{5,6}	90.85%	77.32%
D1_4	{1,2};{3,4};{5,6}	87.73%	77.32%
D1_5	{1};{2,3,4,5};{6}	89.50%	77.32%

Table 4.4: The comparison of middle class classification and end class classification of D2

ID	Grouping	M_Accuracy	T_Accuracy
D2_1	{1,2,3};{4,5,6}	93.00%	79.25%
D2_2	{1,2};{3,4,5,6}	96.00%	79.25%
D2_3	{1,2,3,4};{5,6}	90.00%	79.25%
D2_4	{1,2};{3,4};{5,6}	88.75%	79.25%
D2_5	{1,2,3,4};{5};{6}	87.50%	79.25%

The next step is to see whether the integration of information output from middle layer classifiers helps end class classification. Table 4.6, Table 4.7, and Table 4.8 show the comparison of performance among different frameworks. In all three tables, the first column indicates the data set used, the second column shows the different frameworks proposed, and the third column is the corresponding end class classification accuracy. Using Table 4.6 as a specific example, "No middle class" means the framework does not build any middle layer classifiers, while "Applying D1_1" indicates using "D1_1" in Table 4.3 to generate the middle classes and applying the multi-class regression algorithm. "T'_Accuracy" indicates the corresponding end class classification accuracy.

Table 4.5: The comparison of middle class classification and end class classification of T1

ID	Grouping	M_Accuracy	T_Accuracy
T1_1	{1,2,3};{4,5,6,7}	84.49%	62.86%
T1_2	{1,2};{3,4,5,6,7}	83.26%	62.86%
T1_3	{1,2,3,4};{5,6,7}	83.67%	62.86%
T1_4	{1,2};{3,4};{5,6,7}	78.78%	62.86%
T1_5	{1,2,3,4};{5,6};{7}	80.82%	62.86%

It could be observed from the table that the middle class helps improve the end class classification accuracies, which proves that the middle layer classifiers could help the end class classification. In addition, it could be observed that the improvement is also dependent on the grouping scheme for creating the middle classes, which shows the importance of a proper strategy to create middle classes.

Table 4.6: The comparison of different frameworks for D1

Data Set	Framework	T'_Accuracy
D1	**No middle class**	77.32%
D1	**Applying D1_1**	81.00%
D1	**Applying D1_2**	80.75%
D1	**Applying D1_3**	80.33%
D1	**Applying D1_4**	80.00%
D1	**Applying D1_5**	80.66%

4.3.4 Comparison with Other Classification Algorithms

In order to better evaluate the framework, the comparisons were made with other multi-class classification algorithms using the same selected feature set. Table 4.9 shows the performance of different classification algorithms for D1, D2, and T1. "Proposed" denotes the proposed subspace-based classification framework with multi-class regres-

Table 4.7: The comparison of different frameworks for D2

Data Set	**Framework**	**T'_Accuracy**
D2	**No middle class**	79.25%
D2	**Applying D2_1**	81.50%
D2	**Applying D2_2**	81.75%
D2	**Applying D2_3**	81.75%
D2	**Applying D2_4**	82.50%
D2	**Applying D2_5**	82.25%

Table 4.8: The comparison of different frameworks for T1

Data Set	**Framework**	**T'_Accuracy**
T1	**No middle class**	62.86%
T1	**Applying T1_1**	67.34%
T1	**Applying T1_2**	65.31%
T1	**Applying T1_3**	64.08%
T1	**Applying T1_4**	66.53%
T1	**Applying T1_5**	69.86%

sion. The other classifiers used are from Weka [26] where "J48" is the C4.5 decision tree classifier with error reduce pruning; "BN" denotes Bayes network; "NN" indicates the nearest neighbor classifier; "KNN*X*" stands for the *X* nearest neighbors; and "RF*X*" is the Random Forest with *X* trees. "LSVM+L" and "LSVM+R" are LibSVM [186] with linear kernel and RBF kernel correspondingly. "ADA.+LSVM*X*" Adaboost algorithm with LibSVM and *X* denotes the number of iterations. "NeuroNW" indicates the neuro network algorithm. It can be seen that the proposed algorithm outperforms the other classifiers for all three data sets. For other classifiers, the neuro network algorithm gives the second best performance, which indicates that the multi-layer classification is suitable for the classification task. However, during experiments we find the performance is highly sensitive to the parameter of the number of hidden layers and difference could be as large as 40%-50% in terms of accuracy. Accordingly, this raises another research topic of how to choose the number of hidden layers. In addition, the backpropagation approach suffers from the time complexity issue. For D1 and D2, the LibSVM with linear kernel performs relatively well. Interestingly, the LibSVM with RBF kernel, which is a non-linear kernel, performs much worse. That indicates the data favor the linear classification decision boundary rather than the high-degree decision boundary. This matches the observation in [187]. The Adaboost algorithm improves the classification results of LibSVM slightly but the computation time increases by 7 to 8 times. The nearest neighbor approach gives relatively inferior performance, which indicates that the training data contain a significant amount of noise. In our proposed subspace-based model, the data in each class are first pruned to remove noise, which is one of the reasons why our proposed algorithm performs better. Random Forest algorithm builds a set of decision trees and is a relatively popular ensemble learning algorithm. However, it also suffers from the noisy data issue.

4.3.5 Comparison with Other Existing Frameworks

The previous work on the same data sets was implemented to evaluate our current framework. For the data sets D1 and D2, the following two frameworks were implemented. In the first work [188], the original image was divided into 640 blocks of 8×8 pixels. The Log Gabor features were extracted from each block and formed a 24-dimension vector. Therefore, one original image was represented using a 15,360-dimensional vector. The vector was then projected to a lower dimensional space to form a 128-dimensional vector using the modified version of LDA. The nearest neighbor algorithm was then utilized to perform classification. In the second work [187], the authors followed the relatively similar path while picking the significant blocks of the

Table 4.9: The comparison of the proposed framework with other classification algorithms

Classification Algorithm	D1	D2	T1
Proposed	**80.66%**	**82.25%**	**69.86%**
J48	53.22%	52.00%	43.67%
BN	46.44%	58.50%	50.61%
NN	53.56%	59.50%	53.06%
KNN10	57.29%	56.75%	59.59%
KNN20	52.88%	54.50%	57.96%
KNN30	54.24%	53.75%	59.18%
RF10	59.32%	61.50%	49.80%
RF20	64.07%	66.25%	53.88%
RF30	64.75%	66.50%	53.88%
RF40	65.08%	66.75%	53.06%
RF50	67.46%	66.75%	55.92%
RF60	66.78%	66.25%	55.92%
LSVM+L	76.98%	78.75%	40.82%
LSVM+R	58.31%	51.00%	22.45%
ADA.+LSVM20	78.30%	79.75%	41.22%
ADA.+LSVM50	78.30%	79.75%	41.22%
NeuroNW	78.33%	80.21%	60.41%

embryos based on human observation as the starting point for feature extraction; the Gabor features were used and the LibSVM was utilized as the classifier. The classification results are shown in Table 4.10. “RULDA” and “Gabor+LibSVM” represent the approaches in [188] and [187], respectively. The significant areas in [188] were just selected for lateral view images, and the classification result is not available for D2. It can be seen that the proposed framework outperforms other existing frameworks in terms of classification accuracy on both data sets. The significant blocks selected in [188] were specifically to differentiate the ISH images from stage 3 to stage 6 and were difficult to adapt to the tasks for classifying all the stage ranges. In terms of RULDA, the high-dimensional feature vector poses challenges for computation.

For the data set T1, the Wndchrm framework introduced in [119] was deployed for comparison. For this data set, the same set of features was extracted for our framework and the comparison framework. The comparison results are shown in Table 4.11. In [119], the reported cross-validation accuracy for the same data set was 49%. In our experiments, we obtained 47.53% as the three-fold cross-validation result, which indicates that our implementation was successful. It can be seen that the proposed framework could outperform the comparison frameworks by a relatively large margin.

Table 4.10: The comparison of the proposed framework with existing frameworks for D1 and D2

Framework	**D1**	**D2**
The proposed	**80.66%**	**82.25%**
RULDA	75.25%	74.75%
Gabor+LibSVM	65.29%	NA

Table 4.11: The comparison of the proposed framework with existing frameworks for T1

Framework	**T1**
The proposed	**69.86%**
Wndchrm	47.53%

4.3.6 Evaluation of the Unsupervised Middle Class Creation Algorithm

The unsupervised solution is evaluated in this section. The k-means algorithm [26] was utilized in this work. The k value was determined by empirical study for each data set. The comparisons of accuracies of using supervised and unsupervised solution are given in Table 4.12, Table 4.13, and Table 4.14, for $D1$, $D2$, and $T1$ respectively.

Table 4.12: The comparison of accuracies for D1

Algorithms	Accuracy
Supervised	80.66%
Unsupervised	87.34%

Table 4.13: The comparison of accuracies for D2

Algorithms	Accuracy
Supervised	82.25%
Unsupervised	87.66%

It could be seen from the tables above that the proposed unsupervised solutions outperform the supervised solution, which indicates the information captured in the clusters provides help to the classification task.

4.4 Conclusion

In summary, the AAN-based multi-layer model collaboration framework for annotating biological temporal stages is proposed in this chapter. Given the internal sequential relationships among different classes, a prudent and effective grouping strategy is designed to create middle classes to assist end class classification. In addition, an unsupervised clustering based middle class creation strategy is also proposed. By defining a suitable optimization objective and providing a numerical solution, a novel multi-layer model ccollaboration algorithm is given to overcome the notorious error propagation issue in the multi-layer classification model. In addition, the empirical study of feature selection provides insights on how to keep a proper portion of the selected features in a practical case. As the detailed results shown in the experimental analysis section, the proposed framework consistently outperforms other classic multi-class classification algorithms

Table 4.14: The comparison of accuracies for T1

Algorithms	Accuracy
Supervised	69.86%
Unsupervised	78.78%

in Weka and several state-of-the-art frameworks for solving the same problems. It indicates that the proposed framework provides a promising and robust solution to the biological temporal stage annotation problem.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In recent years, there has been a multitude of multimedia big data coming from different domains ranging from entertainment and security to biomedical fields. These data play important roles in everyday life. The automatic annotation or labeling of data instances in multimedia databases has become an important research task. The machine learning-based approach treats each label as a class and converts the problem to a classification problem. Most traditional classification algorithms process each class individually and lose the information about the connection among different classes. In this dissertation, an AAN is designed to model the relationship among different classes and help to improve the classification accuracy. The highlights of this dissertation could be summarized as follows.

