HIGH-THROUGHPUT VISUAL KNOWLEDGE ANALYSIS AND RETRIEVAL IN BIG DATA ECOSYSTEMS

A Dissertation

presented to

the Faculty of the Graduate School

University of Missouri – Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

HONGFEI CAO

Dr. Chi-Ren Shyu, Dissertation Supervisor

JULY 2015

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

HIGH-THROUGHPUT VISUAL KNOWLEDGE ANALYSIS AND
RETRIEVAL IN BIG DATA ECOSYSTEMS

presented by Hongfei Cao,

a candidate for the degree of

Doctor of Philosophy

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Chi-Ren Shyu

Dr. Guilherme DeSouza

Dr. Prasad Calyam

Dr. Sean Goggins

ACKNOWLEDGEMENTS

First of all, I would like to offer my greatest gratitude and most sincere thanks to my advisor, Dr. Chi-Ren Shyu, who is always patient and provided professional direction and guidance to me. With consistently supports and suggestions from him, I learned how to become a good researcher and self-motive person. Under his guidance, I became as a good Teaching Assistant and Researcher in Computer Science department. Additionally, I would like to thank my committee members, Dr. Guilherme DeSouza, Dr. Prasad Calyam and Dr. Sean Goggins. They provided so many suggestions during my Ph.D. study.

I also would like to thank my colleagues in the Inter-discipline of Data Analysis and Search Center who are always supportive. I would specifically like to thank Dr. Nan (Allen) Zhao, D. Bin Pang, Dr. Jason Green, Jing Han, and Mike Phinney for their suggestions, assistance, and ideas throughout my Ph.D. study. A special thanks also goes out to Professor Carla Allen as our medical expert. Without her expertise and support, this work would not be possible. Also, I would like to thank my funding sources, including grants from National Science Foundation (IIS-0812515, DBI-1053204).

Finally, I would like to thank my mother, Mrs. Xiangning Liu, and my wife, Yanli Yang for great spiritual support. Without them, this work would also not be possible.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

HIGH-THROUGHPUT VISUAL KNOWLEDGE ANALYSIS AND RETRIEVAL IN BIG DATA ECOSYSTEMS

HONGFEI CAO

Dr. Chi-Ren Shyu, Dissertation Supervisor

ABSTRACT

Visual knowledge plays an important role in many highly skilled applications, such as medical diagnosis, geospatial image analysis and pathology diagnosis. Medical practitioners are able to interpret and reason about diagnostic images based on not only primitive-level image features such as color, texture, and spatial distribution but also their experience and tacit knowledge which are seldom articulated explicitly. This reasoning process is dynamic and closely related to real-time human cognition. Due to a lack of visual knowledge management and sharing tools, it is difficult to capture and transfer such tacit and hard-won expertise to novices. Moreover, many mission-critical applications require the ability to process such tacit visual knowledge in real time. Precisely how to index this visual knowledge computationally and systematically still poses a challenge to the computing community.

My dissertation research results in novel computational approaches for high-throughput visual knowledge analysis and retrieval from large-scale databases using latest technologies in big data ecosystems. To provide a better understanding of visual reasoning, human gaze patterns are qualitatively measured spatially and temporally to model observers' cognitive process. These gaze patterns are then indexed in a NoSQL distributed database as a visual knowledge repository, which is accessed using various unique retrieval methods developed through this dissertation work. To provide meaningful retrievals in real

time, deep-learning methods for automatic annotation of visual activities and streaming similarity comparisons are developed under a gaze-streaming framework using Apache Spark.

This research has several potential applications that offer a broader impact among the scientific community and in the practical world. First, the proposed framework can be adapted for different domains, such as fine arts, life sciences, etc. with minimal effort to capture human reasoning processes. Second, with its real-time visual knowledge search function, this framework can be used for training novices in the interpretation of domain images, by helping them learn experts' reasoning processes. Third, by helping researchers to understand human visual reasoning, it may shed light on human semantics modeling. Finally, integrating reasoning process with multimedia data, future retrieval of media could embed human perceptual reasoning for database search beyond traditional content-based media retrievals.

CHAPTER ONE

INTRODUCTION

1.1 Motivation

Domain experts observe visual phenomena and amass visual knowledge [1, 2] over time. That visual knowledge developed by experts allows them to understand the implications of domain images. It also can be used in image database design to help user retrieve results in a smarter and better way by visual query, visual clues [3, 4].

For example, in the radiology domain, a radiologic technologist is often given a task, such as “What is the projection of the femur in this X-ray image”. An expert will likely check the image label (if provided) or some particular regions of interest (ROI) to get the candidate answer and then confirm by matching obtained visual phenomena with the tacit knowledge in their memory. This systematic image ROI checking and confirmation lets experts make the diagnosis or answer the question efficiently and precisely. To be sure, there are some different strategies to help experts to answer the same question. Experts may even come up with different visual strategies or different results for the same image. Therefore, the visual tacit knowledge is hard to articulate and share by current computational approaches, let alone to search and exchange. To capture and evaluate this visual tacit knowledge, several methods are used. The straightforward method is to ask experts to type or speak out their thoughts during analysis by think-aloud protocol. Then the think-aloud audio is transcribed afterwards for detecting the strategies the expert was using. This protocol is limited in its ability to explain real-time cognitive processes as it is based on the expert’s awareness of thought processes and the expert’s verbalization of those processes. Because the experts may not be aware of some aspects of their processing and may not choose to share all of their thoughts, think-aloud protocol results are less than comprehensive. Another method of capturing and evaluating visual tacit knowledge is functional MRI, but the imaging environment is not conducive to domain expert's image interpretation. Therefore, the compromise method we used is gaze tracking [5-7], which captures the real-time human reaction and unexplained tacit knowledge through visual stimulus. Gaze tracking has been widely used to identify image salient regions [8-10], assist content based image retrieval [11], compare user behaviors by mining common gaze patterns [12-14], and assist computer-human interaction. By index of cognitive approach – average of changes in pupil dilation per second [15, 16] -- it is also used to measure users' cognitive workload. Moreover, the approach [17] identifies and compares subjects' cognitive strategies. In addition, it has been successfully used to reveal observers' cognitive process [18] through the sequence of visual attention driven by human cognition, despite several issues, such as 'peripheral encoding' [19, 20] and gaze fixation/saccade misdetection [21, 22]. Although it is easy to collect gaze tracking data by gaze tracker, the complexity of the dataset increases the difficulty of analysis, summarization and indexing for future use, search and learning.

In order to better search and exchange gaze tracking data, we index and store it using the case-based reasoning (CBR) system, a technique in artificial intelligence used in learning and problem-solving systems for knowledge acquisition and storage [23-27]. Case-based reasoning [28-30] was first introduced in AI by Roger Schank [31] in his work on dynamic memory and memory organization packets (MOPs) theory of learning and reminding based on prior experience. According to the cognitive science of these theories, people typically solve problems based on concrete past experiences rather than approaching each new problem from scratch. A CBR system allows people to learn vicariously through indexed cases and provides a basis for decision-making in situations where the observer has little prior experience. CBR has also been used to implement knowledge-based methods for automated problem solving [32, 33]. Michael Richter presented knowledge containers [34] which introduced two fundamental issues of CBR: how to represent knowledge in a systematic way and how to improve the system over time. There are many different applications of CBR system used in problem solving. An example of CBR system is the knowledge innovation for technology in education (KITE) [35] which seeks to assist teachers in the integration of technology into their teaching. Another problem solving CBR is Déjà vu, a Hierarchical CBR system aimed at automating plant-control software design [36]. It deals with complex plant-control design tasks by its hierarchical structure and multiple-case reuse method. An image segment system was implemented based on case-based reasoning [37]. By retrieving similar images from the case library and adapting its' segment parameters, new computed tomography (CT) images are segmented efficiently.

Due to the massive amount of streaming gaze tracking data, it is hard to provide real time visual knowledge analysis and reasoning in traditional platform. Therefore, a big data ecosystem (Hadoop & Spark) – including the Hadoop [38, 39]. Ecosystem which offers a substantial library of tools, such as Spark [40], Spark GraphX [41, 42], Spark Streaming [43], Spark MLlib [44], HBase [45-47], Hive [48], and MapReduce [49-51] is used to process high-throughput human gaze tracking data. Spark is an in-memory computing data-analytics framework that works on top of the Hadoop Distributed File System [52] and YARN [53], which provides ability to process streaming data. All the data processing tasks utilize the Spark in-memory distributed computing framework that allows for processing of large datasets in parallel.

1.2 Dissertation Organization

This dissertation is organized as follows. Chapter 2 introduces visual knowledge reasoning for image analysis in various applications. It also contains a pyramid visual knowledge structure designed for our experiment along with the visual knowledge case-based reasoning architecture. Chapter 3 discusses data management in big data ecosystems – Hadoop and Spark and a general case-based reasoning (CBR) system for indexing, reusing and retaining. Chapter 4 describes two different types of representation of gaze tracking, spatial model (subgraph matching) and temporal model (Markov Chain, Conditional Random Field, and Markov Decision Process model). Each individual case is made up of a gaze tracking pattern (a sequence of fixation) along with visual medium and domain knowledge. In Chapter 5, a novel visual computational searching method designed for gaze tracking retrieval based on Markov Chain (MC) and graph models is described. The use of an MC model for gaze tracking analysis has been addressed in previous studies for exploring visual scanning patterns [54, 55]. Chapter 6 discusses the real time gaze tracking streaming analysis in big data ecosystem – Spark. In addition, two experiment configurations' case studies and the statistical significance finding within and across knowledge groups are reported. A gaze tracking application –storytelling based gaze tracking authentication is described in Chapter 7. Finally, the dissertation then ends with conclusions and future directions in Chapter 8.

CHAPTER TWO

VISUAL KNOWLEDGE CHARACTERIZATION USING GAZE TRACKING

Visual knowledge [56] is reasoned by observers based on the observed visual phenomena and tacit knowledge. An image analyst, such as a radiologic technologist, makes decisions based on domain images (e.g. X-ray) through systematic checks of relevant regions and visual features in an image. The decisions include, but are not limited to the image quality, image orientation, image contrast, part position, central ray direction and shape distortion, etc. Thus, for a radiologic technologist, a set of relevant regions needs to be checked and verified before answering any question or making any decisions. For example, in Figure 1, in order to answer the question "How do you orient yourself to this image?" based on a femur X-ray image, an expert will likely start by checking the acetabulum -- the joint space between the femurs to determine Anterior and superior locations on the image. If the acetabulum cannot be seen clearly which means the image does not have enough penetration, the expert may go back to use different aspects image. However, it is often difficult for both novices and practitioners to describe all the important information in an image. Due to the highly complex visual knowledge, current approaches are not up to the task of capturing it.

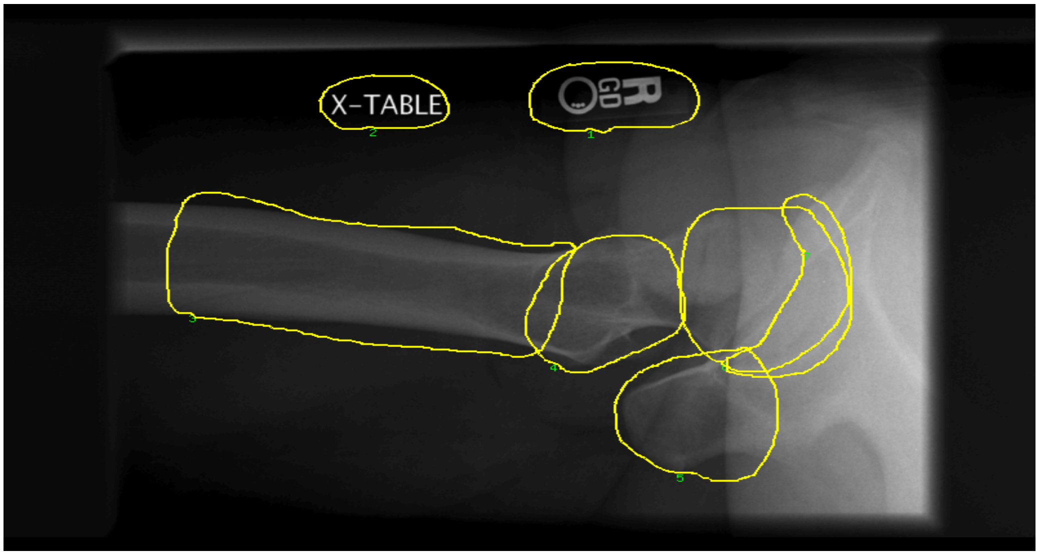


Figure 1: A sample X-ray image of femur with highlighted anatomic regions that are relevant to answer the question “How do you orient yourself to this image?”

2.1 An Overview of Image Reasoning

Image reasoning systems have been used to reason knowledge automatically through image content extraction and retrieval. Typical content-based image retrieval (CBIR) systems [57-59], following three steps below, can help us mine image content from domain images so the computers can understand high-level semantics from image. The first step is image feature extraction, which maps the image from image space to vector space by representing images as a set of image feature vectors. The second step is to generate high-level information by machine learning or data mining method from those features and get low-level knowledge models. At this point, domain knowledge is used to help create a high-level knowledge model. There is a significant gap between low-level and high-level knowledge models. In order to fill the gap, domain experts usually need to be involved and provide feedback or extra knowledge to assist image reasoning systems to create the model.

While the CBIR approach can achieve of high-level semantic understanding, it is still a long journey to index and mine the visual knowledge. However, many studies show that gaze tracking reasoning may be applied to this task. The gaze tracking analysis, in medical domain, has been used to differentiate patterns between children with autism and without [60, 61], identify relevant region in pathology images [62] and other medical images [63, 64]. Furthermore, visual reasoning is used in intelligent database retrieval, picture indexing and so on [65-67]. By reasoning on domain experts' gaze tracking data, we built a general-purpose model to capture this visual knowledge and make it searchable through a gaze-tracking CBR system. The system is designed to bridge the gap between experts and novices on how they look at domain images by the hierarchical CBR system and make gaze-tracking data searchable and adaptable for future using.

2.2 Levels of Abstraction for Visual Knowledge

We adapted a conceptual framework for visual knowledge indexing [68] which groups image features from low level (syntactic) to high level (semantic). This model has been tested and verified to be a robust instrument for indexing medical image [69]. In the framework, the knowledge is indexed by a 10-level pyramid as shown in Table 1. The first four levels refer to syntactic knowledge and the other six refer to semantic understanding. It shows that high levels (semantic) need more knowledge and information to index than low levels (syntactic). Syntactic levels (Level 1 to 4) are the most basic levels and focus on visual elements (e.g., dark spots, line, texture, and density) without considering the meaning of their arrangement. Thus, it does not involve much knowledge to describe these visual elements and no interpretation takes place at these four levels. On the other hand, semantic levels (Level 5 to 10) deal with the meaning of visual elements and require domain knowledge, experiences, and interpretation [69]. In order to elicit and capture the visual reasoning process from gaze tracking data, in the experiment, we designed 10 questions each refers to one knowledge level in 10-level knowledge pyramid. For example, a question like “What body parts does this image demonstrate?” was used to elicit visual reasoning processing when image analysts attempt to identify “Generic Objects” (Level 5). Table 1 shows the entire pyramid with 10 experimental visual knowledge levels and its’ examples. The questions designed by the 10-level knowledge structure help us define relevant regions in each domain image and specified different image features captured by observers. These 10 questions were asked before observers looked at image.

Table 1: 10-Level knowledge structure with correlated experiment questions (levels 1-4 are syntactic 5-10 are semantics).