- A five-step pipeline of building the AAN is designed. The general steps are

  1) "Node Link Modeling," which identifies the entities in a specific application and models them using the network model;

  2) "Link Selection," which selects the significant links based on domain knowledge or automatic approaches;

  3) "Score Normalization," which makes the scores from different input models comparable;

4) “Weight Computation,” which computes weights for all selected links; and

5) “Score Integration,” which summarizes information from different nodes and draws a final conclusion. The whole process takes the advantage of the output from different nodes and refines the final output results.

- The association rule mining-based link selection module is proposed to select the most significant association among different nodes. As shown in Chapter 3, this step affects the performance of the overall framework significantly. The association rule mining is a data mining technique that finds the significant association patterns from a transaction data set. Inspired by the idea of the association rule mining, an automatic link selection module is proposed to filter the insignificant links between nodes. Two selection criteria, the support ratio and the interest ratio, are proposed and formulated.

- The Pearson product-moment correlation coefficient-based related concepts filtering module is proposed to consider the model accuracy. It is common that different models have different accuracies. The noisy input from a relatively inaccurate model could affect the output adversely. Therefore, this module is added to take the accuracy of models into consideration.

- A Bayesian posterior probabilistic score estimation approach is also proposed. Different models may output scores with distinct ranges and characteristics, which makes it difficult to integrate them together directly. This module normalizes all the scores into the range between 0 and 1.

- Different models of score integration are utilized to enhance the correctness and smoothness of the initial predictions. The logistic regression posterior probability estimation is proposed to integrate the probabilistic scores and weights together. In addition, inspired by the collaborative filtering approach, the multi-model collaboration approach utilizes the output scores and labels directly, which could improve the classification accuracy. To the best of our knowledge, this is the first work to apply the collaborative filtering approach for score integration in the multimedia data mining field.

- In the multi-class classification framework, a closed-form cost function is defined. By minimizing this cost function using the gradient descent algorithm, a set of optimized parameters is computed to model both the positive and negative relationships simultaneously. In addition, an unsupervised middle class creation strategy is proposed and shows promising results.

- The negative correlation, which has rarely been studied, is mined from data sets and utilized in current framework. Experimental results indicate it is also helpful in improving classification and retrieval performance.

In sum, the proposed AAN provides a systematic solution to utilize label correlations to improve the performance of a single model. Experimental results indicate the proposed framework could help improve multi-label classification as well as multi-class classification accuracy. The success of its application in the multimedia semantic concept detection and biomedical temporal pattern annotation suggests it is a promising framework to integrate information from different models.

5.2 Future Work

On the basis of the current framework and experimental results, several future research directions are proposed and introduced as follows.

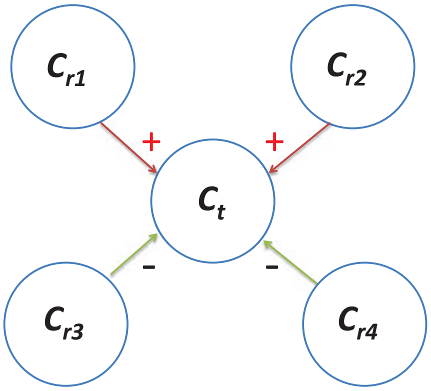


Figure 5.1: The first fusion approach

5.2.1 Integrating Negative Correlations with Positive Correlations in AAN for Multimedia Concept Detection

In the existing framework, positive and negative correlations are studied separately. A natural extension is to investigate the possibility of integrating positive and negative correlations. To achieve this goal, two directions are proposed to be explored. The first direction is to integrate positive correlations and negative correlations directly and form a single AAN. This idea is illustrated in Figure 5.1. In this example, for one target concept C_t, C_{r1} and C_{r2} are two positively correlated concepts, while C_{r3} and C_{r4} are two negatively correlated concepts. In order to compute the final score for a target concept, positive correlations and negative correlations are directly integrated.

Formally, for the C_t, let C_1, C_2, ... C_p represent p positively correlated concepts and C_{p+1}, C_{p+2}, ... C_{p+q} represent q negatively correlated concepts. S_1, S_2, ..., S_{p+q} are output scores from primary concept detectors. In addition, S_t is the detection score for C_t. Following the same notation in Section 3.3.2, a score matrix $\boldsymbol{S}$ is formed and each row of this matrix ($\boldsymbol{S}^i$) is a vector $[1, S_t^i, S_1^i, S_2^i, ..., S_{p+q}^i]$ for the instance i ($1 \leq i \leq m$), where m is the total number of instances. A weight vector $\boldsymbol{\theta}$ is the column vector $[\theta_0, \theta_1, \theta_2, ..., \theta_{p+q+2}]^T$. Let y^i represent the label for the instance i. λ is the regularization parameter. A cost function can be formulated as follows:

$$\begin{aligned} J(\boldsymbol{S};\boldsymbol{\theta}) \quad &= -\log L(\boldsymbol{S};\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_2, \qquad (5.1)\\ \text{subject to} \quad & \theta_1, \theta_2, ..., \theta_{p+2} \geq 0,\\ & \theta_{p+3}, \theta_{p+4}, ..., \theta_{p+q+2} \leq 0.\\ \text{where} \quad L(\boldsymbol{S};\boldsymbol{\theta}) \quad &= \prod_{i=1}^{m} (g(\boldsymbol{S}^i\boldsymbol{\theta}))^{y^i} \cdot (1 - g(\boldsymbol{S}^i\boldsymbol{\theta}))^{(1-y^i)}, \qquad (5.2)\\ \text{and } g(x) \quad &= \frac{1}{1+e^{(-x)}} \qquad (5.3) \end{aligned}$$

The cost function can be solved by numerical approaches. To refine the model, MCA-based weights can be added to the vector. One disadvantage of this approach is that when the number of related concepts increases, there are more constraints added to the optimization problem, which makes the optimization problem more difficult to solve. Therefore, another direction is to further filter the related concepts for each target concept.

At the same time, another fusion framework can also be explored. The idea is illustrated in Figure 5.2. In this framework, a positive AAN and a negative AAN are trained separately using the proposed framework. For each target concept, the output scores from the two networks are combined together. Assume for a target concept C_t and a data instance i, the output scores from the positive AAN and negative AAN are S^i_{pos} and S^i_{neg}. Assume the final score for the instance i is S^i_F. There can be different ways of combining scores. The first approach is the linear combination approach. The most intuitive solution is adding the two scores together, which is similar to the approach utilized in a recently published paper in IEEE Transaction on Multimedia [189]. This

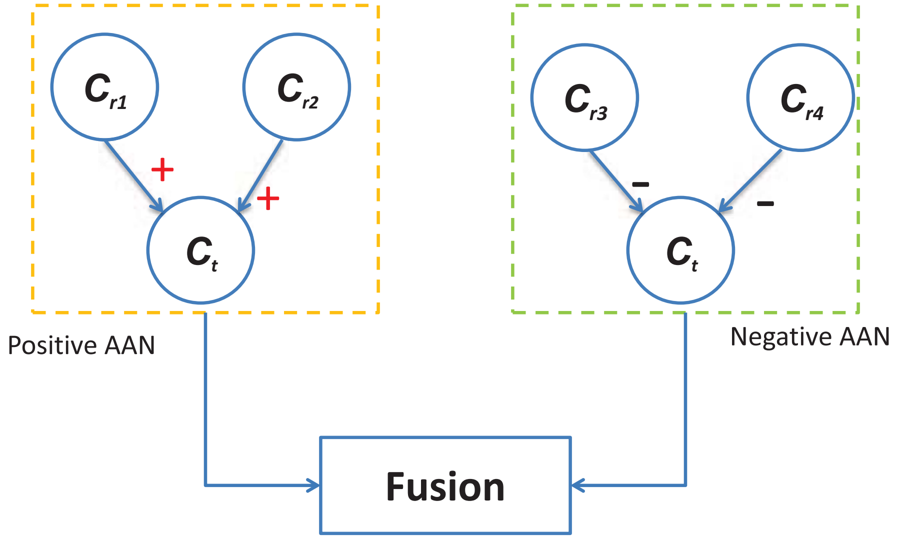


Figure 5.2: The second fusion approach

intuitive solution seems to give reasonable performance in that paper. This solution can be generalized to the following form:

$$S_F^i = W_1 * S_{pos}^i + W_2 * S_{neg}^i \tag{5.4}$$

Here, W_1 and W_2 are two weights. The weights can incorporate different prior knowledge from domain experts or data mining researchers. For example, one weight determination strategy proposed in literature [190] incorporating accuracies of models is given in Equation (5.5).

$$W_1 = 1 - Err(S_{pos}) \tag{5.5}$$

Here, the "Err" indicates the error rate of the concept detection model for S_{pos}. However, based on the experimental results from Section 3.3.3, the linear combination might give marginal improvement. More studies need to be carried out in this direction. It is noticed that the solution of using the minimum value or the maximum value actually is a special case of this approach.

The second direction is the Bayesian conditional probabilistic fusion model. Using this model, a ratio score (P_r) can be computed. Let C_t^1 indicates that an instance is positive and C_t^0 indicates it is negative. $p(x)$ indicates the probability density function for a random variable x. Then P_r can be formulated as follows.

$$P_r = \frac{p(C_t^1|S_{pos}^i, S_{neg}^i)}{p(C_t^0|S_{pos}^i, S_{neg}^i)} \tag{5.6}$$

As shown in the equation above, P_r can be used as a new score. If we assume the inference processes from the positive AAN and the negative AAN are independent, using the Bayesian rule, Equation (5.7) can be further expanded and simplified as follows.

$$\begin{aligned} Pr = & \frac{p(S_{pos}^i|C_t^1)p(S_{neg}^i|C_t^1)p(C_t^1)/p(S_{pos}^i, S_{neg}^i)}{p(S_{pos}^i|C_t^0)p(S_{neg}^i|C_t^0)p(C_t^0)/p(S_{pos}^i, S_{neg}^i)} \\ Pr = & \frac{p(S_{pos}^i|C_t^1)}{p(S_{pos}^i|C_t^0)} \frac{p(S_{neg}^i|C_t^1)}{p(S_{neg}^i|C_t^0)} \frac{p(C_t^1)}{p(C_t^0)} \end{aligned} \tag{5.7}$$

It is clear that the ratio consists of three factors. One observation is that the prior knowledge of $p(C_t^1)$ and $p(C_t^0)$ can be incorporated in the score integration. The other conditional probabilities can be estimated using a non-parametric probability density estimation approach such as the Parzen window density estimation or parametric probability density estimation, which raises another question of what distribution model to assume. The Gaussian probability model can be the initial attempt to fit all distributions. The maximum likelihood estimation can be utilized to estimate parameters.

The third direction is to train a meta-model over both scores. In this direction, another classification model is trained utilizing S^i_{neg} and S^i_{pos}. The probabilistic output of the new model is used as the final output. One classification model can be the support vector machine. However, one concern is that S^i_{neg} and S^i_{pos} tend to be small since they were fit to zero and one labels. Another round of model training could face challenges numerically.

As a matter of fact, combining output from different models is an active research area since it usually helps to reduce variances of predictions in practice. This research task itself deserves much deeper exploration.

5.2.2 Incorporating Multi-Model Co-Optimization for Multimedia Concept Detection

So far we focus on binary links between a related concept and a target concept. The co-distribution of all scores is not modeled explicitly. In order to address this issue, a multi-model co-optimization framework is proposed. This framework is an extension from the multi-layer collaboration framework introduced in Chapter 4. A schematic example is given in Figure 5.3 (only positive links are considered for now). In this framework, the selected related concept models and target concept model are first transformed to form a two-layer structure. In this structure, the nodes marked C_{r1}, C_{r2} and C_t indicate original classification models, and M_{r1}, M_{r2} and M_{rt} indicate a new level of integration models. Each model receives output scores from the original models. The final output is from M_t. In order to train such a model, an optimization problem is formulated as follows: suppose for a data instance i ($1 \leq i \leq m$, m is total number of instances), a target is represented as C_0 for simplicity of notation. The p positively correlated concepts are represented as $C_1, C_2, ..., C_p$; suppose the output score for a concept C_x for the instance

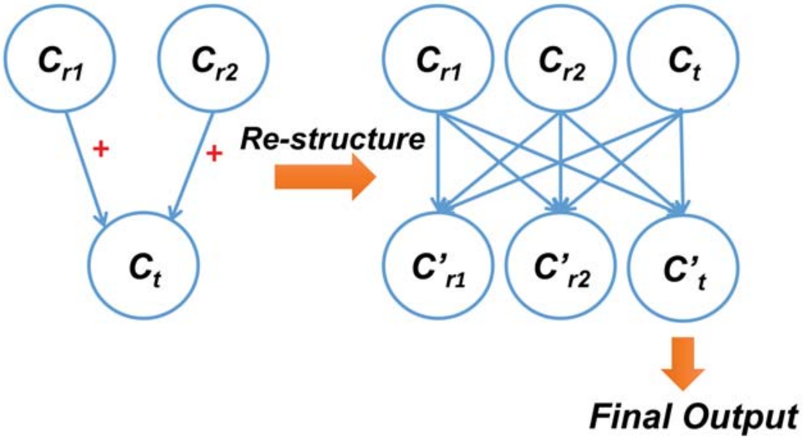


Figure 5.3: A sample model for multi-model co-optimization

i is represented as S_x^i; In order to simplify the computation in the process, a matrix $\boldsymbol{S}$ is formed as follows.

$$\begin{bmatrix} S_0^1 & S_1^1 & \dots & S_p^1 \\ S_0^2 & S_1^2 & \dots & S_p^2 \\ \dots & \dots & \dots & \dots \\ S_0^i & S_1^i & \dots & S_p^i \\ \dots & \dots & \dots & \dots \\ S_0^m & S_1^m & \dots & S_p^m \end{bmatrix}$$

, assume $\boldsymbol{L}$ is a matrix of which each row $[L_0^i, L_1^i, ..., L_p^i]$ represents labels for concept C_0, C_1, ... C_p for one instance i. Assume $\boldsymbol{\alpha}$ is a square matrix which has a dimension of $(p+1)$ by $(p+1)$. For simplicity of notation, the matrix is indexed from zero. Let each element α_{uv} $(0 \leq u, v \leq p)$ represent the weight from concept detector model C_u

to the integration model M_v. If $\boldsymbol{g}(\boldsymbol{X})$ indicates the matrix of which each element equals to the result of applying the sigmoid function on the corresponding element in matrix $\boldsymbol{X}$, a cost function that considers overall errors can be defined as follows

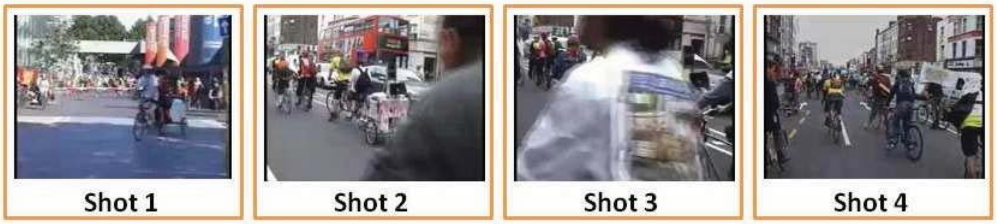
$$J = Tr((\boldsymbol{g}(\boldsymbol{S\alpha}) - \boldsymbol{L})(\boldsymbol{g}(\boldsymbol{S\alpha}) - \boldsymbol{L})^T) + \lambda Tr(\boldsymbol{\alpha}^T \boldsymbol{\alpha}) \tag{5.8}$$

λ is the regularization parameter. $Tr(\boldsymbol{X})$ indicates the trace of matrix $\boldsymbol{X}$. The parameters can be trained using numerical approaches. λ can be determined using the cross-validation approach. After α is learned, the final score can be computed accordingly.

There are two potential challenges of this framework. First, there are few positive training examples. Second, the number of parameters in the parameter matrix $\boldsymbol{\alpha}$ is a quadratic function of the number of related concepts. Therefore, the time complexity issue needs to be addressed. One approach to handle this issue might be using the LASSO-based approach by adding the L1 penalty. More future work can be explored in this direction.

5.2.3 Incorporating Temporal Correlations in AAN for Video Concept Detection

The existing work utilizes the correlations of different concepts in the same image or video shot. This is the spatial dimension of the correlation between concepts. For video data sets, besides the spatial dimension, the temporal dimension could also be taken into consideration. In other words, the spatial context describes concept relationships within a single shot, while the temporal context describes the dependency between temporally continuous shots. Compared with research in exploiting the spatial context, relatively less effort is put on exploiting the temporal relationship. In this research direction, we are going to integrate the temporal relationship into the AAN.



Concepts	Shot 1	Shot 2	Shot 3	Shot 4
Bike	1	1	1	1
People	1	1	1	1
Urban	1	1	1	1
Car	0	1	1	0

Figure 5.4: An example of temporal semantic consistency within a video

It is observed that the video data exhibit strong consistency along the temporal domain, which ensures the footage is visually smooth and semantically coherent. The temporal consistency refers to the observation that temporally adjacent video shots have similar visual and semantic contents. This implies that relevant shots that contain the same semantic concept tend to gather in temporal neighborhoods or even appear next to each other consecutively. Figure 5.4 shows an example of the semantic consistency. It could be observed that some concepts such as “bike” and “people” exist in every shot.

Given this observation, we plan to integrate temporal correlations into the current AAN framework. First, we focus on the correlations for the same concept in consecutive shots. Using the Bayesian conditional probability model, for one target concept C, the probability that the concept is positive in shot s is represented as $p(C^{+},(s))$; if this probability is related to the probability in the previous shot, it can be formulated as:

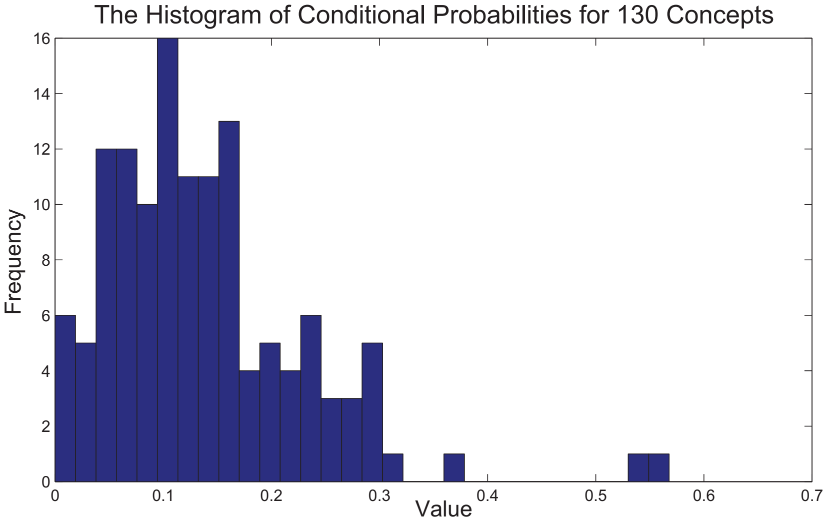


Figure 5.5: The distribution of conditional probabilities of different concepts

$$
\begin{aligned}
p(C^{+}(s)) &= p(C^{+}(s)|C^{+}(s-1))p(C^{+}(s-1)) + p(C^{+}(s)|C^{-}(s-1))p(C^{-}(s-1)) \qquad (5.9)\\
&= p(C^{+}(s)|C^{-}(s-1)) \\
&+ p(C^{+}(s-1))(p(C^{+}(s)|C^{+}(s-1)) - p(C^{+}(s)|C^{-}(s-1))) \qquad (5.10)
\end{aligned}
$$

Here, the $p(C^{+}(s-1))$ can be estimated from the output score from concept detector for C. Crude models for estimating $p(C^{+}(s)|C^{-}(s-1))$ and $p(C^{+}(s)|C^{+}(s-1))$ are to estimate these probabilities from training labels. Figure 5.5 shows the histogram of the values $p(C^{+}(s)|C^{+}(s-1))$ for 130 concepts estimated from training labels in the IACC.1.A data set. As shown in the figure, some concepts show relatively high values and could be targeted at first.