Level	**Original Visual Knowledge Pyramid**	**Definitions and Examples**
1	Type Technique	Visual knowledge required to recognize the techniques and modalities used to produce the image (e.g., CT, X-ray, MRIs).
2	Global Distribution	Visual knowledge about the overall image characteristics such as “high or low contrast”.
3	Local Structure	Visual knowledge required to identify the basic visual elements (e.g., dot, line) and local details (e.g., shadow”).
4	Global Composition	Visual knowledge required to analyze the image as a whole but use the basic elements (e.g., symmetry, collimation”)
5	Generic Objects	Visual knowledge required to identify the most general level of object description with everyday knowledge (e.g., left leg, marker).
6	Generic Scene	Visual knowledge required to identify scenes at the most general level of description such as patient position or projection (e.g., AP position, oblique).
7	Specific Objects	Specific visual knowledge required to identify and name objects that are the primary subject of the overall image or local objects (e.g., lumbar spine, ET tube),
8	Specific Scene	Specific visual knowledge required to make judgment of the quality of the image (e.g., proper central ray position”).
9	Abstract Objects	Specific and interpretative visual knowledge required to summarize the information gleaned from the image (e.g., male patient, pediatric patient).
10	Abstract Scene	Specialized and interpretative knowledge required to carry out diagnostic or pathology judgments such as “fracture”, “bowel obstruction”.

2.3 Visual Knowledge Case-Based Reasoning

In our visual knowledge case-based reasoning system (VK-CBR), the overall architecture shown in Figure 2 includes four components: case raw data, case library, gaze distance measurements and a full CBR cycle [70]. During the data collection, subjects' gaze trackings were recorded by GazeTracker an open source infrared gaze tracker [71, 72]. A concurrent think-aloud protocol was used to solicit verbalized observations from the domain image and answers to the task questions. In VK-CBR system, each case consists of two parts: problem description and solution. The problem description includes gaze tracking data and the region of interest (ROI) in domain image. The case solution is the text description transcribed by observers' think-aloud audio. In the case library, system stores and indexes each gaze case and models them in two different representations – spatial and temporal (See also section 3). Then, system uses two different components to measure the distance between cases based on their spatial and temporal distance components. To measure the spatial distance by graph representation, we used a subgraph distance tool -- SAGA [73]. Three sub-components were designed to measure the gaze properties (gaze graph structure, total number of fixation and fixation duration) shown in Figure 2. For the temporal distance component, the system applies several Markov Chain comparison methods such as Markov forward algorithm to measure the temporal pattern. The overall distance combines both spatial and temporal distances as the final ranking results (See also section 4). Through a full CBR cycle with four processes, namely retrieve, reuse, revise and retain, after collecting and indexing each gaze case, similar cases will be retrieved and returned to the user. In the retrieval step, the CBR system allows user to retrieve similar gaze tracking cases with the domain knowledge to help them better understanding and interpreting domain images. Then, they can further adapt the retrieved results if necessary by temporal representation through the CBR reuse and adaptation process. Furthermore, a simulated ROI observation sequence is generated as the system recommends results. Last, by retaining, the system returns the new case with its solution to the user and stores them to the case library for future using.

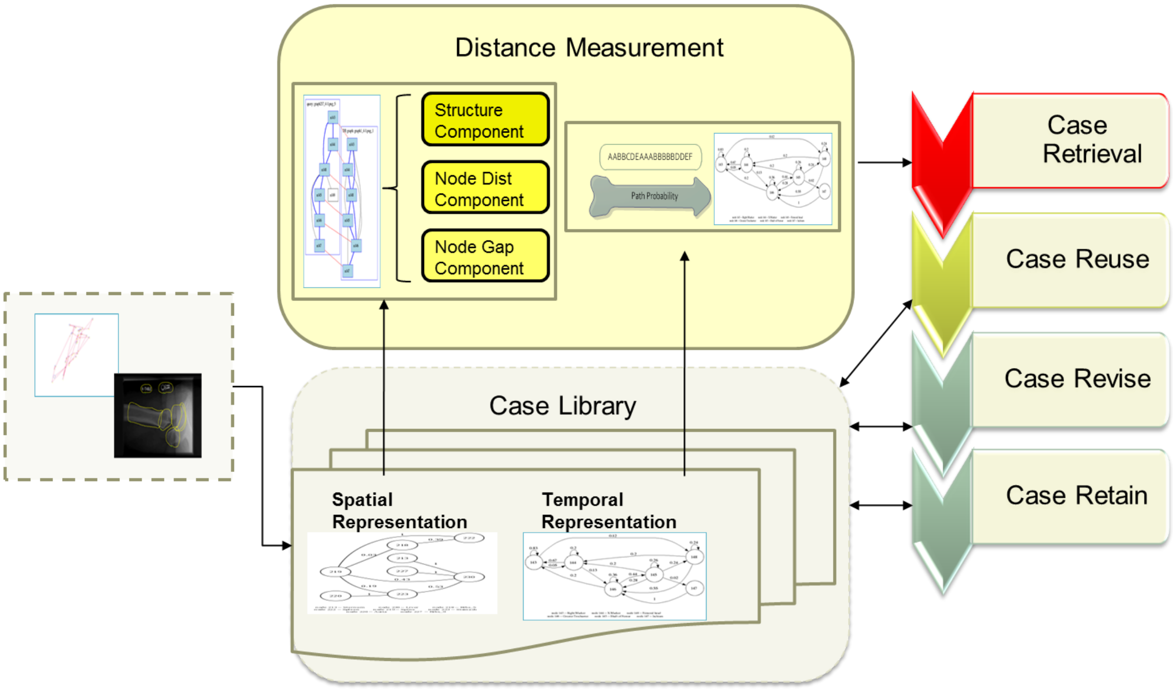


Figure 2: The overall Visual Knowledge CBR architecture and case base management.

CHAPTER THREE

CASE-BASED REASONING SYSTEM IN BIG DATA ECOSYSTEM

3.1 Overview

The visual knowledge developed by domain experts helps them to interpret or diagnose domain images. This tacit knowledge seldom articulates explicitly and is hard to capture and transfer in real time. In this dissertation, a distributed case-based reasoning (CBR) system was designed using a conditional random field (CRF) [74] to automatically label visual activities from gaze tracking streaming data, the reinforcement learning – Markov Decision Process (MDP) [75] and Spark graph model to capture human visual knowledge temporally and spatially on domain images (medical image, X-ray, Geospatial images, etc.). CRF is an undirected probabilistic graphic model for sequence data segmentation and labeling. Compared to a hidden Markov Model (HMM) [76, 77] or Markov Chain model (MC) [78], CRFs relax the independency assumption between hidden states (y_1, …, y_n). Moreover, generative models such as HMMs or MCs are unable to represent multiple interacting features or long-range dependencies of the observations (mixed-ordered sources). To solve the above issue, non-generative finite-state models such as Maximum entropy Markov models (MEMMs) have been designed [79]. MEMMs have another weakness - label bias problems. The transition weight in the model could have a bias calculation for some edges whose end state has fewer outgoing transitions. In contrast, conditional models - CRFs are able to model mixed-order sources and prevent modeling label bias issues. Researchers used a CRF model to capture human visual strategies in various applications. The CRF is used to model general reading strategies among readers

[80]. By training the CRF model using all users' gaze data, the accuracy of predicted reading strategies using CRF model can reach 94.62%. In addition, researchers use CRF to track user behavior in eye movement challenge problems [81]. The accuracy of classification is about 73%, which shows the difficulty of the task. Traditional prediction models were compared to CRF model as well, for example – Logistic model, Naïve Bayesian and SVM.

After generating gaze action from the CRF model, we used a Markov Decision Process model (MDP) to capture visual knowledge temporally. MDP models also known as reinforcement learning are designed for incremental learning processes. For gaze tracking analysis, it is able to capture visual knowledge's temporal characteristics. Through gaze actions and reward functions, MDP models can be trained to favor the desired visual reasoning pattern (domain experts).

As the size of the dataset grows, a traditional case based reasoning system fails to analyze gaze-tracking data due to the large volume. Moreover, the fast streaming speed of online dataset presents significant challenges to normal CBR systems, which analyze data at lower latencies. Through Apache Spark, a distributed computing framework, and MapReduce programming model, a distributed CBR system can capture and index the visual knowledge from humans in real time. The system allows us to efficiently partition data into different machines in a cluster and cache them in memory for fast processing. Furthermore, using Spark's GraphX component, the system can analyze gaze spatial patterns. In this dissertation, we proposed a probabilistic model (Markov Decision Process model) for real time gaze tracking and an online Expectation Maximization (EM) learning algorithm for training the parameters in the proposed model.

3.2 Discretized Streams

In order to stream gaze-tracking efficiently, analyze data in real time and run ad-hoc queries interactively, a distributed computing framework – Apache Spark - was used. One of the advantages of Spark is the built-in streaming component using Discretized Streams (D-streams) [43]. Through D-streams, Spark keeps the data in memory for operation and stores the streaming data reliably across the cluster on Hadoop Distributed File System (HDFS).

D-streams, such as Spark Streaming, [82] are distributed large-scale stream models designed for real-time data processing. Similar to MapReduce [83] in Hadoop, D-streams can read multiple resources from different nodes in a big data cluster. They can also read from various inputs/datasets, such as HDFS, web sockets (TCP/IP), plain text file, etc. Aside from the MapReduce computing model in Hadoop, users can also create general workflow on D-streams in Spark using operations such as *filter*, *flatMap*, *sample*, *union*, *intersection*, *reduceBy*, *aggregateBy*, *sortBy*, *groupBy*, and *join*. The results from the D-stream can be stored in multiple output resources and formats. For example, *saveAsTextFile* for plain text output, *saveAsSequenceFile* for sequence output and *saveAsObjectFile* for object output.

The streaming gaze tracking is first stored as D-streams and Spark performed a series of deterministic batch computations on D-streams within small time intervals. The input data can be stored on to HDFS periodically. Using Spark standard operations such as filter, reduceBy and groupBy, the received D-streams data are transformed into Resilient Distributed Datasets (RDDs) – a distributed storage that keeps the data in memory [84] across the cluster. The advantages of RDDs include in-memory computing, scalability and fault-tolerance. The data in RDD format can be distributed across cluster and caching in memory. It is extremely helpful in the analysis of data in real time since the gaze tracking data can be processed in memory on different nodes simultaneously. Meanwhile, RDDs allow Spark Streaming to reconstruct missing data when a node fails in cluster. Using D-streams in Spark Streaming, the system is able to achieve scaling, reliable and fault-tolerant in a commodity hardware cluster. In addition, Spark allows users to query streaming component interactively through defined D-Streams API. Thus, users can retrieve similar visual attention at any time of their image scan. It makes the Visual knowledge Case Based Reasoning interactively and response to users' searches in real time.

3.3 Raw Sequence Processing

Through a TCP/IP connection, gaze tracking raw sequence data is first sent to the Spark Streaming Component. Then, the data is stored on HDFS in different size of blocks. HDFS provides fault tolerant storage by default; three replications are created for each chunk of data. Next, using a sliding window approach, the system calculates fixation points and saccades from raw gaze stream in real time.

Figure 3 is the diagram for the processing input of the raw gaze tracking sequence. The input streaming gaze-tracking sequence is processed using a sliding window scan. In a time *t* window, we calculate potential fixation points using a velocity-based method [85]. The size of the window is defined in real time by streaming frequency and fixation duration. For example, the sample frequency of infrared gaze-tracker we used is gaze_frequency=34 fps. According to previous study, human fixation threshold is often defined as fixation_threshold=100-200ms. In this experiment, we choose 150ms. The sliding window size can be calculated as fixation_threshold * gaze_frequency = 0.15 (s)*34=5.1, which means we can set the window size to 6 to ensure each time only one fixation point is inside the sliding window.

After obtaining the fixation sequence in real time from the raw input, we can then map each fixation point into predefined region of interested (ROI). For now, we asked domain experts to manually label each ROI in an image. Other approaches include image processing methods such as image segmentation and labeling, which allow us to automatically obtain this ROI data. Since fixation-ROI mapping also needs to be done on the fly, it requires that each machine in the distributed cluster maintains its own copy of the ROI data (best in memory). One of issues of using this "broadcast" type of solution is the capacity of machine memory. If the ROI region data is too big, each machine/node in the cluster will quickly exhaust all of the memory resources to process the gaze-tracking data. In our case, the ROI data can be loaded into RAM and each time I can limit the number of images used. More details about mapping fixations to ROI regions will be discussed in Section 4.1.

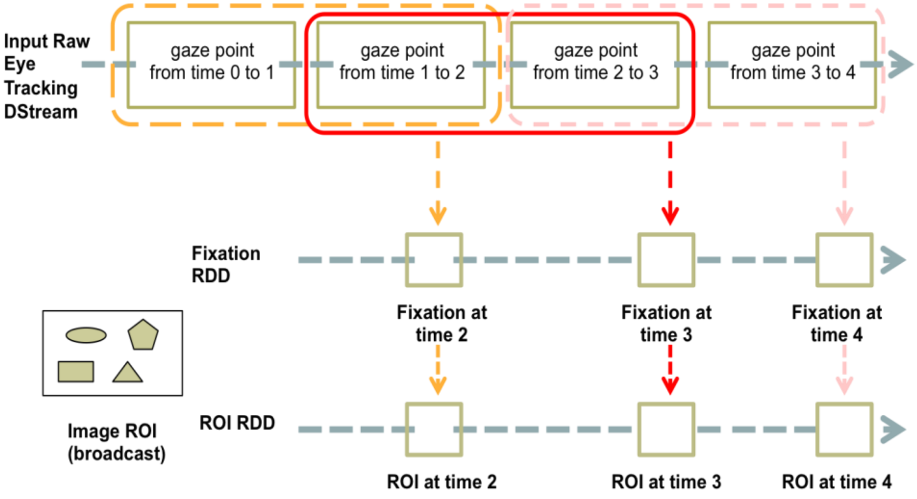

Input Raw Eye Tracking DStream
gaze point from time 0 to 1
gaze point from time 1 to 2
gaze point from time 2 to 3
gaze point from time 3 to 4
Fixation RDD
Fixation at time 2
Fixation at time 3
Fixation at time 4
Image ROI (broadcast)
ROI RDD
ROI at time 2
ROI at time 3
ROI at time 4

Figure 3: Preprocessing of gaze tracking data.

3.4 Gaze Feature Generation

The next step, after obtaining fixation and the corresponding ROI sequence, is to generate features for both gaze temporal and spatial representation. For a temporal model – linear chain Markov decision process, this includes a set of actions, a set of states with rewarding function and a sequence of observations. We used a Conditional Random Field (CRF) [86] model to automatically segment and annotate the gaze tracking activities. Three activities are defined from the streaming ROI sequence: read, scan and focus. In order to build this gaze tracking label CRF model, we first collect training data. Each visual action is manually labeled on collected gaze tracking data. Then, five fixation features are used for training CRF model: fixation duration, fixation dispersion, angle speed, pixel speed, and acceleration. Since training the CRF model requires iterative learning algorithms such as Viterbi, Belief Propagation, which is hard to learn in real time, it is better to train this CRF model offline. Equation 1 represents the CRF model in mathematic format, where Y is a set of actions as we defined above and observation X is the sequence of region of interested generated by the raw data processing.

$$P(Y|X) = (1/Z_\theta(x)) \times \exp(\theta \cdot \textstyle\sum_i f(X,Y,i)) \qquad (1)$$

The learning problem can be stated as: given observation X, the most likely sequence of Y is calculated by Equation 2. Since the CRF model has a convex cost function, the training solution is global optimal.

$$Y^* = \arg\max_Y P(Y \mid X) \qquad (2)$$

To evaluate the CRF predicted label results, the system simulates a gaze action label based on a simple threshold classification by Equation 3. Using the generated label, the system then evaluates the CRF prediction results. In Equation 3, dispersion is the maximum distance between current fixation and its neighbors in a range of θ.

$$A = \begin{cases} focus: dispersion < \theta_d \\ read: dispersion > \theta_d \land angle\_speed < \theta_a \\ scan: dispersion > \theta_d \land angle\_speed \ > \theta_a \end{cases} \tag{3}$$

For spatial representation G={V,E}, the gaze tracking graph is defined as vertex set V={ROI} and edge set E={e_{ij} | transition between node i and j}. The feature for spatial representation is generated from the sequence ROIs. Each ROI represents a vertex in the graph and the edge between node i and j is the transition between ROIs.

3.5 Learning from streaming

In order to retrieve similar gaze tracking cases in real time, the system must learn both temporal and spatial models on the fly. To accomplish this, we use an online learning approach to train the models. The first step is to split the entire observation time into fixed number of slots (f). Each time slot's length equals to $t_{slot} = t_{total}/f$. For example, if the total time is 15s and number of slots is 3, $t_{slot} = \frac{t_{total}}{f} = 5s$. Then, the online learning approach buffers each streaming data point into the memory until current time slot is completed. Thus, the system updates both temporal and spatial models incrementally. The Markov decision process (MDP) is represented in Equation 4, where the state S is a set of ROIs, and action A is the CRF predicted visual actions results. The transition matrix is calculated by Equation 5. Finally, the reward function R maps each state to an action. The MDP model is then trained to maximize the expected value $V^{\pi}(s) = E[R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \cdots |\pi]$ where π is the mapping function which maps state to action. Finally γ is the reward parameter.

$$\begin{aligned} &S : ROIs \\ &A : \{Read, Scan, Focus\} \\ &P_{sa} : TransitionMatrix \\ &R : S \times A \mapsto \mathbb{R} \end{aligned} \qquad (4)$$

$$P_{sa}(s') = \frac{\#\,times\ took\ \text{action a in state s and got to s'}}{\#\,times\ \text{took action a in state s}} \qquad (5)$$

For real time visual knowledge analysis, a new set of MDP parameters π is trained in Equation 6.

$$\begin{gathered} V^{*}(s) = maxV^{\pi}(s) \\ V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi}(s')V^{\pi}(s') \end{gathered} \qquad (6)$$

Similarly, the gaze spatial model is updated on each node automatically, since the whole data is processed on the Spark Cluster. The system requires each node to maintain a copy of its Spatial and Temporal Model locally, so syncing the model periodically between nodes is necessary. In the system, we setup the sync time as t_{slot}, which is same to the buffer length. The system tracks the node updates the temporal or spatial model and distributes the new model from that node to the rest of cluster.

The average number of fixation points within t_{slot} buffer is around 7-8. After obtaining the new fixation points, Equation 5 is used to update existing MDP model when the new knowledge (gaze points) is available. Thus, different user's temporal and spatial gaze tracking models can be updated at the same time across the cluster. Based on Equations 5 and 6, the new transition matrix is calculated. Similar to the temporal model, the graph model is updated by Equations 9-12. The system updates node, structure and gap distance based on new coming fixation points.

3.6 System architecture

Traditional MySQL or Oracle database driven CBR is not designed for large scale streaming computing. Thus, a distributed CBR for visual knowledge using HBase and HDFS is designed in this dissertation. HBase is a column oriented distributed database on top of HDFS which allows users to randomly access large amounts of data (read and write) in real time in a key-value format. The gaze tracking data is indexed and retrieved in real time by its temporal and spatial characteristics. The whole system architecture is shown in Figure 4. The streaming gaze tracking data is first sent to a Spark Streaming component for fixation segmentations. Then, the sequence of fixation is further annotated by a linear-chain Conditional Random Field to obtain visual activities (Focus, Read, and Scan) for each time stamp. The system then generates a Markov Decision Process model based on the annotated visual action and associated ROI sequence. The MDP model will be used for gaze temporal pattern retrieval in Distance Measurement component. All the generated MDP models and raw sequence gaze data are stored in HBase – a distributed database on the top of Hadoop Distributed File System. To achieve large scale re-ranking during the retrieval in real time, the system uses the Mean Shift clustering results for shuffling.

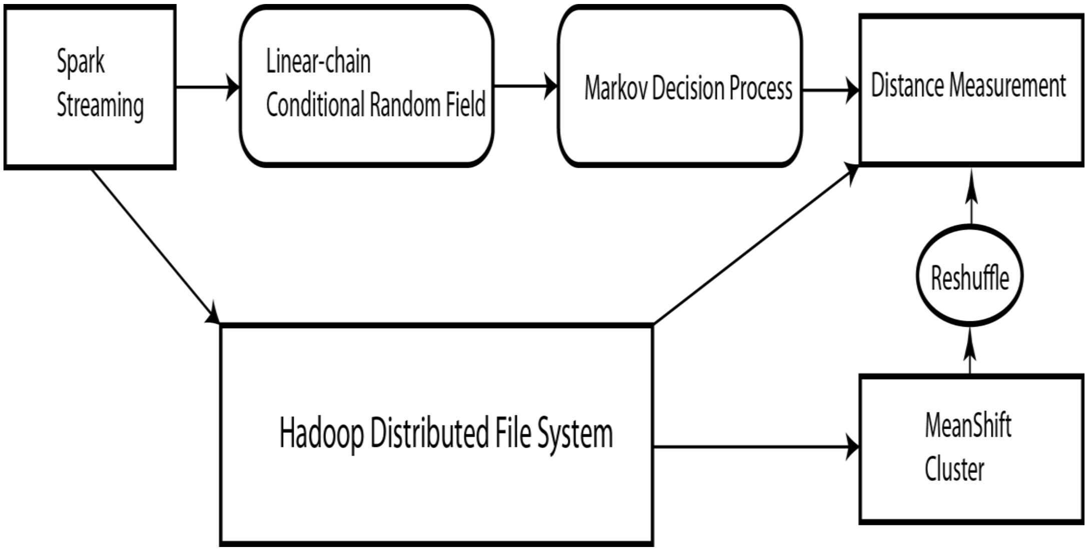


Figure 4: Distributed visual reasoning system architecture.

Similar to gaze-tracking raw sequence processing, the distributed CBR also automatically partitions data into different machines in a cluster. Each gaze-tracking case is represented in the form of a ROI sequence, temporal model, and spatial model. Gaze-tracking cases are stored in different machines across the cluster using HBase. Using Solr [87], an in-memory indexing open software from Apache, HBase indexes data in memory and eventually achieves real time query. Table 2 shows HBase table schema for gaze tracking raw sequence. Similar schema are used for table fixation-ROI as well.

Table 2: HBase Table Schema for Gaze Tracking Raw Sequence.

RowID	CF: coordinate	CF:fixation	CF:ROI
Case1	(X,Y)	Fixation1	ROI1
Case2	(X,Y)	Fixation2	ROI8

To avoid the issue of data skew, where a small number of nodes store the majority of data, we can presplit each table before the data comes in. This rowID presplit approach balances data loading and insertion across different machines in the cluster. Since each gaze tracking case has same duration ~ t_{total}, by presplitting the data in each table we can avoid an unbalanced data problem.

3.6.1 Distributed CBR Cycle

In a distributed CBR retrieval process, we proposed two models for gaze tracking: conditional random field (CRF) and Spark GraphX model. These two models represent the gaze tracking's temporal and spatial characteristics. The conditional random field builds a gaze temporal model from a raw ROI sequence. Since different users may have different ways to look at the same domain image, it is hard to model this mixture order of sources. Thus, CRF is a good temporal model for these types of the data due to the loose of independent assumption.

After obtaining similar gaze cases from the case base, the next step is adapting the retrieved result to a query gaze case. Because of the distributed framework, all results are split and sent to different nodes across the whole cluster. In order to auto-learn and re-rank retrieved results based on query case, the system needs to maintain a query on each node using a Spark broadcast function. Then, the CBR Revise step can generate a new CRF and spatial models using both query and retrieval cases.

In the CBR revise step, users first evaluate the adapted result, then they can change the transition weights between gaze fixations. The system regenerates the CRF and Spatial models using new weights provided by the users. Finally, after users have accepted the adapted solution, the distributed CBR system stores the results with the query case together in HBase for future use.

3.7 Evaluation

To evaluate the whole real time visual analysis Spark system, a combination of online think-aloud and offline user rating protocol were designed for this research. In the think-aloud section, the system asks users to speak out their reasoning process while they look at the domain images. This reasoning process will were recorded and translated into text data for later evaluation. In order to evaluate each gaze tracking result queried from visual knowledge database, the system uses Jaccard similarity to measure think-aloud data between the query and retrieved results. In addition, an offline user-rating web page is designed to record user satisfaction with retrieved results. In the rating web page, each user is asked to give a rating (from 1 to 5) for each retrieved result. This rating will be used to update CBR adaptation process later see in Section 5.3.

More specifically, for temporal model retrieval results evaluation, the system uses Longest Common Substring distance [88, 89]. The similarity is calculated between the query and retrieval results’ fixation sequences. The LCS distance is used to validate the temporal MDP retrieval results.

CHAPTER FOUR

GAZE TRACKING CASE REPRESENTATION

After collecting gaze tracking data and region of interest in domain images, the system is ready to represent the case. The first process in this phase of the system is case generation. Given the gaze tracking raw data, a set of attributes of gaze tracking such as gaze coordinates, durations, saccade lengths and angles are calculated automatically. Based on different fixation definitions, a fixation point is defined as the centroid (x, y) of a set of gaze points within a time period such as 100ms or 150ms. Time periods for fixation vary throughout the literature with most fixation lengths defined between 50 and 200ms [90]. In our experiment, we set 100ms as minimum dwell time to calculate fixation points. Figure 5 shows the gaze tracking on a domain image where the gaze fixation points are marked in solid green dots and the saccades (jump between two fixation points) are marked with a red straight line. Another attribute for gaze tracking is duration (dwell time) which measures how long a fixation point dwells on the image region. Figure 6 shows a sample of Lumbar Spine X-ray image with the raw sequence gaze data on the left and corresponding fixation data on the right. In Figure 5, note that solid dot increases in size with increased duration of the fixation. Next, each gaze fixation is mapped to ROIs based on their spatial relationship. Both spatial and temporal representations were then generated for each gaze case in the VK-CBR system.

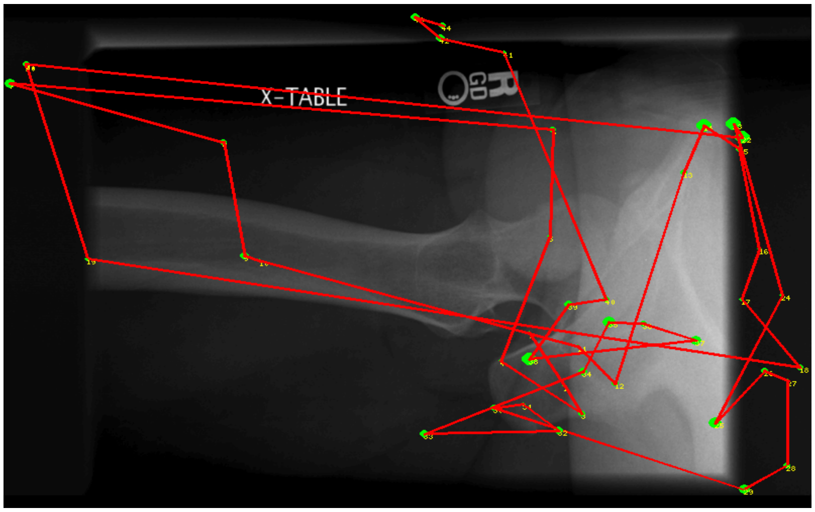


Figure 5: A sample of Eye-tracking fixation sequence on the X-ray femur image.

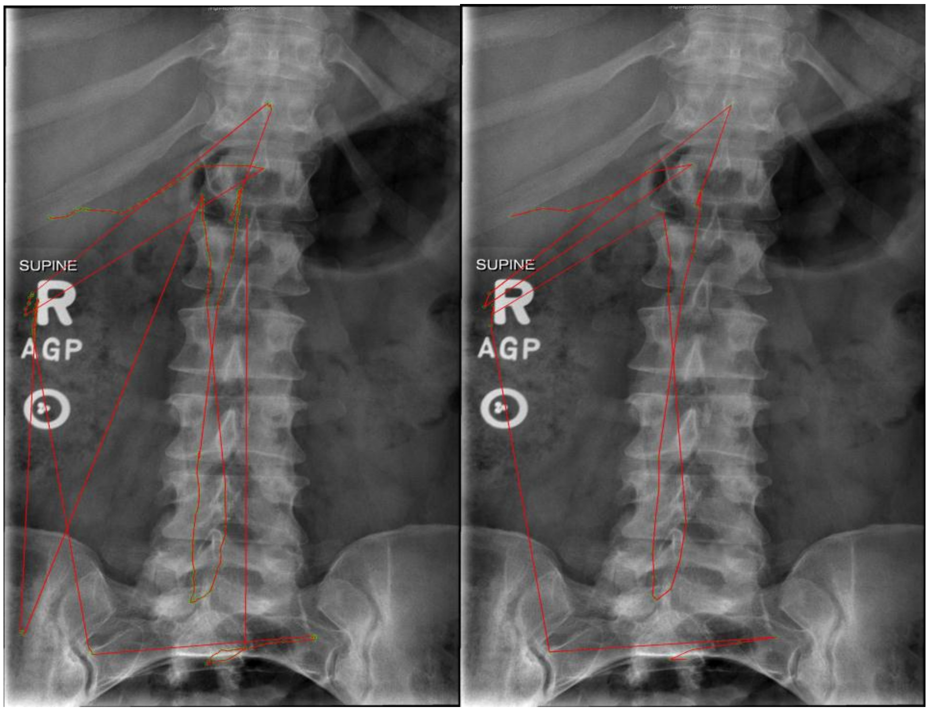


Figure 6: A sample of Lumbar Spine X-ray raw gaze tracking (left) and fixation sequence (right) results.

4.1 Mapping fixation to Region of Interest (ROI)

Based on ROIs defined by domain experts, the VK-CBR system first maps fixation points to the corresponding ROIs. Figure 7 shows a set of highlighted ROIs marked as polygons in an X-ray image. Each fixation point can fall into zero (undefined region), one or multiple regions of interest. Based on the following three rules, the system assigns each fixation point to a unique ROI when given the fixation data. (1) If the fixation point falls into an unmarked region (non-ROI), the system will assign this gaze point to the closest ROI. The distance is defined as the ROI exhibiting minimum distance between the fixation and the nearest region boundary. (2) When the fixation point is within one ROI, the system assigns the fixation to this region. (3) When the fixation falls into an overlapping region of multiple ROIs, the system assigns the fixation to the ROIs with a minimal area, which is likely to be relevant for the reasoning. Table 3 shows an example of fixation-ROI mapping result. Each fixation point was mapped to one unique corresponding ROI. Based on above rules, the system transfers the gaze tracking data from a fixation sequence to an ROI sequence. Next, the CBR system represents the gaze temporal characteristics using a Markov Chain representation, as well as the graph representation was generated to capture spatial feature.

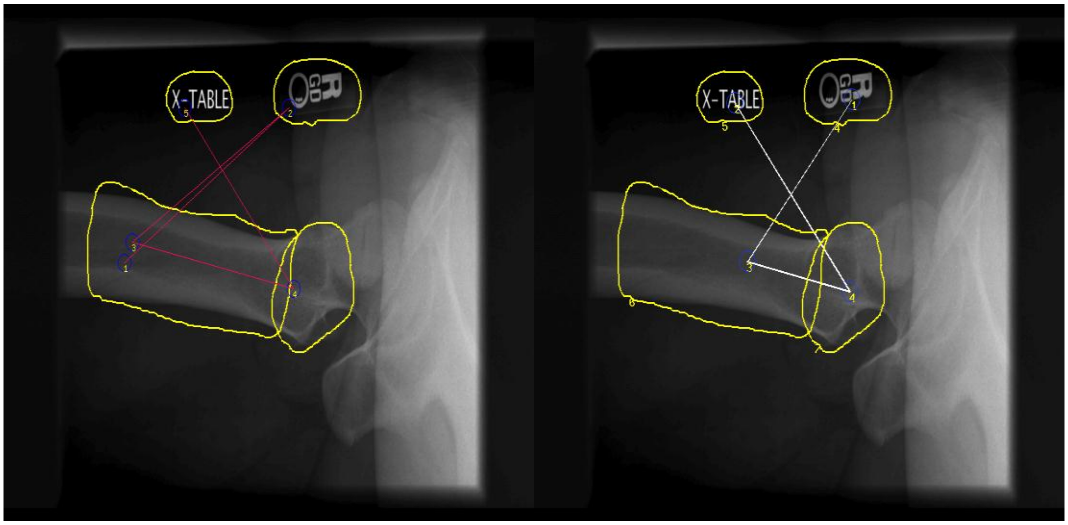


Figure 7: A sample X-ray image with gaze tracking and correlated graph mapping result.