The aforementioned model is relatively coarse. More information can be incorporated by considering the feature values for shots $(s-1)$ and s. Incorporating feature information can be a double-edged sword, which introduces useful information and noises at the same time. In addition, a more advanced probabilistic graphic model such as the CRF model can be applied here, while the proper selection of models and efficient approaches of training models need to be taken into account. The next step is to mine temporal correlations of different concepts.

In summary, different approaches integrating positive and negative correlations, high-order correlations and temporal correlations are from different perspectives of modeling the inter-concept correlations. Probabilistic-based models and regression-based approaches are promising directions to pursue. It is expected with the joint efforts from data mining, machine learning and statistics researchers, multi-class and multi-label classification models will be improved in terms of both accuracy and efficiency.

Appendix A

The Top 50 Positive Correlations Selected in the IACC.1.A Data Set

Table A.1: The top 50 positive correlations selected in the IACC.1.A data set

Concept 1	Concept 2	Normalized Weight
Flowers	Vegetation	100
Airplane_Flying	Airplane	100
Dogs	Animal	100
Horse	Animal	100
Beach	Outdoor	100
Sitting_Down	Person	100
Teenagers	Person	100
Dark-skinned_People	Person	100
Anchorperson	Person	100
Reporters	Person	100
Flowers	Plant	100
Government-Leader	Politicians	100
Plant	Vegetation	99.9

Continued on Next Page…

Concept 1	Concept 2	Normalized Weight
Ground_Vehicles	Vehicle	99.7
Female-Human-Face-Closeup	Face	99.7
Car	Vehicle	99.6
Car	Ground_Vehicles	99.6
Female-Human-Face-Closeup	Female_Person	99.5
Walking	Walking_Running	99.4
Girl	Female_Person	99.3
Politicians	Politics	98.9
News_Studio	Indoor	98.8
Scientists	Science_Technology	91.6
Anchorperson	News_Studio	74.6
Anchorperson	Reporters	71
Beach	Waterscape_Waterfront	69.7
Stadium	Athlete	69.7
Reporters	Indoor	66.8
Cityscape	Building	63.9
Teenagers	Face	63.4
Vegetation	Plant	62.1
Road	Streets	59.9
Basketball	Throwing	59.4
Reporters	News_Studio	59.4
Streets	Road	59.3

Continued on Next Page…

Concept 1	Concept 2	Normalized Weight
Bicycles	Bicycling	57.9
Anchorperson	Face	57.9
Single_Person	Face	57.7
Basketball	Stadium	56.3
Demonstration_Or_Protest	Crowd	56
Female_Person	Face	55.9
News_Studio	Reporters	55.1
Soccer_Player	Stadium	53.7
Beach	Sky	52.1
Instrumental_Musician	Entertainment	51.3
Teenagers	Female_Person	51.2
Mountain	Sky	51.1
Meeting	Crowd	50
People_Marching	Crowd	49.6
Cityscape	Suburban	49.3

Appendix B

The Top 50 Positive Correlations Selected in the IACC.1.B Data Set

Table B.1: The top 50 positive correlations selected in the IACC.1.B data set

Concept 1	**Concept 2**	**Weight**
Person	News_Studio	100
City	Cityscape	97.6
Airplane_Flying	Airplane_Takeoff	96.7
Airplane	Airplane_Takeoff	96.7
Vehicle	Airplane_Takeoff	96.6
Outdoor	Islands	95.9
Person	News	92.5
Person	Election_Campaign_Address	91.7
Sports	Throw_Ball	89.8
Outdoor	House_Of_Worship	88.5
Vehicle	Military_Aircraft	87.7
Person	Politics	85.7

Continued on Next Page. . .

Concept 1	Concept 2	Weight
Indoor	Hockey	85.6
Eukaryotic_Organism	Vegetation	84.8
Person	Football	83.8
Person	Suits	82.7
Mammal	Quadruped	81.3
Skating	Hockey	80.8
Sports	Throwing	80.2
Indoor_Sports_Venue	Hockey	79.5
Person	Election_Campaign	79.1
Outdoor	Windows	78.2
Building	House_Of_Worship	78.1
Person	Election_Campaign_Debate	76.5
Airplane	Military_Aircraft	76.1
Snow	Ski	76.1
Person	Stadium	75.7
Indoor	Studio_With_Anchorperson	75.3
Outdoor	Apartment_Complex	74.8
Mammal	Herbivore	73.8
Sports	Athlete	72.8
Outdoor	Apartments	72.5
Cattle	Cows	72
Anchorperson	Male_Reporter	71.4

Continued on Next Page…

Concept 1	Concept 2	Weight
Muslims	Religious_Figures	71.3
Herbivore	Sea_Mammal	70.5
Waterscape_Waterfront	Islands	70.4
Plant	Vegetation	69.8
Quadruped	Herbivore	69.6
Athlete	Throw_Ball	69.2
Male_Person	Government-Leader	69
Person	Sports	68.6
Male_Person	Election_Campaign_Debate	67.5
News_Studio	Studio_With_Anchorperson	67.4
Quadruped	Mammal	67.1
Person	Baseball	66.4
Athlete	Hockey	66
Quadruped	Sea_Mammal	65.9
Sports	Person_Drops_An_Object	65.9
Person	Indoor_Sports_Venue	65.7

Appendix C

The Top 50 Negative Correlations Selected in the IACC.1.A Data Set

Table C.1: The top 50 negative correlations selected in the IACC.1.A data set

Concept 1	Concept 2	ICF Value
Indoor	Outdoor	-0.08499
Daytime-Outdoor	Indoor	-0.05333
Indoor	Vegetation	-0.04883
Two-People	Single-Person	-0.04660
Male-Person	Female-Human-Face-Closeup	-0.03672
Trees	Indoor	-0.03531
Indoor	Building	-0.03302
Suburban	Indoor	-0.03239
Indoor	Plant	-0.03232
Road	Waterscape-Waterfront[1]	-0.03222
Indoor	Sky	-0.03138
Indoor	Road	-0.03035

Continued on next page. . .

[1] **The concepts before including this concept were selected using $Th1$.**

Concept 1	Concept 2	ICF Value
Crowd	Indoor	-0.02997
Crowd	Single-Person	-0.02932
Indoor-Sports-Venue	Outdoor	-0.02897
Indoor	Streets	-0.02889
Daytime-Outdoor	Face	-0.02857
News-Studio	Outdoor	-0.02809
Daytime-Outdoor	Nighttime	-0.02735
Celebrity-Entertainment	Outdoor	-0.0273
Computers	Face	-0.0264
Adult	Outdoor	-0.02627
Face	Outdoor	-0.02437
Adult	Road	-0.02378
Outdoor	Politicians	-0.02372
Dark-skinned-People	Outdoor	-0.02251
Politics	Singing	-0.02204
Streets	Waterscape-Waterfront	-0.02191
Chair	Single-Person	-0.02179
Face	Vegetation	-0.02172
Female-Person	Male-Person	-0.02165
Male-Person	Streets	-0.02156
Beach	Streets	-0.02133
Face	Overlaid-Text	-0.0213

Continued on next page. . .

Concept 1	Concept 2	ICF Value
Crowd	Two-People	-0.02128
Ground-Vehicles	Waterscape-Waterfront	-0.02099
Car	Waterscape-Waterfront	-0.02058
Beach	Road	-0.02057
Car	Indoor	-0.0205
Celebrity-Entertainment	Daytime-Outdoor	-0.02042
Meeting	Single-Person	-0.02036
Building	Landscape	-0.02021
Male-Person	Waterscape-Waterfront	-0.02007
Adult	Suburban	-0.01992
Indoor	Landscape	-0.01989
Face	Trees	-0.0198
Celebrity-Entertainment	Vegetation	-0.01972
Cityscape	Plant	-0.01958
Daytime-Outdoor	Politics	-0.01955
Daytime-Outdoor	News-Studio	-0.01953

Appendix D

The Top 50 Negative Correlations Selected in the IACC.1.B Data Set

Table D.1: The top 50 negative correlations selected in the IACC.1.B data set

Concept 1	Concept 2	ICF Value
Indoor	Outdoor	-0.10799
Daytime-Outdoor	Indoor	-0.06875
Single-Person	3-Or-More-People	-0.06315
Female-Person	Single-Person-Male	-0.05351
Two-People	Single-Person	-0.05225
Amateur-Video	Professional-Video	-0.04865
Male-Person	Single-Person-Female	-0.0472
Room	Outdoor	-0.04393
Indoor	Building	-0.04368
Sky	Indoor	-0.04239
Adult-Female-Human	Single-Person-Male	-0.04203
3-Or-More-People	Single-Person-Male	-0.03937