Table 3: Sample of Eye-tracking Gaze Point with Corresponding ROI.

Fixation	X	Y	Duration	ROI	ROI Name
1	726	217	0.166	R1	X MARKER
2	972	96	0.226	R2	FEMORAL HEAD
3	1041	57	0.156	R3	SHAFT OF FEMUR
…	…	…	…	…	…

4.2 The Temporal Representation

4.2.1 Markov Chain model

After obtaining the corresponding ROI sequence for gaze tracking, VK-CBR system generates two different representations for each gaze tracking case – temporal and spatial. In this section, we discuss how to generate temporal representations in CBR system. The description of a temporal behavior is an *n*th-order Markov Chain (MC) model with m finite states (here *n* is up to 5 and *m* is the number of ROIs in one image). The MC model is defined by its initial vector π and the transition matrix contains the transition probability between each pair of states. The initial vector π defines the probability of any ROI occurring first when the user looks at the image. The transition matrix defines the probability between two states (ROIs) that user will look at them consecutively. Based on this MC model, the system captures temporal patterns for gaze activities. Further, the MC model is used to generate an adapted solution for the user by combining both query and retrieved cases during CBR adaptation process (See more detail in Section 4.3).

In order to generate the *n*th-order MC model, we assume that gaze tracking has the Markov property that the future observation only depends on past n fixation points. Specifically, for a first-order MC model, it means that the future observation is independent of past fixation points when given a current observation. In our cases, the *n*th-order MC model assumption means current visited ROI only depends on past n regions of interest (ROI) checked by user. Let ROI$_i$ be the state X_i, at time t, the *n*th-order Markov property is described as Equation 7 and first order Markov Property shown in Equation 8:

$$P(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_1 = x_1) \quad (7)$$
$$= P(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_{t-n-1} = x_{t-n-1})\ for\ t > n + 1$$

$$P(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_1 = x_1) \quad (8)$$
$$= P(X_{t+1} = x_{t+1} | X_t = x_t)$$

We demonstrate how to generate a first-order MC model for a gaze tracking case. Then, higher order MC models can be obtained in a similar way. The first step for generating an MC model is to count the number of transition between each pair of adjacent fixation points with corresponding regions of interest. Next, the system calculates the Markov Chain transition matrix *A* by iteratively calculating the weighted probability for each pair of states by Equation 9. The weight w is calculated by Equation 10, where $DT(ROI_j)$ is defined as total duration w the j-th ROI. Table 4 shows a sample of the transition matrix for the first-order Markov Chain model with transition count and probability listed in each cell.

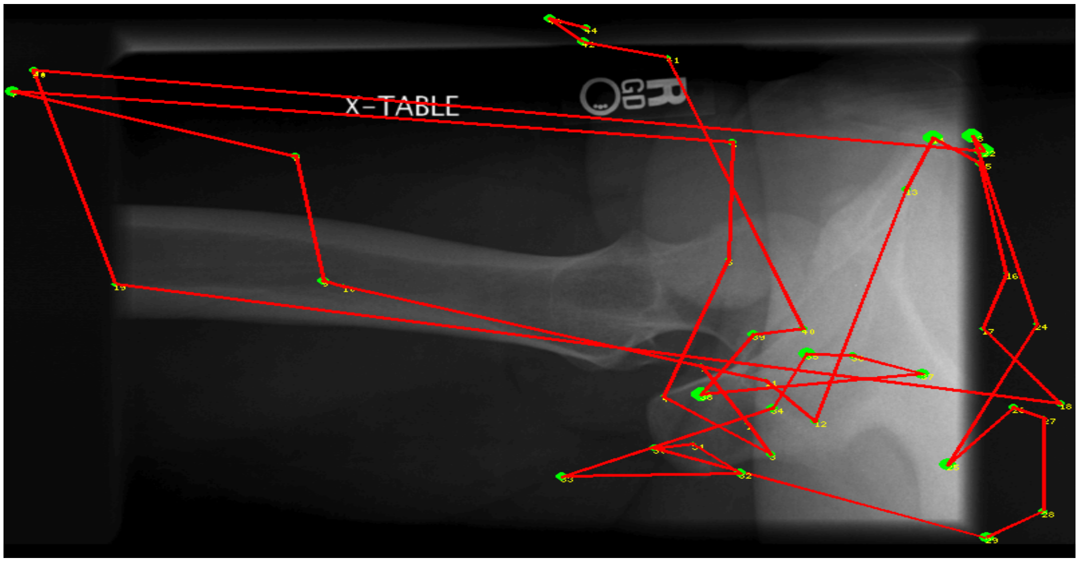
$$P(x_i|x_j) = w_j \times \frac{Count\ of\ transition\ from\ x_j\ to\ x_i}{All\ transition\ start\ from\ x_j} = \frac{w_j \times \mathrm{c}_{x_j x_i}}{\sum_{x_i} \mathrm{c}_{x_j x_i}} \tag{9}$$

$$w_j = \frac{DT(ROI_j)}{\sum_{\mathrm{j}} DT(ROI_j)} \tag{10}$$

Table 4: Sample transition count/probability result for first-order Markov Chain Model.

	Right	X	Shaft of	Great	Ischiu	Femoral	Acetabulu
Right Marker	4/0.25	4/0.25	1/0.06	4/0.25	0/0.00	1/0.06	2/0.13
X Marker	2/0.40	1/0.2	2/0.40	0/0.00	0/0.00	0/0.00	0/0.00
Shaft of	4/0.23	0/0.00	8/0.47	4/0.23	1/0.06	0/0.00	0/0.00
Greater	1/0.07	0/0.00	2/0.15	4/0.30	1/0.07	5/0.38	0/0.00
Ischium	0/0.00	0/0.00	1/0.10	1/0.10	5/0.50	3/0.30	0/0.00
Femoral Head	3/0.15	0/0.00	3/0.16	1/0.05	1/0.05	8/0.42	3/0.16
Acetabulum	1/0.16	0/0.00	1/0.17	0/0.00	2/0.33	1/0.17	1/0.17

Meanwhile, the system determines the initial vector π for MC model by the gaze corresponding ROI sequence. For example, in Table 4, region *R1* is the first ROI that the observer checked. Then, the initial vector π was assigned to (0,0,…,1,0,..0) where '1' corresponds to the first visited region *R1*. After obtaining the transition matrix *A* and initial vector π, the first order MC model is represented in the VK-CBR system. Figure 8 shows a sample of gaze tracking case with corresponding first-order MC model. Each node is an ROI with label name listed at bottom. The directed edge between node *R2* (Femoral Head) and node *R3* (Pelvis) with 0.33 weights indicates that there is a 33% chance that an observer will look at the region Femoral Head followed by Pelvis. The self-loop on node *R3* suggests that observers checked region *R3* continually within two gaze points. The VK-CBR system generates this temporal MC model for gaze tracking cases from each individual observer. Similarly, the system also constructs higher-order Markov Chain models using Equations 9 and 10 where $c_{x_j x_i}$ is the transition count between past *n* consecutive ROIs (x_j) and current ROI (x_i).



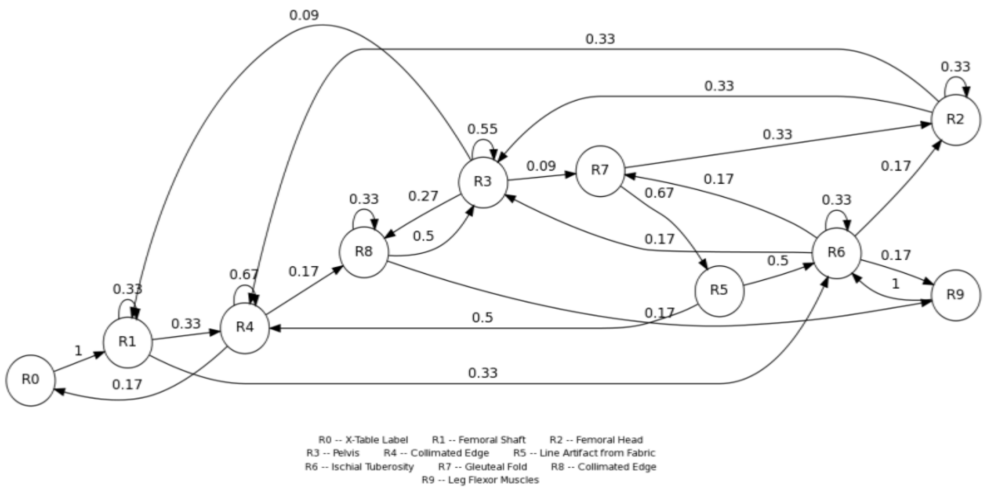


Figure 8: An example of gaze tracking case (Case662) with corresponding customized first-order MC Model.

4.2.2 Conditional Random Field

Besides a Markov Model temporal representation, the gaze tracking data is also presented as linear-chain Conditional Random Field (CRF). Through the CRF model, the gaze tracking data is auto segmented and mapped to visual action {X->Y}. Figure 9 shows an example of CRF model, where X is streaming gaze tracking fixation feature and Y is visual action, which includes {Read, Scan, Focus}. The fixation features include fixation duration, dispersion, angle speed, pixel speed, and acceleration. Duration of a fixation point is the dwell time [91] for measuring how long a user focus on the image. Fixation dispersion is the maximum distance between current fixation and its neighbors in a range of θ. It measures the spatial distribution between fixations. The angle speed is the angle changes per second which measures the fixation spatial relationship as well. The pixel speed and acceleration measure the user attention shifts frequency.

One of advantages of CRF model is it works well when the true data distribution's order is higher than the model (e.g. Hidden Markov Model). For normal visual task, it is usually true that users' current visual attention is depended on multiple previous fixations based on their short term memory. Thus, CRF model for visual action segmentation and prediction is preferred in this project.

A fully-labeled data set is required to train a CRF model. In our project, the data was manually labeled by expert for three visual actions {Focus, Read, Scan}. After obtaining the training data, a CRF model was learned based on Equation 2 using Stochastic Gradient Descend (SGD) with L1/L2 regularization method by Wapiti - a discriminative sequence labeling toolkit with linear-chain CRF model [92]. With L1 regularization, the training process can perform feature selection to reduce the model size but not as stable as L2. In this project, we chose L1 regularization since it allows us to handle large scale CRF model with hundreds of label output Y and billions of observation X. As a result of Wapiti, it suggests that SGD is recommended for most large scale application due to its quick converge speed even though it does not guarantee to find the optimal one.

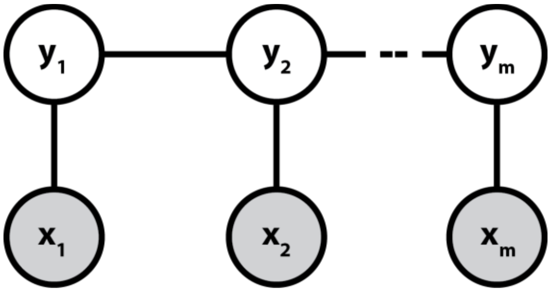


Figure 9: An example of linear-chain CRF model for visual action segmentation.

4.2.2.1 CRF Training

In order to train a CRF for visual activities segmentation, a fully-annotated training dataset is required. The input raw gaze tracking sequence is further segmented and annotated with defined uniform time period in the range of 450-600ms for three different visual actions {read, scan, focus}. This range can be customized later for different users and visual actions. Therefore, we can define the sliding window length as visual action duration (period time) divided by fixation sample rate (=34), as Equation 11 shows:

$$sw = \frac{Unified\ Visual\ Action\ Duration}{sample\ rate} \tag{11}$$

In our case, the sliding window length is in the range of (13-17) for a raw gaze tracking sequence, which was used in training set generation. After collecting the raw gaze tracking data, the first step is to generate a sequence of fixation by clustering temporal and spatial closed raw data points. Then, we generated five different fixation features, fixation duration, dispersion, anglespeed, speed and acceleration. The fixation duration is defined as dwell time for the current fixation obtained by gaze tracker, in our case is 150ms. Equations 12-15 show the formulae to calculate dispersion, anglespeed, speed and acceleration for fixation *j* based on previous adjacent fixation *i*. In Equation 12, dispersion of fixation *j* is the maximum distance between *j* to its neighbor fixation in a range with *j* as centroid and window size θ, where in our case, θ is equal to 5.

$$dispersion_j = \max_i \left(\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \right),$$

$$where\ i\ is\ j's\ neighbour\ in\ the\ range\ \theta \tag{12}$$

$$anglespeed_j = \frac{\cos^{-1}\frac{|x_i - x_j|}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}}{|t_i - t_j|} \tag{13}$$

$$speed_j = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{|t_i - t_j|} \tag{14}$$

$$acceleration_j = \frac{speed_j - speed_i}{|t_i - t_j|} \tag{15}$$

Thus, the training dataset is obtained by calculating the fixation features for each timestamp in one gaze tracking case. Table 5 shows an example of training data for CRF model with input fixation features and output label. Two patterns were defined for training after generating fixation features as well – unigram and bigram. Since the CRF model is trained offline for unigram pattern, it is reasonable to consider current observation fixation with both its previous and future fixation in each timestamp, within the range of sliding window length (34-68). The current fixation is in the center of the window. For bigram patterns, the data feature is generated combining the adjacent two fixation points' features (total 10). Through defined unigram and bigram patterns, the CRF model is trained using Stochastic Gradient Descent (SGD) with L1 regularization. Equation 16 shows the formula for SGD with L1 regularization, where $J(\theta)$ is the cost function shown as Equation 1.

$$\theta_j = \theta_j - \alpha \frac{d}{d\theta_j}\left(J(\theta) + \lambda \sum_{j=1}^{n} \theta_j\right) \tag{16}$$

One of the advantages of using SGD is its quick converge to find a local optimal result. L1 regularization can avoid overfitting (high variance) for model training and at the same time, generate sparse result for built feature selection. Thus, it can significantly reduce the complexity of the CRF model. Section 6.2.2 shows the training model and annotation results for CRF model.

Table 5: An example of CRF training data file with duration, dispersion, anglespeed, speed, and acceleration as features and Focus, Read, and Scan as labels output.

ID	Duration (ms)	Dispersion (pixel)	AngleSpeed $(degree/s)$	Speed $(pixel/s)$	Acceleration $(pixel/s^2)$	Label
1	183	100	10	25.5	0.3	Focus
2	210	140	20	30.1	4.1	Focus
3	200	200	24	22	1.1	Read
4	130	350	50	43	3.2	Scan
5	170	220	7	14	2.4	Read

4.2.3 Markov Decision Process

In order to capture the visual strategy and knowledge in real time, we proposed to use a Markov Decision Process (MDP) model [93] for gaze tracking. The MDP model is designed to simulate human visual reasoning in a computational manner. In our case, with predicted visual actions by CRF model, the actions are visual actions (focus, read, and scan). The learner/user then defines a policy for mapping from MDP state to the visual actions for highest reward. The MDP model contains a state set S – anatomy regions in the domain images; an action set A – visual actions (Read, Scan and Focus); a probability transition matrix P – defines the transition probability given state s_i and action a to state s_j; and a reward function R maps state s and action a to a real number reward. In our case, for each single state (an anatomy region in the image), it can take any of three actions but with different rewards.