Continued on Next Page…

Concept 1	Concept 2	ICF Value
Female-Human-Face	Single-Person-Male	-0.03923
Adult-Male-Human	Single-Person-Female	-0.03876
Indoor	Trees	-0.03762
Indoor	Road	-0.03524
Indoor	Streets	-0.03484
Single-Person-Female	Single-Person-Male	-0.03462
Single-Person	Crowd	-0.03438
Outdoor	Indoor-Sports-Venue	-0.03433
Two-People	3-Or-More-People	-0.03351
Two-People	Single-Person-Male	-0.03166
Daytime-Outdoor	Room	-0.03144
Urban-Scenes	Indoor	-0.03109
Daytime-Outdoor	Nighttime	-0.03101
Indoor	Suburban	-0.02949
Indoor	Sunny	-0.02776
Single-Person-Female	3-Or-More-People	-0.0263
Female-Human-Face-Closeup	Male-Person	-0.02582
Single-Person-Female	Man-Wearing-A-Suit	-0.02559
Two-People	Single-Person-Female	-0.02543
Indoor	Cityscape	-0.02478
Indoor	City	-0.02424
Outdoor	News-Studio	-0.024

Continued on Next Page…

Concept 1	Concept 2	ICF Value
Single-Person-Female	Male-Human-Face-Closeup	-0.02367
Indoor	Vegetation	-0.0233
Daytime-Outdoor	Indoor-Sports-Venue	-0.0228
Adult-Male-Human	Female-Human-Face-Closeup	-0.02264
Single-Person-Male	Crowd	-0.02253
Indoor	Fields	-0.02213
Indoor	Waterscape-Waterfront	-0.02206
Single-Person-Female	Beards	-0.02185
Room	Building	-0.02131
Indoor	Clouds[1]	-0.021
Urban-Park	Indoor	-0.02039
Indoor	Car	-0.02005
Two-People	Crowd	-0.01998
Indoor	Residential-Buildings	-0.01966
Sky	Room	-0.01958
Male-Human-Face-Closeup	News-Studio	-0.01918

[1] **The concepts before including this concept were selected using** $Th1$

Bibliography

[1] J. M. Tien, “Big data: Unleashing information,” *Journal of Systems Science and Systems Engineering*, vol. 22, no. 2, pp. 127–151, June 2013.

[2] D. Laney, “3-d data management: Controlling data volume, velocity and variety,” *META Group Research Note, February*, vol. 6, pp. 1–4, February 2001.

[3] R. M. Ward, R. Schmieder, G. Highnam, and D. Mittelman, “Big data challenges and opportunities in high-throughput sequencing,” *Systems Biomedicine*, vol. 1, no. 1, pp. 29–34, March 2013.

[4] K. S. Vilas, “Big data mining,” *International Journal of Computer Science and Management Research*, vol. 1, no. 1, pp. 12–17, August 2012.

[5] W. Fan and A. Bifet, “Mining big data: current status, and forecast to the future,” *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1–5, December 2012.

[6] M. Hirzel, H. Andrade, B. Gedik, G. Jacques-Silva, R. Khandekar, V. Kumar, M. Mendell, H. Nasgaard, S. Schneider, R. Soule *et al.*, “Ibm streams processing language: Analyzing big data in motion,” *IBM Journal of Research and Development*, vol. 57, no. 3/4, pp. 7:1–7:11, May-July 2013.

[7] C.-H. Lee and T.-F. Chien, “Leveraging microblogging big data with a modified density-based clustering approach for event awareness and topic ranking,” *Journal of Information Science*, vol. 39, no. 4, pp. 523–543, July 2013.

[8] J. Lerman, “Big data and its exclusions,” *Stanford Law Review Online*, vol. 66, no. 1, pp. 55–56, September 2013.

[9] C. A. Bhatt and M. S. Kankanhalli, “Multimedia data mining: state of the art and challenges,” *Multimedia Tools Applications*, vol. 51, no. 1, pp. 35–76, January 2011.

[10] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li, “An analysis of facebook photo caching,” in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, November 2013, pp. 167–181.

[11] B. Henne, C. Szongott, and M. Smith, “Snapme if you can: privacy threats of other peoples’ geo-tagged media and what we can do about it,” in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, April 2013, pp. 95–106.

[12] P. Tomancak, B. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S. Celniker, and G. Rubin, “Global analysis of patterns of gene expression during drosophila embryogenesis,” *Genome Biology*, vol. 8, no. 7, p. R145, July 2007.

[13] O. V. Laere, S. Schockaert, and B. Dhoedt, “Finding locations of flickr resources using language models and similarity search,” in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, October 2011, pp. 48:1–48:8.

[14] E. Gabarron, L. Fernandez-Luque, M. Armayones, and A. Y. Lau, “Identifying measures used for assessing quality of youtube videos with patient health information: A review of current literature,” *Interactive Journal of Medical Research*, vol. 2, no. 1, pp. e6.1–e6.9, March 2013.

[15] J. Pokorny, “Nosql databases: a step to database scalability in web environment,” *International Journal of Web Information Systems*, vol. 9, no. 1, pp. 69–82, 2013.

[16] M. Bhandarkar, “Hadoop: a view from the trenches,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2013, pp. 1138–1138.

[17] J. R. Smith, “Riding the multimedia big data wave,” in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July-August 2013, pp. 1–2.

[18] C. Allan, J.-M. Burel, J. Moore, C. Blackburn, M. Linkert, S. Loynton, D. MacDonald, W. J. Moore, C. Neves, A. Patterson, M. Porter, A. Tarkowska, B. Loranger, J. Avondo, I. Lagerstedt, L. Lianas, S. Leo, K. Hands, R. T. Hay, A. Patwardhan, C. Best, G. J. Kleywegt, G. Zanetti, and J. R. Swedlow, “Omero: flexible, model-driven data management for experimental biology,” *Nature Methods*, vol. 9, no. 3, pp. 245–253, March 2012.

[19] J. Fan, H. Luo, and A. K. Elmagarmid, “Concept-oriented indexing of video databases: Toward semantic sensitive retrieval and browsing,” *IEEE Transaction On Image Processing*, vol. 13, no. 7, pp. 974–992, July 2004.

[20] D. Liu, M.-L. Shyu, and G. Zhao, “Spatial-temporal motion information integration for action detection and recognition in non-static background,” in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2013)*, August 2013, pp. 626–633.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2005, pp. 886–893.

[22] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, August 1999, pp. 1150–1157.

[23] J. R. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *Proceedings of the 2003 International Conference on Multimedia and Expo*, July 2003, pp. 445–448.

[24] R. Benmokhtar and B. Huet, "An ontology-based evidential framework for video indexing using high-level multimodal fusion," *Multimedia Tools and Applications*, vol. 55, no. 3, pp. 1–27, December 2011.

[25] Y.-G. Jiang, J. Wang, S.-F. Chang, and C.-W. Ngo, "Domain adaptive semantic diffusion for large scale context-based video annotation," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (ICCV2009)*, September-October 2009, pp. 1420–1427.

[26] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[27] X. Wu, X. Zhu, G. Wu, and W. Ding, "Data mining with big data," *IEEE Transaction on Knowledge and Data Engineering*, vol. 26, no. 26, pp. 97–107, January 2014.

[28] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, January 2008.

[29] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. R. Bradski, A. Y. Ng, and K. Olukotun, "Evaluating mapreduce for multi-core and multiprocessor systems," in *the Twentieth Annual Conference on Neural Information Processing Systems*, December 2006, pp. 281–288.

[30] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis, "Evaluating mapreduce for multi-core and multi-processor systems," in *Proceedings of the IEEE 13th International Symposium on High Performance Computer Architecture*, February 2007, pp. 13–24.

[31] S. Papadimitriou and J. Sun, "Disco: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining," in *Proceedings of the*

8th IEEE International Conference on Data Mining (ICDM 2008), December 2008, pp. 512–521.

[32] J. Lin and A. Kolcz, “Large-scale machine learning at twitter,” in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, May 2012, pp. 793–804.

[33] R. Sumbaly, J. Kreps, and S. Shah, “The big data ecosystem at linkedin,” in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, June 2013, pp. 1125–1134.

[34] S. Das, Y. Sismanis, K. S. Beyer, R. Gemulla, P. J. Haas, and J. McPherson, “Ricardo: integrating r and hadoop,” in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, June 2010, pp. 987–998.

[35] D. Wegener, M. Mock, D. Adranale, and S. Wrobel, “Toolkit-based high-performance data mining of large data on mapreduce clusters,” in *2009 IEEE International Conference on Data Mining Workshops*, December 2009, pp. 296–301.

[36] Z. Prekopcsák, G. Makrai, T. Henk, and C. Gáspár-Papanek, “Radoop: Analyzing big data with rapidminer and hadoop,” in *Proceedings of the 2nd RapidMiner Community Meeting and Conference (RCOMM 2011)*, June 2011, pp. 1–12.

[37] A. Ghoting and E. Pednault, “Hadoop-ml: An infrastructure for the rapid implementation of parallel reusable analytics,” in *NIPS 2009 workshop on Large-Scale Machine Learning: Parallelism and Massive Datasets*, December 2009.

[38] S.-F. Chang, “How far we’ve come: Impact of 20 years of multimedia information retrieval,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 9, no. 1, pp. 42:1–42:4, October 2013.

[39] T.-R. Hsiang, Y. Fu, C.-W. Chen, and S.-L. Chung, “A mapreduce-based indoor visual localization system using affine invariant features,” *Computers and Electrical Engineering*, vol. 39, no. 7, pp. 2369–2378, November 2013.

[40] T. Suzuki and T. Ikenaga, “Sift-based low complexity keypoint extraction and its real-time hardware implementation for full-hd video,” in *Proceedings of the 2012 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, December 2012, pp. 1–6.

[41] J. S. Hare, S. Samangooei, and P. H. Lewis, “Practical scalable image analysis and indexing using hadoop,” *Multimedia Tools and Applications*, vol. 61, no. 2, pp. 1–34, November 2012.