One assumption of MDP is that current action only depends on current state. In our case, the states are the anatomical regions in the image, such as Sacrum or Lumbar Body, and visual actions are Focus, Read, and Scan. To learn and train this model, we first calculate the state transition matrix for the MDP states based on maximum likelihood estimates Equation 17 and initialize the reward function to obtain the policy based on Equation 18. If there is no transition between state s_i and s_j taken action a; we estimate the transition probability $P_a(s_i s_j) = 1/|S|$ for uniform distribution over all states. Different orders of MDP are also generated and compared. In our study, we tested on order 1, 2 and 3, due to different visual action time ranges. This MDP order can be customized later along with the time range for each visual action. We generated three different types of MDP models based on three different dataset. The first one is a single gaze tracking case, which each one is a single subject's gaze tracking on a single domain image. The second type is for an aggregated dataset from a single subject's gaze tracking on all eight domains images (total 3 visual knowledge levels + one overall question). The last type of MDP model is learned by the all subjects in the same knowledge group (junior and senior).

$$P_a(s_i s_j) = \frac{\#times\ tranit\ from\ state\ s_i\ to\ s_j\ took\ action\ a}{\#times\ in\ state\ s_i\ took\ action\ a} \tag{17}$$

$$R_{sa} = \frac{\#times\ took\ action\ a\ in\ state\ s}{\#times\ actions\ in\ state\ s} \tag{18}$$

After generating Markov Decision Process models for three different types of datasets (single, subject level, and group level), we calculate lifetime reward by value iteration algorithm as Equation 19. In Equation 19, R(s) is the immediate reward which gives the starting value for each state. The R(s) can be defined by domain experts and customized for different knowledge levels. In our case, there is no negative reward for the state / anatomy. The positive reward will be assigned to those salient regions which help answer the given question. The parameter γ is the discounted factor, which decreases the reward as the time increases. Thus, the model prefers to accrue positive rewards as soon as possible. In our study, we set γ to 0.9. The lifetime reward is the sum of all received rewards. Each state first initialize to zero. Then, algorithm repeat updates value for each state based on Bellman equations. The algorithm stops when all states have stable values, in other word, when the algorithm can obtain the best possible expected sum of discounted rewards (optimal value) V^* for each state.

$$\begin{aligned}
&\textit{Initialize each state}\, V(s) := 0 \\
&\text{Re } \textit{peat until convergence do}\, \{ \\
&\textit{For each state}, V(s) := R(s) + \max\nolimits_{a \in A} \gamma \sum\nolimits_{s'} P_{sa}(s')V(s') \\
&\}
\end{aligned} \tag{19}$$

Next, based on Equation 20, a visual reasoning process is simulated by a greedy policy iteration algorithm. Our goal is to maximize the expected value of the total reward: $V^{\pi}(s) = E[R(S_0) + \gamma R(S_1) + \gamma^2 R(S_2) + \cdots |\pi]$ by choosing a visual action in each timestamp. The algorithm keeps finding the current best visual action *a*, which gains maximum reward, and then update the state value V until the whole state's values are stable.

```
Re peat until convergence do {
    $V := V^{\pi}$
    For each state s, $\pi(s) := \arg\max_{a \in A} \sum_{s'} P_{sa}(s')V(s')$
}
```

(20)

4.3 The Spatial Representation

4.3.1 Graph Representation in Single Machine

In addition to capturing gaze temporal pattern, a graph model was created in the VK-CBR system for gaze cases. Thus, CBR can retrieve similar gaze tracking cases from the case base through a sub-graph matching function. This spatial representation for gaze case is an undirected graph with finite vertices and weighted edges. In this graph, each vertex represents a region of interest in the image and the edges between vertices indicate a direct transition from one vertex to another. The weight on each edge measures the linkage between two vertices. Higher weights indicate that the user more likely check the two ROIs together.

The first step in generation of the spatial representation is grouping the fixation points corresponding to the same ROI to a single set. Then, using the corresponding ROI sequence shown in Table 4, the system adds edges in the graph if there is a direct transition between two related fixation points. Finally, the system trims off the self-loop and double edge in the graph to obtain an undirected simple graph. The weight on each edge was obtained, and equals the total dwell time of ROI, *DT(Vi)*, weighted with count of transition, *T(Vi, Vj)*, between those two vertices. Figure 10 shows a sample of our spatial representation. Similar to the temporal representations described previously, which characterize temporal patterns, the graph model represents spatial patterns for gaze tracking. In this example, the edge (*R8,* *R9*) with weight 0.17 shows that there is a 17% chance that a user will look at region *R8* and region *R9* consecutively.

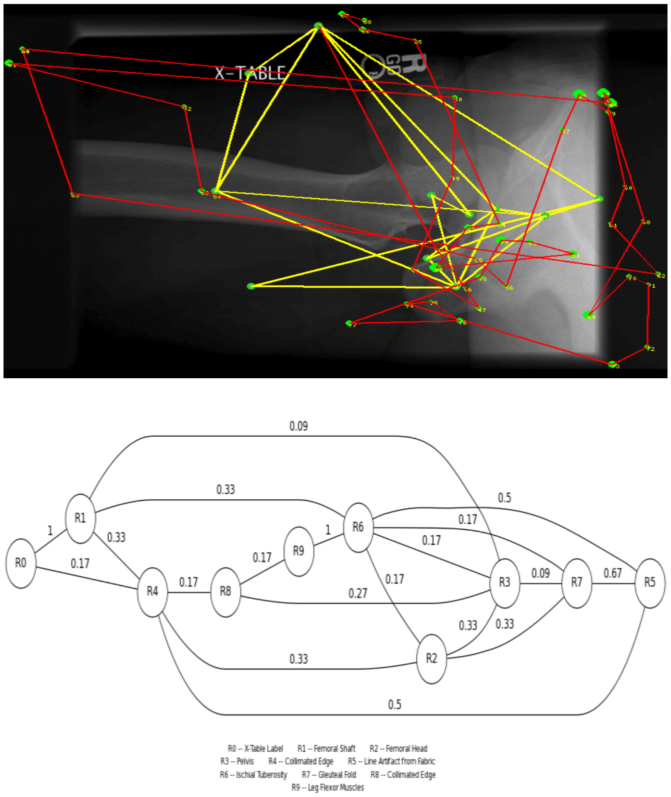


Figure 10: An example of eye-tracking spatial representation with ROI label.

4.3.2 Graph Representation in Spark – GraphX

With proposed distributed solution using Spark, each gaze tracking case is also represented as graph in Spark using GraphX [42] component. As Figure 10 shows, each graph {V,E} has node set V – region of interests from predefined domain image and edge set E – probability measurement between two nodes transition. In Spark GraphX, each graph is further split across different node in the cluster. The vertices are partitioned by vertex ID. Edges are partitioned as well, using predefined edge partition functions.

Figure 11 shows an example of graph representation distribute across on a cluster. In this way, each gaze tracking case can be represented in a distributed graph in GraphX. Moreover, the user can define partition function to horizontally partition the edge sets. Following the data locality rule – minimize data transferring across different machine - each machine in the cluster is tried to utilized as many as local edge set as possible. Finally, a routing table is designed to enable cluster quickly located partitioned edge and vertex subsets.

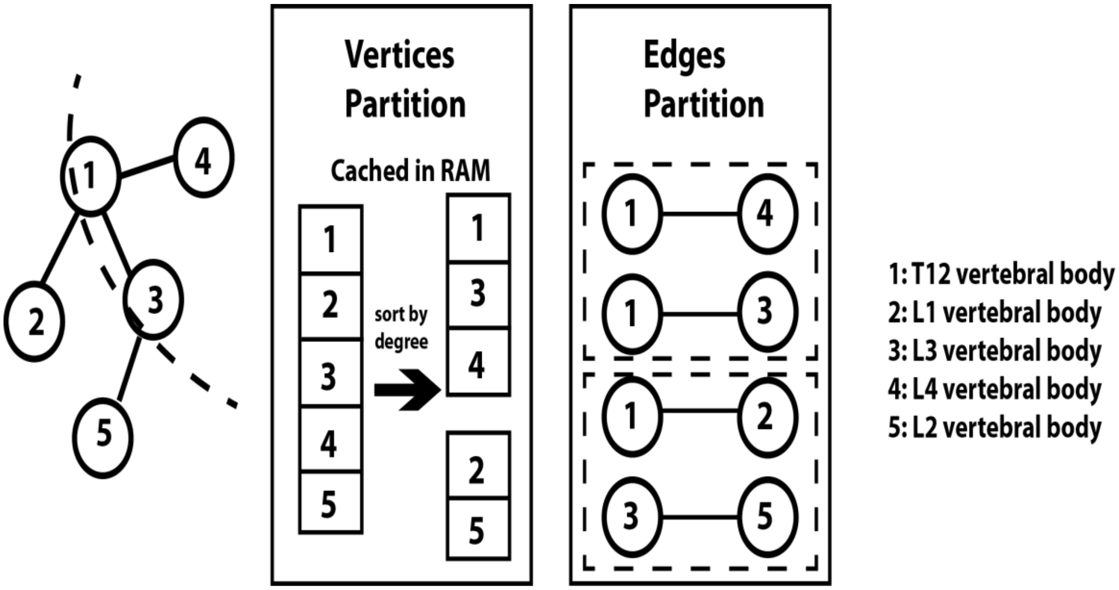


Figure 11: An example of graph partition in GrpahX adapted from [42].

CHAPTER FIVE

SIMILARITY FUNCTION BASED ON SPATIAL AND TEMPROAL COMPONENTS

In order to measure the distance between the query case and cases in the library, we utilize the following gaze tracking distance.

$$distance(Q,R) = w_g \times SGD_\lambda(Q,R) + w_t \times TD_\lambda(Q,R) \quad (21)$$

, where w_g and w_t are the weights of sub-graph distance (SGD) and temporal distance (TD), respectively. Observers can adjust the weights of the distance function.

5.1 Spatial Component

Spatial structure analysis is particularly useful for retrieving visual activities based on the task type of interest. It is known that visual behavior is dependent upon the task undertaken [94-97]. In medical imaging, understanding the orientation of the body part is dependent upon recognizing the relationship between two or more imaged structures. For example, when determining the alignment of the hips with the image receptor on an image of the abdomen or pelvis, the radiologic technologist will compare the symmetry of the ilia or wings of the pelvis. If the ilia are symmetrical, they know that the hips were equidistant from the tabletop and the patient was not rotated to one side or the other [98]. If rotation is detected, it can be confirmed by looking at the relationship of the spinous process to the associated lumbar vertebral body or the relationship of the sacrum and coccyx to the symphysis pubis. Because of this need to compare the relationship of two or more structures, we can expect visual behavior that transitions between those structures. It is also likely that matched transitions between two structures represent the same type of visual task. In the example above, the task was to determine rotation of the body part; however, if the task was to determine the phase of the blood cycle captured by the image, the fixations would be more likely focused on comparing the brightness of various vascular structures – renal arteries and veins, hepatic arteries and veins, splenic arteries and veins, etc. A search for images by matching spatial transitions enables the user to retrieve images that are not only structurally similar, but were also analyzed by previous users on a similar task and therefore can provide the searcher with greater information relevant to the task at hand.

After generating a graph model for query, CBR searches for matches among the whole case library using SAGA -- a sub-graph similarity method originally designed for biological pathways [73]. The spatial component allows user to efficiently search for sub-graph matches through the entire gaze tracking case library. In addition to matching the sub-graph structure, it also measures the node distances for similar nodes and absent nodes. The sub-graph distance component is defined in Equation 22.

$$SGD_{\lambda}(G_1, G_2) = w_e \times d_{struct} + w_n \times d_{node} + w_g \times d_{node\_gaps} \tag{22}$$

, where

$$d_{struct} = |E_1 - E_2| / |E_1| \tag{23}$$

$$d_{node} = \sum_{u \in \hat{V}_1} \left(w_f \times d_{att}(u, \lambda u) + w_d \times d_{dur}(u, \lambda u) \right) \tag{24}$$

$$d_{node\_gaps} = \sum_{u \epsilon V_1 - \hat{V}_1} gap_{G1}(u) \, / |V_1| \tag{25}$$

The sub-graph distance Equation 22 has three components, structure distance (d_{struct}), node distance (d_{node}), and node gaps (d_{node_gaps}). The d_{struct} component measures the structure difference between graph representations of the query and a case from the library. In Equation 23-25, $\hat{V}_1$ is a set of matched vertices/nodes between graph G_1 and G_2. λ is a mapping function, which maps similar nodes (ROIs) from different gaze tracking cases. The d_{node} is the distance between two matching nodes calculated based on their gaze fixations' attention, duration and underlining semantic relationship [73] or other controlled vocabularies for non-radiology applications. And the d_{node_gaps} is the penalty for the absent nodes. In Equation 22, w_s, w_n and w_g are the weights corresponding to d_{struct}, d_{node} and d_{node_gaps} components, respectively. By adjusting the above weights, this gaze distance function gives observers the freedom to choose emphasis between three spatial components for gaze spatial distance. The following examples demonstrate the reasoning of the various distances for each component.

The d_{struct} component measures the structural differences in the two cases' graphs. It is defined in Eq. 23 which equals to the number of unmatched edges from query graph to the matched graph. Figure 12 shows an example of a matched result in which the query case is on the left and the result case is on the right. The highlighted bold black edges are matched edges. Thus, d_{struct}, calculated by the number of unmatched edges (marked as black thin edges), in this case, it equals to five. The dashed lines between two nodes (marked as grey nodes) in both cases indicate that those nodes are matched. In addition to calculating unmatched edges, nodes that are absent from the query or a case in the case library are also considered. In Figure 12 the query has Node 149 as white node, which is absent from the retrieved case. All edges linked to this node (thin edges) are mismatched edges and used to penalize the similarity. In addition, if two nodes in one case are directly linked while these two nodes in another case are indirectly linked through multiple nodes, penalty will be assigned. For example, the unmatched edge (n144, n148) showing in the retrieved case but missing in the query indicates the visual reasoning process in the case has a consecutive visit of nodes n144 and n148, but the user looked at n144 and n148 through other nodes, such as n145 and/or n147. Figure 13 shows the top and Figure 14 bottom results based on structure distance d_{struct} component. To provide an easy visualization and comparison, we choose the same query for different distance components in this subsection.

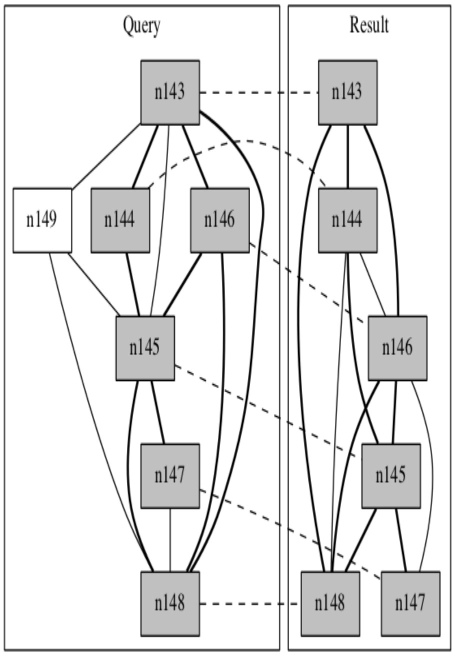


Figure 12: SAGA sub-graph matching result with highlight matching edges and nodes.

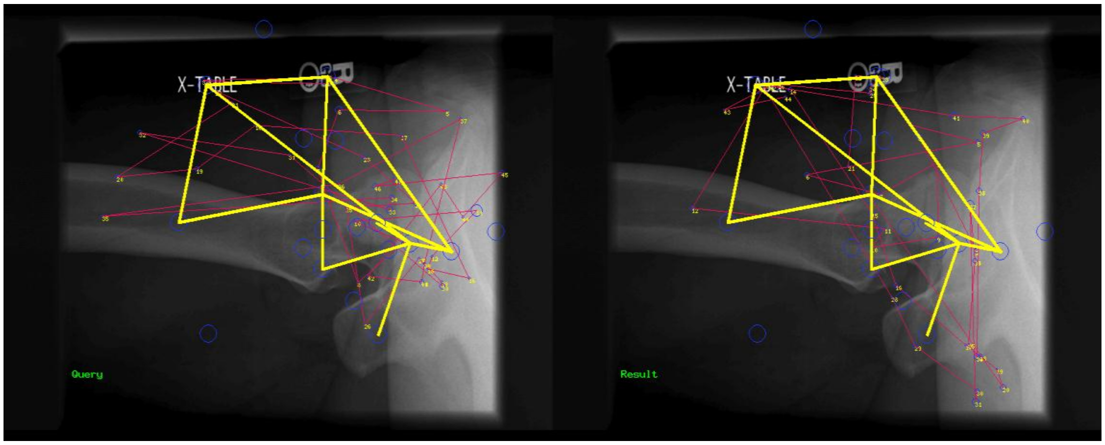


Figure 13: Top matching results for the same query case (right side) based on structure distance component.

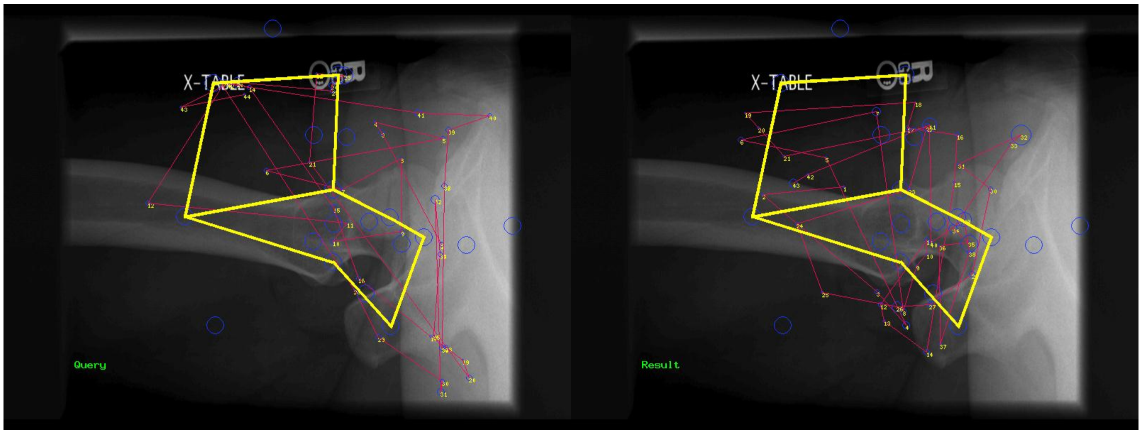


Figure 14: The bottom matched results for the same query case (left image and gaze activities) based on structure distance component.

Analysis of features such as duration of fixation and total dwell time are useful in categorizing the expertise level of the viewer. In expertise studies, it has been found that experts display longer and fewer fixations than novices given the same visual stimuli [99]. Furthermore, experts have been found to more quickly fixate on relevant areas of interest and more quickly identify those patterns which are most important to the task at hand [100]. By retrieving scan paths with similar features, we can use demographic information to "locate" the viewers' relative expertise level on a particular task.

To capture attention differences between two reasoning processes, when a user looks at the same region with different duration and number of visits, we use d_{node} component Equation 24 to measure the distance of corresponding nodes. d_{node} is the weighted summation of two components: gaze fixation attention distance and gaze duration distance, as defined in Equation 26-27.

$$d_{att}(u, \lambda u) = |num\ of\ fixation(u) - num\ of\ fixation(\lambda u)| \\ /total\_fixations\_num(u) \tag{26}$$

$$d_{dur}(u, \lambda u) = |Dwell\ Time(u) - Dwell\ Time(\lambda u)| / \sum_{u} Dwell\ Time(u) \tag{27}$$

The fixation attention distance d_{att} is defined in Equation 26 as the difference in the total number of fixations within two matched nodes. Similarly, the gaze fixation duration distance d_{dur} Equation 27 is the absolute difference in fixation duration in the matched nodes. Figure 15 and Figure 16 show the top and bottom matched results based on node distance d_{node} component. Tables 6 and 7 show the fixation attention and duration's normalized distance (scale to 0-1) for each node. There are two visualization features for each node, namely shading and size for attention level differences. The gray-scale of each node indicates the difference of total number of fixations (visits) on that region between two corresponding nodes with darker circles mean larger differences. The duration differences are represented by the sizes of the nodes with larger sizes mean larger differences. Both contribute to the overall attention difference.

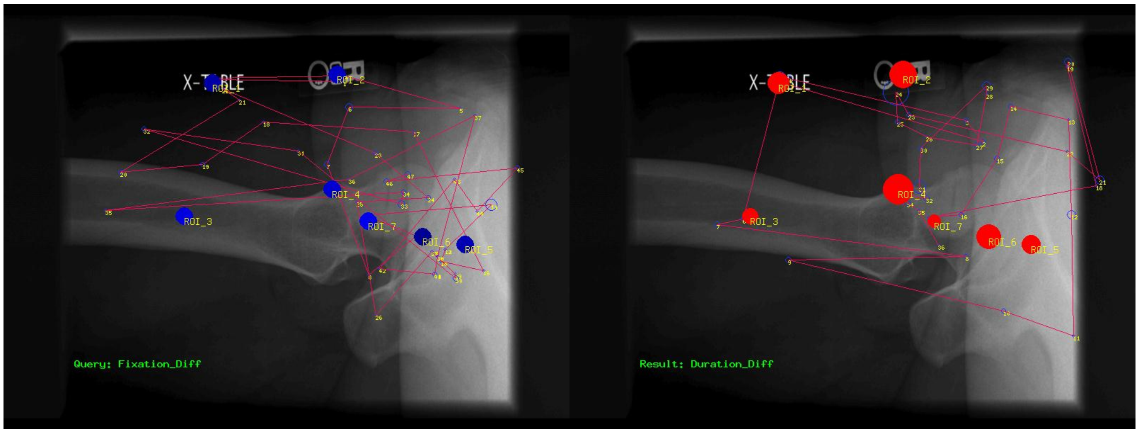


Figure 15: The top matched result (right) for a query case (left) based on fixation attention and duration.

Table 6: Top matched result fixation attention and duration's normalized distance (scale to 0-1).

Node_ID	ROI_7	ROI_3	ROI_5	ROI_1	ROI_6	ROI_2	ROI_4
$\boldsymbol{d_{att}}$	0	0	0.25	0.25	1	0	0
$\boldsymbol{d_{dur}}$	0	0.021	0.325	0.574	0.81	0.893	1

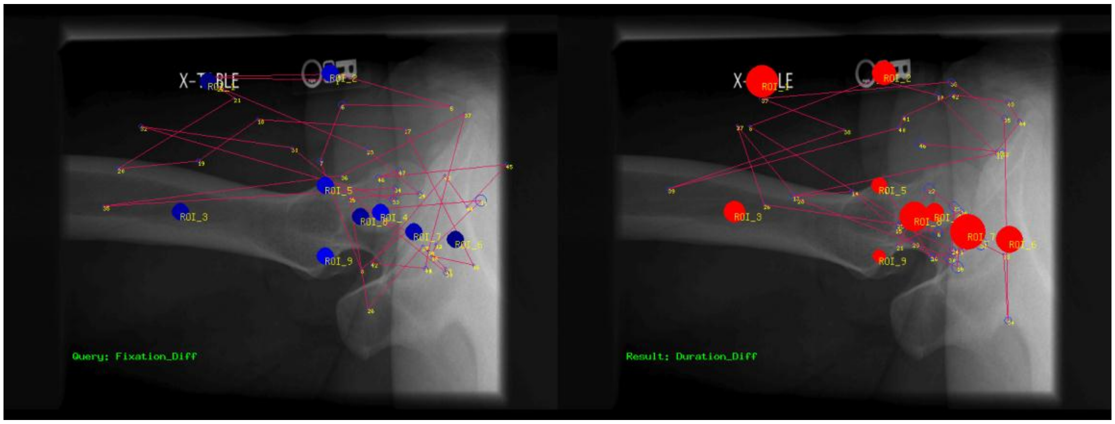


Figure 16: The bottom matched result (right) for a query case (left) based on fixation attention and duration.

Table 7: Bottom matched result fixation attention and duration's normalized distance (scale to 0-1).

Node_ID	ROI_9	ROI_5	ROI_4	ROI_3	ROI_2	ROI_6	ROI_8	ROI_1	ROI_7
$\boldsymbol{d_{att}}$	0	0	0	0.25	0	0.5	0.5	1	0.25
$\boldsymbol{d_{dur}}$	0	0.023	0.048	0.051	0.11	0.125	0.222	0.306	1

Analysis of unmatched nodes is particularly useful when utilizing the case-based reasoning library as a learning aid. As a model of cognition, CBR makes the representation of experiences the primary focus and allows the user to borrow the experiences of others in learning to make decisions. It has been shown that experts have decreased numbers of fixations because they omit fixations on irrelevant structures. By showing users the things that experts deem important to look at, the novice learns which structures are less important and which should be focused on. Likewise, by alerting novices to areas they may have missed, the novice adds an area to their list of things to check.

The d_{node_gaps} component in Equation 25 is the penalty of the unmatched nodes (gap nodes) in the query case. Figure 12 shows one absent node -- n149 in query graph (left side). This component is adapted from original SAGA method and the penalty can be set by domain experts for each individual node. Also, the model allows observers to choose the percentage of the gap nodes allowed in matching result. By setting all $gap_{G1}(u)$ to ∞, users can let the search function display only matches without gaps. Figure 17 shows the least similar result based on node gaps component $d_{nodegaps}$. The gap nodes are represented in red circle shown on the both cases.

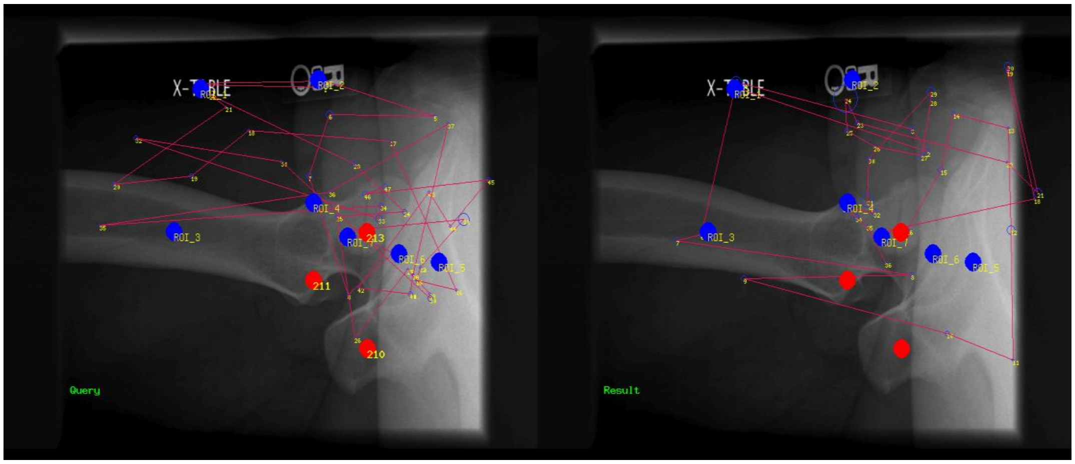


Figure 17: A side-by-side display of a query case (left side) and its least similar matched result (right side) based on node gaps component.

5.2 Temporal Component

As described above, users can retrieve gaze tracking cases sharing similar sub-graph structure with query case by spatial component. By temporal representation – Markov Chain for gaze cases described in Section 3, the system captures temporal similarity. The distance between the query and database cases is calculated using temporal component. Four different distance measurements are used in system for Markov Chain model. The first one is Markov model's path probability Equation 28, calculated by forward algorithm. In Equation 28, $(X_1 = x_1, X_2 = x_2, \ldots X_t = x_t)$ is the mapped ROI sequence from gaze tracking and $P(X_1 = x_1)$ is the initial probability assigned to 1 during calculation. The $p_{x_{t-1}x_t}$ represents transition probability between x_{t-1} and x_t in transition matrix A. The ROI_seq is the query case's gaze fixation sequence. $x_i x_j$ is the pair of adjacent ROI in matching case's ROI_seq. According to Equation 28, the path probability equals to the multiplication of each transition probability $p_{x_i x_j}$.

$$Path\ Pro(A, ROI\_seq) = P(X_1 = x_1, X_2 = x_2, \dots X_t = x_t) \tag{28}$$

$$= P(X_1 = x_1) \times p_{x_1x_2} \times p_{x_2x_3} \times \dots \times p_{x_{t-1}x_t}$$

$$A = \begin{pmatrix} p_{x_1x_1} & p_{x_1x_2} & \cdots & p_{x_1x_m} \\ p_{x_2x_1} & p_{x_2x_2} & \cdots & p_{x_2x_m} \\ \cdots & \cdots & \cdots & \cdots \\ p_{x_nx_1} & p_{x_nx_2} & \cdots & p_{x_nx_m} \end{pmatrix} \tag{29}$$

Moreover, considering the query and retrieved gaze Markov Chain models as two vectors, $\overrightarrow{M_1}$ and $\overrightarrow{M_2}$, in a vector space, system computes the temporal distance by k-Norm distance Equation 30, angle Equation 31 or Kullback-Leibler Divergence Equation 32. In this way, system measures the temporal similarity between query and database cases. Computational results using Markov Forward and Kullback_Leibler distance will be reported in Section 5.

$$\left\|\overrightarrow{M_1} - \overrightarrow{M_2}\right\|_k = \sqrt[k]{\sum_s \left|\overrightarrow{M_1}(s) - \overrightarrow{M_2}(s)\right|^k} \tag{30}$$

$$Angle\left(\overrightarrow{M_1}, \overrightarrow{M_2}\right) = \cos^{-1}\left(\frac{\langle \overrightarrow{M_1}, \overrightarrow{M_2} \rangle}{\sqrt{\langle \overrightarrow{M_1}, \overrightarrow{M_1} \rangle \times \langle \overrightarrow{M_2}, \overrightarrow{M_2} \rangle}}\right) \tag{31}$$

$$H\left(\overrightarrow{M_1}, \overrightarrow{M_2}\right) = \sum_s \overrightarrow{M_1}(s) \log \frac{\overrightarrow{M_1}(s)}{\overrightarrow{M_2}(s)} \tag{32}$$

5.3 Case Adaptation

By case representation and retrieval functions, users can retrieve gaze cases sharing similar spatial (sub-graph structure) and temporal (MC model) patterns from the case library. After CBR retrieval, the system allows users to either directly reuse retrieved results as solutions to the current problem (such as assisting their diagnostic or image interpretation) or choose the best fitting case(s) for further adaptation. In the reusing and revising step, the retrieved gaze cases are adapted using the query case by reconstructing its temporal Markov Chain model. Each retrieval case's MC model will be reformed, or adapted, based on the query's MC model.

The first step of this revising is combining the query and chosen cases' MC transition matrices. By Equation 33, system calculated a new transition counting matrix for the adapted MC model. Once the new counting matrix was obtained, the next step is the normalization of the obtained counting matrix and calculation of the transition matrix by equations 3-4, as described in section 4.2.

$$Adapt\left(A_{ij}\right) = Query\left(A_{ij}\right) + Result(A_{ij}) \qquad (33)$$

Figure 18 shows an example of an adapted MC model created by combining query and result MC models. The two Markov Chain models for query (top) and result (bottom) cases are shown on the left side and the adapted result is displayed at right. By combining features of both MC models, CBR adapts to user-selected results to better fit the query.