[42] F. Cai and H. Chen, “A mapreduce scheme for image feature extraction and its application to man-made object detection,” in *Proceedings of the Fifth International Conference on Digital Image Processing*, April 2013, pp. 88 782D–1:88 782D–7.

[43] A. Bergamo and L. Torresani, “Meta-class features for large-scale object categorization on a budget,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3085–3092.

[44] H. Wang, Y. Shen, L. Wang, K. Zhufeng, W. Wang, and C. Cheng, “Large-scale multimedia data mining using mapreduce framework,” in *Proceedings of the IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, November-December 2012, pp. 287–292.

[45] M. Yamamoto and K. Kaneko, “Parallel image database processing with mapreduce and performance evaluation in pseudo distributed mode,” *International Journal of Electronic Commerce*, vol. 3, no. 2, pp. 211–228, November 2012.

[46] N. K. Alham, M. Li, Y. Liu, M. Ponraj, and M. Qi, “A distributed svm ensemble for image classification and annotation,” in *Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, May 2012, pp. 1581–1584.

[47] B. Zhao, F. Li, and E. P. Xing, “Large-scale category structure aware image categorization,” in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS 2011)*, December 2011, pp. 1251–1259.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2009)*, June 2009, pp. 248–255.

[49] Y. Tsuji, H.-H. Huang, and K. Kawagoe, “Extending a distributed online machine learning framework for streaming video analysis,” in *Proceedings of the 2013 IIAI International Conference on Advanced Applied Informatics (IIAIAAI)*, August-September 2013, pp. 279–283.

[50] T. Skripcak and P. Tanuska, “Utilisation of on-line machine learning for scada system alarms forecasting,” in *Proceedings of the 2013 Science and Information Conference (SAI)*, October 2013, pp. 477–484.

[51] X.-J. Wang, L. Zhang, and C. Liu, “Duplicate discovery on 2 billion internet images,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2013, pp. 429–436.

[52] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3485–3492.

[53] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, November 2008.

[54] A. F. Smeaton, P. Over, and W. Kraaij, “Evaluation campaigns and TRECVid,” in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, October 2006, pp. 321–330.

[55] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, “Multiple feature hashing for real-time large scale near-duplicate video retrieval,” in *Proceedings of the 19th ACM International Conference on Multimedia*, November-December 2011, pp. 423–432.

[56] L. An, X. Chen, M. Kafai, S. Yang, and B. Bhanu, “Improving person re-identification by soft biometrics based reranking,” in *Procceedings of the 2013 ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, October 2013, pp. 1–6.

[57] T. Meng and M.-L. Shyu, “Model-driven collaboration and information integration for enhancing video semantic concept detection,” in *Proceedings of the 13th IEEE International Conference on Information Integration and Reuse (IRI2012)*, August 2012, pp. 144–151.

[58] D. Liu and M.-L. Shyu, “Semantic retrieval for videos in non-static background using motion saliency and global features,” in *Proceedings of the 2013 International Conference on Semantic Computing (ICSC)*, September 2013, pp. 294–301.

[59] Y. Yang, H.-Y. Ha, F. C. Fleites, and S.-C. Chen, “A multimedia semantic retrieval mobile system based on hidden coherent feature groups,” *IEEE Multimedia*, 2013, in press.

[60] L.-C. Chen, J.-W. Hsieh, D.-Y. Chen, and Y. Yan, “Vehicle make and model recognition using sparse representation and symmetrical surfs,” in *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems*, October 2013, pp. 1143–1148.

[61] Q. Zhu, L. Lin, M.-L. Shyu, and D. Liu, “Utilizing context information to enhance content-based image classification,” *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 2, no. 3, pp. 34–51, July-September 2011.

[62] H.-Y. Ha, Y. Yang, F. C. Fleites, and S.-C. Chen, “Correlation-based feature analysis and multi-modality fusion framework for multimedia semantic retrieval,” in *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo (ICME2013)*, July 2013, pp. 1–6.

[63] C. Chen, T. Meng, and L. Lin, “A web-based multimedia retrieval system with mca-based filtering and subspace-based learning algorithms,” *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 4, no. 2, pp. 13–45, April-June 2013.

[64] L. Lin, C. Chen, M.-L. Shyu, and S.-C. Chen, “Weighted subspace filtering and ranking algorithms for video concept retrieval,” *IEEE MultiMedia*, vol. 18, no. 3, pp. 32–43, March 2011.

[65] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, “Objects in context,” in *Proceedings of the 2007 IEEE International Conference on Computer Vision*, June 2007, pp. 1–8.

[66] C. Galleguillos, A. Rabinovich, and S. Belongie, “Object categorization using co-occurrence, location and appearance,” in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 144–151.

[67] M. Bar and S. Ullman, “Spatial context in recognition,” *Perception*, vol. 25, no. 3, pp. 324–352, 1996.

[68] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, “Multi-class segmentation with relative location prior,” *International Journal of Computer Vision*, vol. 80, no. 3, pp. 300–316, December 2008.

[69] A. B. Torralba, K. P. Murphy, and W. T. Freeman, “Contextual models for object detection using boosted random fields,” in *Proceedings of the 2004 Neural Information Processing Systems Workshop*, Vancouver, British Columbia, Canada, December 2004, pp. 1401–1408.

[70] A. Torralba, “Contextual priming for object detection,” *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, July 2003.

[71] G. Heitz and D. Koller, “Learning spatial context: Using stuff to find things,” in *Proceedings of the 10th European Conference on Computer Vision*, October 2008, pp. 30–43.

[72] W. Jiang, S.-F. Chang, and A. C. Loui, “Active context-based concept fusion with partial user labels,” in *Proceedings of the 2006 IEEE International Conference on Image Processing (ICIP 06)*, October 2006, pp. 2917–2920.

[73] G. Ciocca, C. Cusano, S. Santini, and R. Schettini, “Halfway through the semantic gap: Prosemantic features for image retrieval,” *Information Sciences*, vol. 181, no. 22, pp. 4943–4945, November 2011.

[74] M. R. Naphade, T. Kristjansson, B. Frey, and T. S. Huang, “Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems,” in *Proceedings of the 1998 IEEE International Conference on Image Processing*, October 1998, pp. 536–540.

[75] J. Tang, X.-S. Hua, M. Wang, Z. Gu, G.-J. Qi, and X. Wu, “Correlative linear neighborhood propagation for video annotation,” *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 409–416, April 2009.

[76] R. Yan, M. yu Chen, and A. Hauptmann, “Mining relationship between video concepts using probabilistic graphical models,” in *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo*, July 2006, pp. 301–304.

[77] W. Jiang, S.-F. Chang, and A. Loui, “Context-based concept fusion with boosted conditional random fields,” in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, vol. 1, April 2007, pp. 949–952.

[78] M. J. Choi, A. Torralba, and A. S. Willsky, “A tree-based context model for object recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 240 –252, Febrary 2012.

[79] Y.-G. Jiang, Q. Dai, J. Wang, C.-W. Ngo, X. Xue, and S.-F. Chang, “Fast semantic diffusion for large scale context-based image and video annotation,” *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 3080–3091, June 2012.

[80] T. Meng and M.-L. Shyu, “Leveraging concept association network for multimedia rare concept mining and retrieval,” in *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo*, July 2012, pp. 860–865.

[81] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, M. Wang, and H.-J. Zhang, “Correlative multilabel video annotation with temporal kernels,” *ACM Transaction on Multimedia Computing, Communications, and Applications*, vol. 5, no. 1, pp. 3:1–3:27, October 2008.

[82] A. F. Smeaton, C. Foley, D. Byrne, and G. J. Jones, “ibingo mobile collaborative search,” in *Proceedings of the 2008 international conference on Content-based image and video retrieval*, July 2008, pp. 547–548.

[83] T. Meng and M.-L. Shyu, “Concept concept association information integration and multi model collaboration for multimedia semantic concept detection,” *Information System Frontiers*, vol. 15, no. 2, pp. 1–13, April 2013.

[84] C. G. M. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring, “Adding semantics to detectors for video retrieval,” *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 975–986, August 2007.

[85] M. Naphade, J. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, “Large-scale concept ontology for multimedia,” *IEEE MultiMedia Magazine*, vol. 13, no. 3, pp. 86–91, July-Septermber 2006.

[86] Y. Wu, B. Tseng, and J. Smith, “Ontology-based multi-classification learning for video concept detection,” in *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo*, June 2004, pp. 1003–1006.

[87] Z.-J. Zha, T. Mei, Z. Wang, and X.-S. Hua, “Building a comprehensive ontology to refine video concept detection,” in *International workshop on multimedia information retrieval*, Augsburg, Germany, September 2007, pp. 227–236.

[88] X.-Y. Wei, C.-W. Ngo, and Y.-G. Jiang, “Selection of concept detectors for video search by ontology-enriched semantic spaces,” *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1085–1096, October 2008.

[89] L. Ballan, M. Bertinti, A. D. Bimbo, and G. Serra, “Video annotation and retrieval using ontologies an rule learning,” *IEEE Multimedia*, vol. 17, no. 4, pp. 80–88, October-December 2010.

[90] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, pp. 39–41, 1995.

[91] N. Elleuch, M. Zarka, A. B. Ammar, and A. M. Alimi, “A fuzzy ontology-based framework for reasoning in visual video content analysis and indexing,” in *Proceedings of the Eleventh International Workshop on Multimedia Data Mining*, August 2011, pp. 1–8.

[92] Y. Aytar, O. B. Orhan, and M. Shah, “Improving semantic concept detection and retrieval using contextual estimates,” in *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo*, July 2007, pp. 536–539.

[93] Y.-H. Yang, “Video search reranking via online ordinal reranking,” in *Proceedings of the 2008 IEEE International Conference on Multimedia and Expo*, June 2008, pp. 285–288.

[94] K.-H. Liu, M.-F. Weng, C.-Y. Tseng, Y.-Y. Chuang, and M.-S. Chen, “Association and temporal rule mining for post-filtering of semantic concept detection in video,” *IEEE Transactions on Multimedia*, vol. 10, no. 2, pp. 240–251, February 2008.