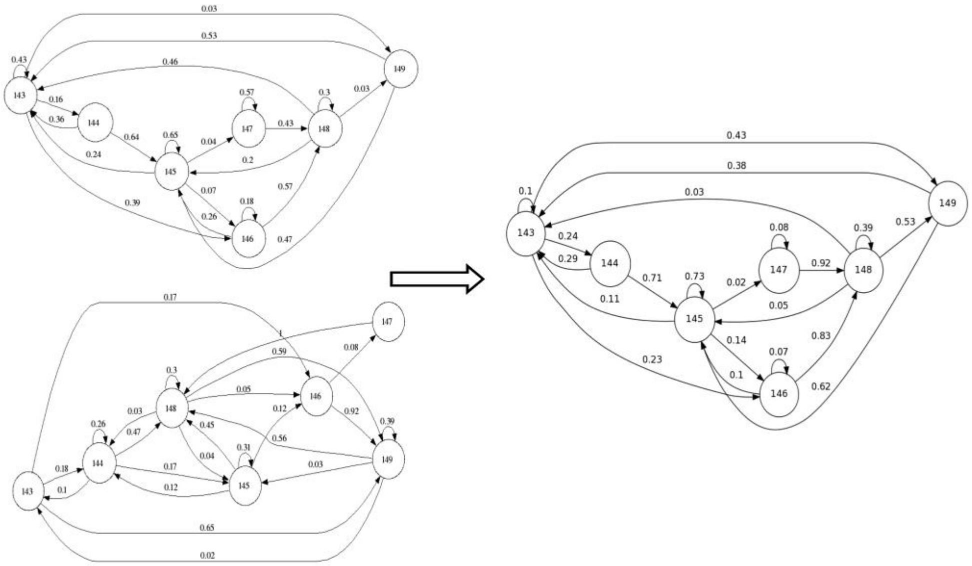


Figure 18: An example of Markov chain model adaptation result.

Once the adapted model is generated, the last step is to revise the adapted case (if necessary) and retain it in the case base. The CBR allows user to update the transition probability in adapted MC model. Finally, system retains the solution and the query cases to the case base for future use. In this way the CBR adapts to user needs in order to return more accurate results. Additionally, CBR can generate a simulated gaze sequence based on the adapted gaze tracking Markov Chain model, and this simulated gaze sequence is used to model search behavior based on user-specific tasks.

CHAPTER SIX

HUMAN SUBJECT EXPERIMENTS AND RESULTS

To ensure the effectiveness of the distance measurements in understanding the visual reasoning process for case retrievals, we conducted two human subject experiments. Both experiments used Radiology medical X-ray images in 10 level knowledge pyramid structure as shown in Table 1. Appendix A shows two image datasets for two experiments. For each experiment's configuration, subjects (domain experts or students) were asked to first look at the question and then showed them images. At the same time, their gaze tracking raw sequence data was recorded through a gaze tracker device. The subjects don't need to answer the question or speak aloud anything during the image scan process. After they finished whole images (total 13 images for experiment 1 and 8 images for experiment 2), we asked them to speak out their thinking process through a think-aloud protocol and recorded the audio for later usage. Table 8 shows the 10 knowledge levels and associated questions in experiment configuration 1 in the following section. Comparing to experiment configuration 1, we selected three knowledge levels in this experiment, level 2, 5, and 8. Table 9 shows the question in experiment configuration 2.

Table 8: 10 knowledge level and associated question in experiment configuration 1.

Level	Original Visual Knowledge Pyramid	Questions
1	Type Technique	What is the modality of this image?
2	Global Distribution	Describe the overall photographic properties of this image.
3	Local Structure	What basic textual elements do you identify on this image?
4	Global Composition	How do you orient yourself to this image?
5	Generic Objects	What body part does this image demonstrate?
6	Generic Scene	What is the projection of this image?
7	Specific Objects	Identify 3 foreign objects on this image
8	Specific Scene	Evaluate the positioning of this image.
9	Abstract Objects	Describe this patient based on what you see in this image
10	Abstract Scene	What problem(s) do you think this patient has?

Table 9: an example of question and knowledge level in experiment configuration 2.

Level	Original Visual Knowledge Pyramid	Questions
General	General	Please evaluate the positive and negative aspects of this image.
2	Global Distribution	Was the amount of x-rays used for this image adequate?
5	Generic Objects	Does the image demonstrate all of the required anatomy?
8	Specific Scene	Are the relationships between the anatomical structures accurate?

6.1 Experiment configuration 1

In the first experiment, we developed a webcam based eye-tracking tool which we used to capture eye movement scan tracings from 39 subjects including radiologic technologists (15), senior students (11) and junior students (13) in radiography department. Figure 19 shows the distribution of gaze cases on each domain image. A modified OpenGazer software [101] was chosen to collect a series of screen coordinates as eye-tracking raw data in our tool. The OpenGazer system is unobtrusive, as the users only need to sit in front of the computer with center webcam and look at a domain image shown on the screen. This system much better mimics the natural clinical environment, enabling visual trace data collection without any constraints such as wearing head gear. The domain images included 13 X-ray images with corresponding questions for data collection. Each image was chosen based on different questions in the modified 10-level visual structure in the work of Jaimes and Chang. The questions were asked before the user saw the domain image. Users were given 15 seconds to consider each image. During this time, users' eye traces were captured by webcam and assigned coordinate positions by the OpenGazer software.

We tested the distance model on three different knowledge groups: junior students, senior students, and experts. The subject study of this experiment was approved by the University of Missouri Health Science Internal Review Board (IRB #1172279). Our hypothesis is that user's visual reasoning process within the same knowledge group should have smaller variations than those in other groups using a specific distance function if such a function is able to provide relevant distinction across multiple groups. We performed the following group comparisons: Expert vs. Expert (EE), Expert vs. Senior (ES), Expert vs. Junior (EJ) and Expert vs. Novice (EN). These comparisons were performed using all-against-all distance measurement between and within groups. The novice group combines junior and senior students. The Expert vs. Expert (EE) distances allow us to identify the level of internal consistency within the expert group. The Mann Whitney U test [102] was used for null hypothesis testing on combinations of these four different group comparisons to obtain the *p*-value tables. The *p*-value less than 0.05 indicates that two distance distributions are statistically different.

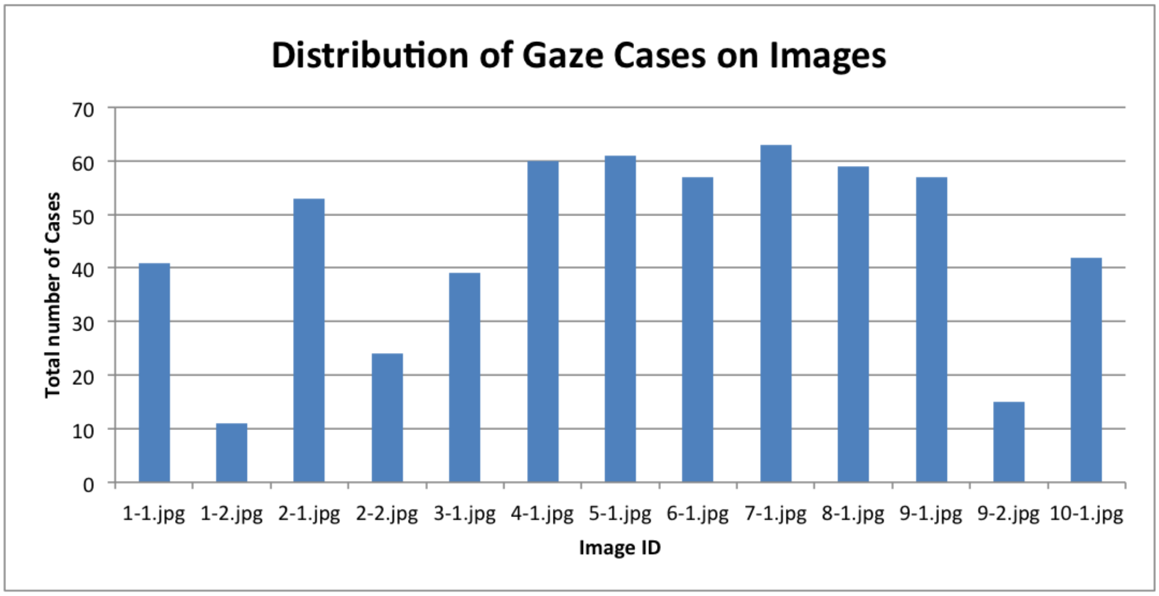


Figure 19: Distribution of gaze cases on each domain image.

6.1.1 Spatial Component

To test the effectiveness of the three components of the spatial distance (Eq. 6) individually and compositely, we tested cross expertise group comparisons. Table 10 lists the results for individual spatial components. Based on three separate spatial subcomponents - node gap, structure distance and node distance, the ratios of cases with statistical significance using U-test between groups are between 29% and 69%. The overall spatial components are essential to provide an aggregated spatial distance that can distinguish subjects from different knowledge group based on their gaze tracking. From Table 11, we observe that experts' visual knowledge pattern is different from novice (junior and senior students) based on aggregated spatial component analysis while there is no significant differences between junior and senior students from the novice group on most tasks especially when dealing with high-level tasks, such as level 10. We also see that both expert and novice groups do not have significant differences for image 1-2 (task 1) and 2-

1 (task 2). For image 1-2 (task 1), one of the reasons is inefficient gaze tracking data on the image for statistical analysis as Figure 19 shows. For image 2-1 (task 2), the density of hydrogen atoms in the tissue controls the brightness of the signal, where tissues with greater proportions of hydrogen appear brighter or white on an image. So, when a technologist looks at a cross-sectional image, they should look at the subcutaneous fat and note whether it is dark or bright to determine if the image was created by CT or MR. Thus, from our results, based on spatial component, there is no difference between novice and expert group on image 2-1. In the junior and senior groups, we can see that at low-level knowledge tasks (Levels 1-4), there is differences between junior and senior students' gaze pattern comparing to export's. But for most of high-level knowledge tasks (Levels 5-10), senior and junior students perform similarly, which can be explained by the facts that both subgroups in the novice may lack of needed knowledge to answer those questions.

Table 10: The number of tasks (13 images across 10 knowledge levels for all subjects) having significant differences between two knowledge groups using separate spatial sub-components and aggregated component.

		EE/EN	EE/EJ	EE/ES	EJ/ES
Spatial Subcomponents	Node Gap	3	3	3	5
	Structure Distance	8	7	7	6
	Node Distance	9	6	6	4
Aggregated Spatial Component		11	10	10	4

Table 11: p-value for Mann-Whitney test on overall spatial distance component α=0.05.

			Knowledge Groups			
		Image	EE/EN	EE/EJ	EE/ES	EJ/ES
Visual Tasks	Task_1	1-1	**3.63E-09**	**1.22E-12**	**0.00266**	**3.83E-07**
		1-2	1.00000	0.78396	0.88521	1.00000
	Task_2	2-1	0.82325	0.86489	0.81736	0.88572
		2-2	**3.70E-05**	**2.91E-05**	**0.00072**	0.69306
	Task_3	3-1	**8.98E-05**	**6.69E-06**	**0.00610**	**0.01881**
	Task_4	4-1	**1.54E-05**	**9.99E-11**	0.22811	**5.86E-11**
	Task_5	5-1	**0.03406**	0.07367	**0.03984**	0.69921
	Task_6	6-1	**6.85E-16**	**8.95E-12**	**1.42E-14**	0.65600
	Task_7	7-1	**1.50E-05**	**5.86E-06**	**0.00039**	0.09678
	Task_8	8-1	**0.00077**	**0.00370**	**0.00083**	0.49800
	Task_9	9-1	**6.06E-06**	**0.00468**	**3.33E-06**	0.26815
		9-2	**9.04E-09**	**1.42E-10**	**0.00013**	**0.00061**
	Task_10	10-1	**0.01110**	**0.01861**	**0.02046**	0.72867

6.1.2 Temporal Component

Same as the spatial distance testing, after obtaining four different types of spatial distance, Mann-Whitney U test is applied to the temporal component. Table 12 is total number of tasks having significant differences using U test based on Markov Forward Distance, k-norm, Angle, and Kullback-Leibler. We found that Markov Forward distance fails to distinguish most tasks and concluded that fixation order is not important when comparing different knowledge group gaze tracking cases. However, Kullback-Leibler distances show the significant differences within the 10 tasks. Furthermore, 2-norm and Kullback-Lerbler measurements have similar results which better than angle measurements. Table 13 shows *p*-value calculated by Kullback-Leibler measurement. Comparing to spatial component analysis, we have similar observations from the experiments using temporal component. Same as Table 11, for image 1-2 (task 1), both expert and novice groups do not have significant differences. One of the reasons is inefficient gaze tracking data on these two images for statistical analysis. Table 13, for image 9-1 (task 9), shows that there is no significant difference between knowledge groups using ROI visiting order only for both novice and expert for image understanding. The image is about the patient as female, lacking muscle tone with clear lungs, decreased bone density, indicating possible osteoporosis, and normal heart size. The decreased bone density and lack of muscle tone indicate that the patient is likely older. For the novice group, junior students who have one year less of training perform closely to the senior students for most of the tasks. This could be explained as ROI visiting order may not be important for both novice students for image understanding. On the other hand, experts perform differently compare to both junior and senior students. With well-trained knowledge, expert can quickly locate interesting regions in the images and perform quite differently from novice students in temporal visual activities.

Table 12: The number of task (total 13 images) having significant differences between two knowledge groups.

		EE/EN	EE/EJ	EE/ES	ES/EJ
Temporal Distance Measurement	Markov Forward	3	3	3	3
	k-Norm	11	11	10	8
	Angle	4	5	6	7
	Kullback-Leibler	12	11	12	5

Table 13: p-value for Mann-Whitney test on temporal Kullback-Leibler component.

			Knowledge Groups			
		Image	EE/EN	EE/EJ	EE/ES	EJ/ES
Visual Tasks	Task_1	1-1	**3.19E-05**	**3.15E-06**	**0.00811**	**0.03633**
		1-2	0.37553	0.54945	0.34461	0.19940
	Task_2	2-1	**0.00273**	**0.01984**	**0.00214**	0.51729
		2-2	**0.00253**	**0.01066**	**0.00423**	0.70261
	Task_3	3-1	**1.87E-08**	**1.01E-09**	**1.74E-05**	**0.00116**
	Task_4	4-1	**2.71E-07**	**6.79E-08**	**8.48E-05**	**0.01666**
	Task_5	5-1	**1.13E-05**	**1.77E-07**	**0.00335**	**0.01216**
	Task_6	6-1	**9.86E-14**	**1.36E-08**	**2.29E-14**	0.36494
	Task_7	7-1	**3.74E-10**	**1.00E-11**	**9.47E-07**	**0.00903**
	Task_8	8-1	**3.68E-12**	**2.30E-10**	**4.54E-11**	0.50250
	Task_9	9-1	0.07199	0.09109	0.09715	0.30718
		9-2	**0.00331**	**0.00107**	**0.04968**	0.14318
	Task_10	10-1	**0.02123**	0.05590	**0.02003**	0.55283

6.2 Experiment configuration 2

After the first experiment using a webcam-based gaze tracker, we designed a new experiment, and collected data on lumbar spine chest X-ray images using an infrared gaze tracker adapted from GazeGroup lab in IT University of Copenhagen [71]. This tracker can be also found in their recent commercial product, THEEYETRIBE. The new infrared gaze tracker can reach between $0.5^0 - 1^0$ accuracy with 30Hz and 60Hz sample rates. The latency at 60Hz is less than 20ms. When compared to the first webcam gaze tracker, the new infrared setup has several advantages. First, performance does not depend on ambient brightness, since only infrared light can pass through the lens. This makes this device useful even when the light conditions are bad, as may be the case in some laboratory setups. Second, the infrared camera can capture clearer eye pupil images than the webcam setup. It can produce better eye images for later use with the segmentation algorithm for pupil identification. Moreover, the infrared setup can reach much higher accuracies $< 5^0$ than normal webcam setups, and also has better consistency.

The infrared setup includes a high-speed camera, an infrared lens with infrared filter, and two infrared lighting resources. Figure 20 is a demonstration of the gaze tracker setup. As a key to the figure below, 1 is domain image displayed in a monitor, 2 is an infrared camera, 3s are two infrared lights, and 4 is a base for holding the eye tracker.