[95] M.-F. Weng and Y.-Y. Chuang, “Multi-cue fusion for semantic video indexing,” in *Proceedings of the 16th ACM International Conference on Multimedia*, October 2008, pp. 71–80.

[96] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 263–286, January 1995.

[97] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[98] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[99] T. Elomaa, “The biases of decision tree pruning strategies,” in *Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis*, August 1999, pp. 63–74.

[100] S. D. Bay, “Combining nearest neighbor classifiers through multiple feature subsets,” in *Proceedings of the 15th International Conference on Machine Learning*, July 1998, pp. 37–45.

[101] I. Rish, “An empirical study of the naive bayes classifier,” in *the 2001 IJCAI-01 workshop on Empirical Methods in AI*, August 2001, pp. 41–46.

[102] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” in *Data Mining and Knowledge Discovery*, vol. 2, no. 2, June 1998, pp. 121–167.

[103] C. Cortes and V. Vapnik, “Support-vector networks,” *Maching Learning*, vol. 20, no. 3, pp. 273–297, September 1995.

[104] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[105] E. Bredensteiner and K. Bennett, "Multicategory classification by support vector machines," in *Computational Optimization*, J.-S. Pang, Ed. Springer US, 1999, pp. 53–79.

[106] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, no. 1, pp. 265–292, March 2002.

[107] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 67–81, March 2004.

[108] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, no. 2, pp. 101–141, December 2004.

[109] J. C. Platt, N. Cristianini, and J. S. Taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K. R. Mueller, Eds., 2000, pp. 547–553.

[110] S. Escalera, O. Pujol, and P. Radeva, "Error-correcting ouput codes library," *Journal of Maching Learning Research*, vol. 11, pp. 661–664, March 2010.

[111] J.-B. Yang and I. Tsang, "Hierarchical maximum margin learning for multi-class classification," in *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, July 2011, pp. 753–760.

[112] H. Jabeen and A. Baig, "A framework for optimization of genetic programming evolved classifier expressions using particle swarm optimization," in *Hybrid Artificial Intelligence Systems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6076, pp. 56–63.

[113] N. Hamilton, R. Pantelic, K. Hanson, J. L. Fink, S. Karunaratne, and R. D. Teasdale, "Automated sub-cellular phenotype classification: an introduction and recent results," in *Proceedings of the 2006 workshop on Intelligent systems for bioinformatics*, December 2006, pp. 67–72.

[114] H. Peng, "Bioimage informatics: a new area of engineering biology," *Bioinformatics (Oxford, England)*, vol. 24, no. 17, pp. 1827–1836, September 2008.

[115] X. Long, W. Louis Cleveland, and Y. Lawrence Yao, "Multiclass detection of cells in multicontrast composite images," *Computers in Biology and Medicine*, vol. 40, no. 2, pp. 168–178, February 2010.

[116] M. H. Baea, R. Pana, T. Wua, and A. Badeab, “Automated segmentation of mouse brain images using extended mrf,” *Neuroimage*, vol. 46, no. 3, pp. 717–725, July 2009.

[117] H. T. Madhloom, S. A. Kareem, H. Ariffin, Z. A. A., A. H. O., and B. B. Zaidan, “An automated white blood cell nucleus localization and segmentation using image arithmetic and automatic threshold,” *Journal of Applied Sciences*, vol. 10, no. 11, pp. 959–966, June 2010.

[118] R. Minamikawa-Tachino, N. Kabuyama, T. Gotoh, S. Kagei, M. Naruse, Y. Kisu, T. Togashi, S. Sugano, H. Usami, and N. Nomura, “High-throughput classification of images of cell transfected with cdna clones,” *Molecular Biology and Genetics*, vol. 326, no. 10, pp. 993–1001, October-November 2003.

[119] L. Shamir, N. Orlov, D. M. Eckley, T. Macura, J. Johnston, and I. G. Goldberg, “Wndchm - an open source utility for biological image analysis,” *Source Code for Biology and Medicine*, vol. 3, no. 13, pp. 943–947, July 2008.

[120] M. R. Lamprecht, D. M. Sabatini, and A. E. Carpenter, “Cellprofiler: free, versatile software for automated biological image analysis,” *Biotechniques*, vol. 42, no. 1, pp. 71–75, January 2007.

[121] B. Misselwitz, G. Strittmatter, B. Periaswamy, M. C. Schlumberger, S. Rout, P. Horvath, K. Kozak, and W.-D. Hardt, “Enhanced cellclassifier: a multi-class classification tool for microscopy images,” *BMC Bioinformatics*, vol. 11, no. 30, pp. 1–13, January 2010.

[122] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, “Nih image to imagej: 25 years of image analysis,” *Nature Methods*, vol. 9, no. 7, pp. 671–675, June 2012.

[123] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Y. Tinevez, D. J. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, “Fiji: an open-source platform for biological-image analysis,” *Nature methods*, vol. 9, no. 7, pp. 676–682, June 2012.

[124] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, “Knime - the konstanz information miner: version 2.0 and beyond,” *SIGKDD Exploration Newsletter*, vol. 11, no. 1, pp. 26–31, November 2009.

[125] S. Kothari, J. Phan, A. Young, and M. Wang, “Histological image classification using biologically interpretable shape-based features,” *BMC Medical Imaging*, vol. 13, no. 9, pp. 1–17, March 2013.

[126] J. C. Caicedo, A. Cruz, and F. A. Gonzalez, “Histopathology image classification using bag of features and kernel functions,” in *Proceedings of the 12th Conference on Artificial Intelligence in Medicine: Artificial Intelligence in Medicine*, March 2009, pp. 126–135.

[127] S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, “Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features,” in *Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, May 2008, pp. 496–499.

[128] P. Wang, S.-M. Krishnan, C. Kugean, and M. P. Tjoa, “Classification of endoscopic images based on texture and neural network,” in *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, October 2001, pp. 3691–3695.

[129] Y. Zhang, S. Wang, and L. Wu, “A novel method for magnetic resonance brain image classification based on adaptive chaotic pso,” *Progress In Electromagnetics Research*, vol. 109, pp. 325–343, October 2010.

[130] K. Suzuki, “Machine learning in computer-aided diagnosis of the thorax and colon in ct: A survey,” *IEICE Transactions on Information and Systems*, vol. E96-D, no. 4, pp. 772–783, April 2013.

[131] E.-L. Chen, P.-C. Chung, C.-L. Chen, H.-M. Tsai, and C.-I. Chang, “An automatic diagnostic system for ct liver image classification,” *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 6, pp. 783–794, June 1998.

[132] A. Bin Tufail, A. Abidi, A. M. Siddiqui, and M. S. Younis, “Multiclass classification of initial stages of alzheimer’s disease using structural mri phase images,” in *Proceedings of the 2012 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, November 2012, pp. 317–321.

[133] K. Oppedal, K. Engan, D. Aarsland, M. Beyer, O. B. Tysnes, and T. Eftestol, “Using local binary pattern to classify dementia in mri,” in *Proceedings of the 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, May 2012, pp. 594–597.

[134] B. Krawczyk and G. Schaefer, “Effective multiple classifier systems for breast thermogram analysis,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*, November 2012, pp. 3345–3348.

[135] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[136] T. Ahonen, A. Hadid, and M. Pietikainen, “Face recognition with local binary patterns,” in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, vol. 3021, ch. 36, pp. 469–481.

[137] D. Markonis, A. García Seco de Herrera, I. Eggel, and H. Müller, “Multi-scale visual words for hierarchical medical image categorization,” in *Proceedings of SPIE Medical Imaging*, April 2012, pp. 83 190F–83 190F.

[138] Y. Xu, J. Liu, N. M. Tan, B. H. Lee, D. Wong, M. Baskaran, S. Perera, and T. Aung, “Anterior chamber angle classification using multiscale histograms of oriented gradients for glaucoma subtype identification,” in *Proceedings of the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, August-September 2012, pp. 3167–3170.

[139] L. Nanni and A. Lumini, “A reliable method for cell phenotype image classification,” *Artificial Intelligence in Medicine*, vol. 43, no. 2, pp. 87–97, June 2008.

[140] M. Soriano, L. Garcia, and C. Saloma, “Fluorescent image classification by major color histograms and a neural network,” *Optics Express*, vol. 8, no. 5, pp. 271–277, February 2001.

[141] M. Hafner, M. Liedlgruber, A. Uhl, A. Vecsei, and F. Wrba, “Color treatment in endoscopic image classification using multi-scale local color vector patterns,” *Medical Image Analysis*, vol. 16, no. 1, pp. 75–86, January 2012.

[142] N. J. Fesharaki and H. Pourghassem, “Medical x-ray images classification based on shape features and bayesian rule,” in *Proceedings of the 2012 Fourth International Conference on Computational Intelligence and Communication Networks (CICN)*, November 2012, pp. 369–373.

[143] S. M. Mohammadi, M. S. Helfroush, and K. Kazemi, “Novel shape-texture feature extraction for medical x-ray image classification,” *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 659–676, January 2012.

[144] U. Castellani, P. Mirtuono, V. Murino, M. Bellani, G. Rambaldelli, M. Tansella, and P. Brambilla, “A new shape diffusion descriptor for brain classification,” in *MICCAI (2)*, ser. Lecture Notes in Computer Science, G. Fichtinger, A. L. Martel, and T. M. Peters, Eds., vol. 6892. Springer, 2011, pp. 426–433.

[145] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua, “Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features,” *IEEE Transactions on Medical Imaging*, vol. 31, no. 2, pp. 474–486, February 2012.

[146] D. You, M. M. Rahman, S. Antani, D. Demner-Fushman, and G. R. Thoma, “Text- and content-based biomedical image modality classification,” in *Proceedings of SPIE Medical Imaging*, June 2013, pp. 86 740L1–86 740L8.

[147] K. S. Deepak, H. G. N. Rai, S. Syed, and P. R. Krishna, “Texture edge statistics for efficient retrieval of biomedical images,” in *Proceedings of the 5th ACM COMPUTE Conference: Intelligent and Scalable System Technologies*, January 2012, pp. 11:1–11:6.

[148] E. I. Zacharaki, S. Wang, S. Chawla, D. Soo Yoo, R. Wolf, E. R. Melhem, and C. Davatzikos, “Classification of brain tumor type and grade using mri texture and shape in a machine learning scheme,” *Magnetic Resonance in Medicine*, vol. 62, no. 6, pp. 1609–1618, December 2009.

[149] H. Qureshi, N. Rajpoot, T. W. Nattkemper, and V. Hans, “A robust adaptive wavelet-based method for classification of meningioma histology images,” in *Proceedings of the MICCAI 2009 Workshop on Optical Tissue Image Analysis in Microscopy, Histology, and Endoscopy*, September 2009.

[150] I. Buciu and A. Gacsadi, “Gabor wavelet based features for medical image analysis and classification,” in *Proceedings of the 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies, 2009 (ISABEL 2009)*, October 2009, pp. 1–4.

[151] S. Dua, U. R. Acharya, P. Chowriappa, and S. V. Sree, “Wavelet-based energy features for glaucomatous image classification,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 1, pp. 80–87, January 2012.

[152] S. Murala, R. Maheshwari, and R. Balasubramanian, “Directional binary wavelet patterns for biomedical image indexing and retrieval,” *Journal of Medical Systems*, vol. 36, no. 5, pp. 2865–2879, October 2012.

[153] O. Sertel, U. V. Catalyurek, H. Shimada, and M. N. Guican, “Computer-aided prognosis of neuroblastoma: Detection of mitosis and karyorrhexis cells in digitized histological images,” in *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009)*, September 2009, pp. 1433–1436.

[154] W. Wang, J. A. Ozolek, and G. K. Rohde, “Detection and classification of thyroid follicular lesions based on nuclear structure from histopathology images,” *Cytometry Part A*, vol. 77, no. 5, pp. 485–494, May 2010.

[155] K. Nguyen, A. Jain, and R. Allen, “Automated gland segmentation and classification for gleason grading of prostate tissue images,” in *Proceedings of the*

20th International Conference on Pattern Recognition (ICPR), August 2010, pp. 1497–1500.

[156] W. Ali, P. Piro, D. Giampaglia, T. Pourcher, and M. Barlaud, “Biological cells classification using bio-inspired descriptor in a boosting k-nn framework,” in *Proceedings of the 25th International Symposium on Computer-Based Medical Systems (CBMS)*, June 2012, pp. 1–6.

[157] L. Wei, Y. Yang, and R. M. Nishikawa, “Microcalcification classification assisted by content-based image retrieval for breast cancer diagnosis,” *Pattern Recognition*, vol. 42, no. 6, pp. 1126–1132, June 2009.

[158] S. H. Raza, R. M. Parry, R. A. Moffitt, A. N. Young, and M. D. Wang, “An analysis of scale and rotation invariance in the bag-of-features method for histopathological image classification,” in *Proceedings of the 14th international conference on Medical image computing and computer-assisted intervention - Volume Part III*, ser. MICCAI’11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 66–74.

[159] I. F. Amaral, F. Coelho, J. F. P. da Costa, and J. S. Cardoso, “Hierarchical medical image annotation using svm-based approaches,” in *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)*, November 2010, pp. 1–5.

[160] P. K. Naik, N. Nitin, A. Janmeja, S. Puri, K. Chawla, M. Bhasin, and K. Jain, “B-mipt: A case tool for biomedical image processing and their classification using nearest neighbor and genetic algorithm,” in *Proceedings of the Second International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, January 2011, pp. 107–112.

[161] A. Marcano-Cedeño, J. Quintanilla-Domínguez, and D. Andina, “Wbcd breast cancer database classification applying artificial metaplasticity neural network,” *Expert Systems with Applications: An International Journal*, vol. 38, no. 8, pp. 9573–9579, August 2011.

[162] J. S. Kippenhan, W. W. Barker, S. Pascal, J. Nagel, and R. Duara, “Evaluation of a neural-network classifier for pet scans of normal and alzheimer’s disease subject,” *Journal of Nuclear Medicine*, vol. 33, no. 8, pp. 1459–1467, August 1992.

[163] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, “Biomedical image classification with random subwindows and decision trees,” in *Proceedings of ICCV workshop on Computer Vision for Biomedical Image Applications (CVIBA 2005)*, October 2005, pp. 220–229.

[164] V. Van Ravesteijn, C. van Wijk, F. Vos, R. Truyen, J. Peters, J. Stoker, and L. Van Vliet, "Computer-aided detection of polyps in ct colonography using logistic regression," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 120–131, January 2010.

[165] H. Pourghassem and H. Ghassemian, "Content-based medical image classification using a new hierarchical merging scheme," *Computerized Medical Imaging and Graphics*, vol. 32, no. 8, pp. 651–661, December 2008.

[166] L. Kuncheva, J. Rodriguez, C. Plumpton, D. Linden, and S. Johnston, "Random subspace ensembles for fmri classification," *IEEE Transaction on Medical Imaging*, vol. 29, no. 2, pp. 531–542, February 2010.

[167] M. Liu, D. Zhang, and D. Shen, "Ensemble sparse classification of alzheimer's disease," *NeuroImage*, vol. 60, no. 2, pp. 1106–1116, April 2012.

[168] B. Ko, S. Kim, and J.-Y. Nam, "X-ray image classification using random forests with local wavelet-based cs-local binary patterns," *Journal of Digital Imaging*, vol. 24, no. 6, pp. 1141–1151, December 2011.

[169] S. Jai-Andaloussi, A. Elabdouli, A. Chaffai, N. Madrane, and A. Sekkaki, "Medical content based image retrieval by using the hadoop framework," in *Proceedings of the 20th International Conference on Telecommunications (ICT)*, May 2013, pp. 1–5.

[170] D. Markonis, R. Schaer, I. Eggel, H. Muller, and A. Depeursinge, "Using mapreduce for large-scale medical image analysis," in *Proceedings of the 2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology (HISB)*, September 2012.

[171] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 1994 International Conference on Very Large Data Bases*, Santiago de Chile, Chile, September 1994, pp. 487–499.

[172] C. Archambeau, M. Valle, A. Assenza, and M. Verleysen, "Assessment of probability density estimation method: Parzen window and finite gaussian mixtures," in *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems*, May 2006, pp. 499–503.

[173] Y.-G. Jiang, "Prediction scores on TRECVID 2010 data set," http://www.ee.columbia.edu/ln/dvmm/CU-VIREO374/, 2010, last accessed on September 8, 2011. [Online]. Available: http://www.ee.columbia.edu/ln/dvmm/CU-VIREO374/

[174] N. Inoue and K. Shinoda, “A fast and accurate video semantic-indexing system using fast map adaptation and gmm supervectors,” *IEEE Transactions on Multimedia*, vol. 14, no. 4-2, pp. 1196–1205, August 2012.

[175] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” *Communications of ACM*, vol. 35, no. 12, pp. 61–70, December 1992.

[176] N. Inoue, Y. Kamishima, T. Wada, K. Shinoda, and S. Sato, “Tokyotech+ canon at trecvid 2011,” in *NIST 2011 TRECVID Workshop*, December 2011.

[177] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, “Dynamic itemset counting and implication rules for market basket data,” in *Proceedings of the 1997 ACM SIGMOD international conference on management of data*, vol. 26, June 1997, pp. 255–264.

[178] K. Lo and R. Gottardo, “Flexible mixture modeling via the multivariate t distribution with the box-cox transformation: an alternative to the skew-t distribution,” *Statistics and Computing*, vol. 22, no. 1, pp. 33–52, January 2012.

[179] D. Liu and M.-L. Shyu, “Semantic motion concept retrieval in non-static background utilizing spatial-temporal visual information,” *International Journal of Semantic Computing*, vol. 7, no. 1, pp. 43–67, 2013.

[180] T. Meng and M.-L. Shyu, “Automatic annotation of drosophila developmental stages using association classification and information integration,” in *Proceedings of the 12th IEEE International Conference on Information Resue and Integration (IRI 2011)*, August 2011, pp. 142–147.

[181] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin, “Systematic determination of patterns of gene expression during drosophila embryogenesis,” *Genome Biology*, vol. 3, pp. 88.1–88.14, December 2002.

[182] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2005, pp. 886–893.

[183] T. Quirino, Z. Xie, M.-L. Shyu, S.-C. Chen, and L. Chang, “Collateral representative subspace projection modeling for supervised classification,” in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’06)*, November 2006, pp. 98–105.

[184] L. Shamir, N. Orlov, D. M. Eckley, T. J. Macura, and I. G. Goldberg, "Iicbu2008: a proposed benchmark suit for biological image analysis," *Medical and Biological Engineering and Computing*, vol. 46, no. 9, pp. 943–947, July 2008.

[185] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–151, January 2008.

[186] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, April 2011.

[187] H. Zhong, W.-B. Chen, and C. Zhang, "Classifying fruit fly early embryonic developmental stage based on embryo in situ hybridization images," in *Proceedings of the 2009 IEEE International Conference on Semantic Computing(ICSC2009)*, September 2009, pp. 145–152.

[188] J. Ye, J. Chen, R. Janardan, and S. Kumar, "Developmental stage annoation of drosophila gene expression pattern images via an entire solution path for lda," *ACM Transactions on Knowledge Discovery from DATA(TKDD)*, vol. 2, no. 1, pp. 4:1–4:21, August 2008.

[189] J. Yi, Y. Peng, and J. Xiao, "Exploiting semantic and visual context for effective video annotation," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1400–1414, October 2013.

[190] F. Huenupán, N. B. Yoma, C. Molina, and C. Garretón, "Confidence based multiple classifier fusion in speaker verification," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 957–966, May 2008.