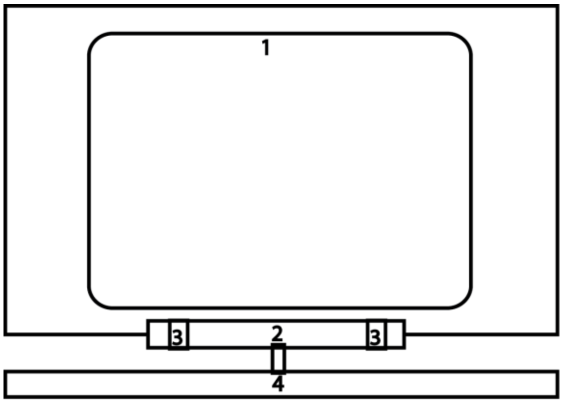


Figure 20: A demonstration of infrared gaze tracker setup.

In this experiment, a total 25 subjects were recruited from Radiology department 12 junior students and 13 senior students. The subject study of this experiment was approved by the University of Missouri Health Science Internal Review Board (IRB #2001653).

6.2.1 Spatial Component

We used Spark GraphX for spatial analysis of the large-scale gaze tracking data collected in Experiment Configuration 2. Each gaze tracking fixation data was first mapped to a Region of Interested (ROI) on the domain image. Based on the spatial relationship between ROIs, we generated a graph model for each gaze tracking case in Spark. As

Figure 11 shows, each gaze tracking graph model includes a set of edges and a set of vertices. In this case, the vertex is each anatomical feature in the domain image, such as the sacrum, T12 vertebral body, pelvis, etc. Each edge represents the transition between two anatomical features. To improve performance, the program was tuned such that it caches both query and retrieval graphs in order to quickly calculate the similarities. After calculations are complete, the retrieval graph is removed from the RAM and the next retrieval graph is cached in. By default, all the graphs will be cached in RAM until memory limitation and force them to leave the RAM.

6.2.2 Temporal Component

6.2.2.1 Visual Action segment results

The first step in a temporal model is to segment and annotate the raw fixation sequence into visual actions for all junior and senior knowledge groups. With a manually-labeled training dataset, we used linear chain CRF model and traditional classifiers (Logistic, Naïve Bayesian, and SVM) to auto-annotate visual actions based on five proposed gaze features (duration, dispersion, speed, anglespeed, and acceleration). To train the linear chain CRF model, we used Stochastic Gradient Descent with an L1 regularization algorithm. Among the traditional classifiers, we found that a logistic model gave the best accuracy. To evaluate the prediction results, with limited labor time, we randomly sampled 14 gaze tracking cases as our testing dataset. Figure 21 shows the distribution of testing dataset across different assessment questions. The average CRF accuracy is 74.4%, which shows the difficulty in predicting visual actions across different task and users using a linear chain temporal model. The average of logistic regression accuracy, however, is 91.7%, which indicates that this approach provides much better prediction results.

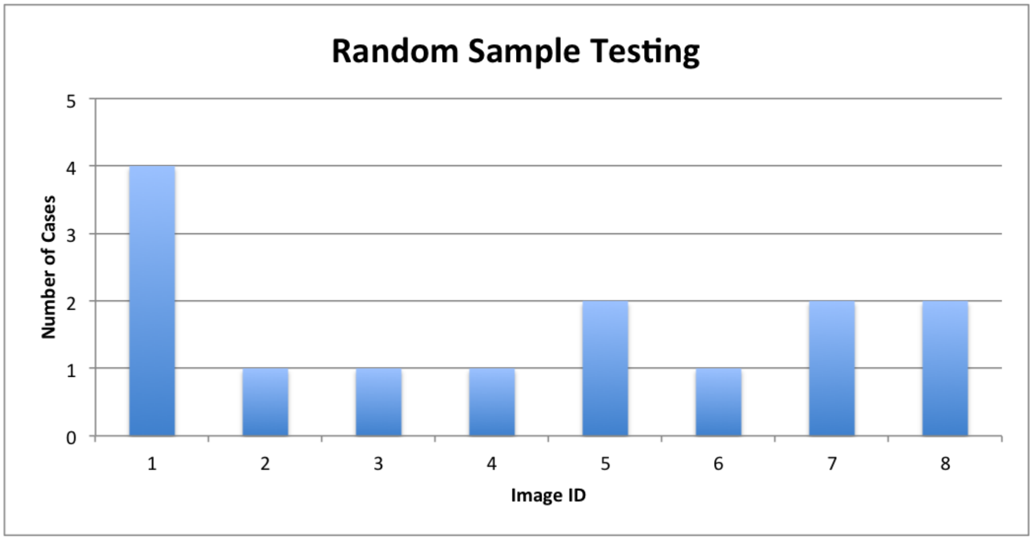


Figure 21: The distribution of testing dataset across different question.

Figures 23-29 show the Logistic Regression annotation case study on three different knowledge questions and overall questions for both junior (left side) and senior (right side) students. Three visual actions were highlighted in different colors on the image, visual action focus is green, read action is yellow, and scan action is blue. Figure 22 displays predicted results for the overarching question: “Please evaluate the positive and negative aspects of this image”. Table 14 and Table 15 are the corresponding confusion matrix for the annotation results in Figure 22, where Table 14 presents juniors’ results (96.8% accuracy) and Table 15 presents seniors’ results (88.4% accuracy).

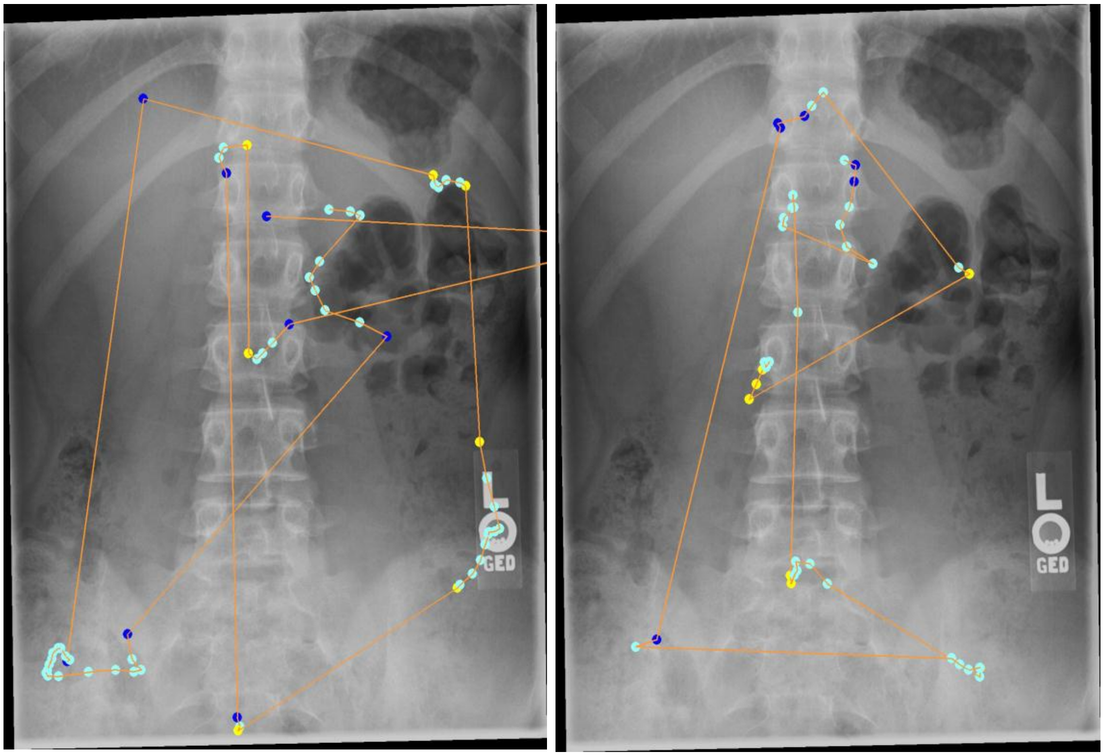


Figure 22: An example of CRF model prediction results on first Overall question “Please evaluate the positive and negative aspects of this image”. Junior student result is on the left and senior’s is on the right. Each predicted visual action is marked as a highlight color. Action focus is marked as light blue, read action is yellow and scan action is blue.

Table 14: Corresponding confusion matrix for visual action prediction for junior student on Figure 22.

Predicted Label Ground Truth	Focus	Scan	Read
Focus	44	1	0
Scan	0	10	0
Read	1	0	6

Table 15: Corresponding confusion matrix for visual action prediction for junior student on Figure 22.

Predicted Label Ground Truth	Focus	Scan	Read
Focus	28	0	1
Scan	0	5	2
Read	2	0	5

Figure 23 with same question as Figure 22. Both junior and senior students focus on Marker, and Sacroiliac Joint. However, junior student focused more on the L1 and L2 vertebral bodies, while senior student tended to focus on T12 vertebral body. Moreover, junior student focused more on the L4 and L5 vertebral bodies, while senior student tended to check the sacroiliac joints. Difference in those visual actions indicates that junior and senior students tend to use different strategies to interpret domain images on open general questions.

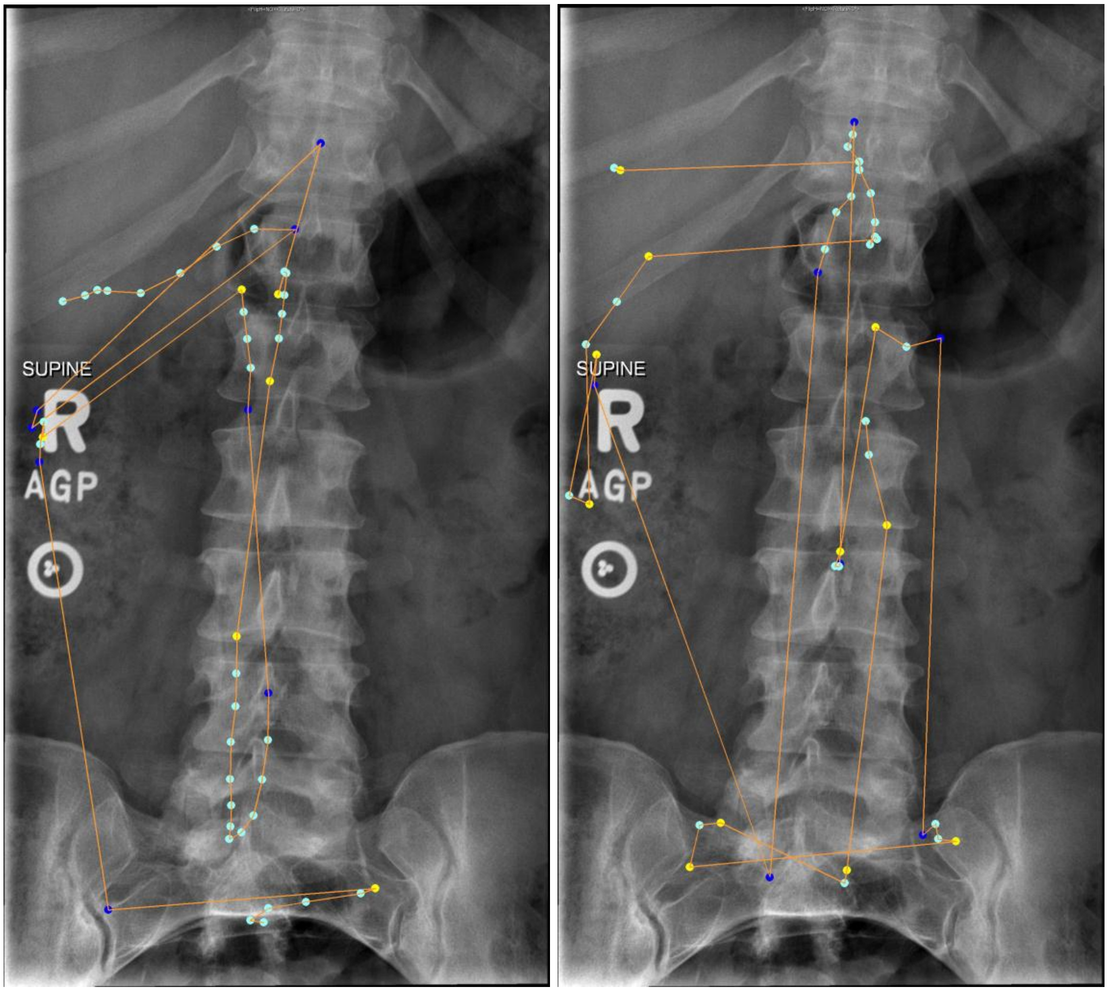


Figure 23: An example of CRF model prediction results on last Overall question "Please evaluate the positive and negative aspects of this image" where junior result is on the left and senior's is on the right. Each predicted visual action is marked as a highlight color. Action focus is marked as light blue, read action is yellow and scan action is blue.

The question for visual knowledge 2 – Global Distribution is designed as following: Was the amount of x-rays used for this image adequate for junior (left) and senior (right). Two Lumbar Spine X-ray images were chosen to represent two different cases: Figure 24 represents a case where there was too much x-ray exposure, resulting in an image that is too dark. Figure 25 is an opposite case, in which the amount of x-rays is not enough and the image is too white and grainy. As we can see from both Figure 24 & 25, instead of focusing on the vertebral bodies, both junior and senior students tended to check the psoas muscle and ascending/descending colon. In contrast to junior students, senior students spent more time on the sacroiliac joint. These anatomic regions need to be clear in order to decide whether the x-ray exposure is correct.

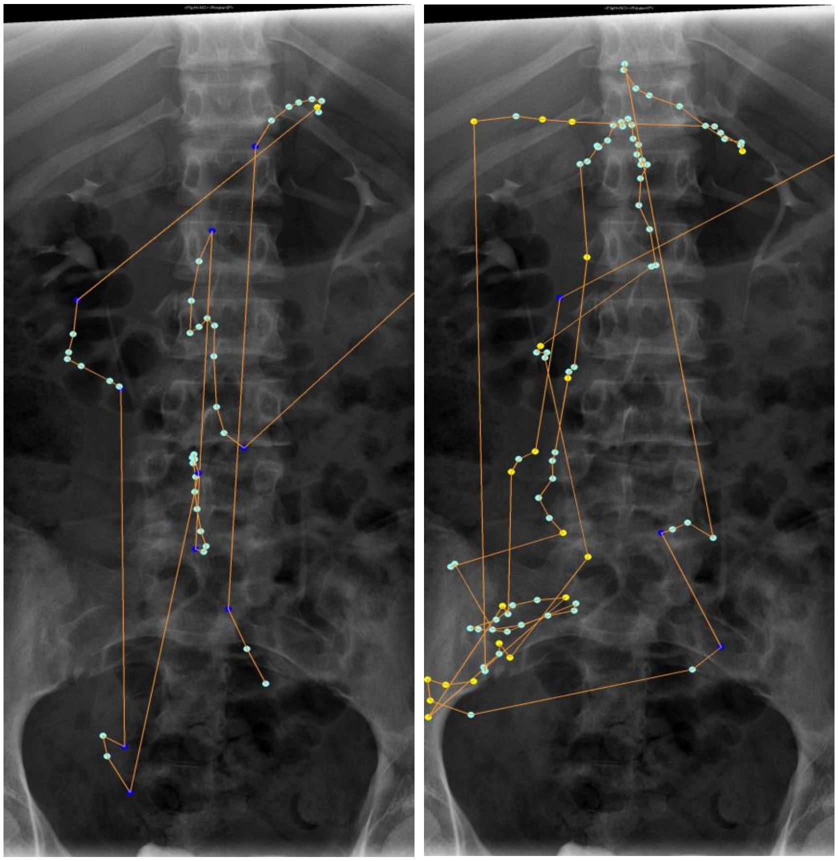


Figure 24: An example of CRF model prediction results on knowledge level 2 – Global Distribution with the question "Please evaluate the positive and negative aspects of this image". The junior results are on the left and senior results are on the right. Each predicted visual action is marked as a highlight color: action focus is light blue, read action is yellow and scan action is blue.

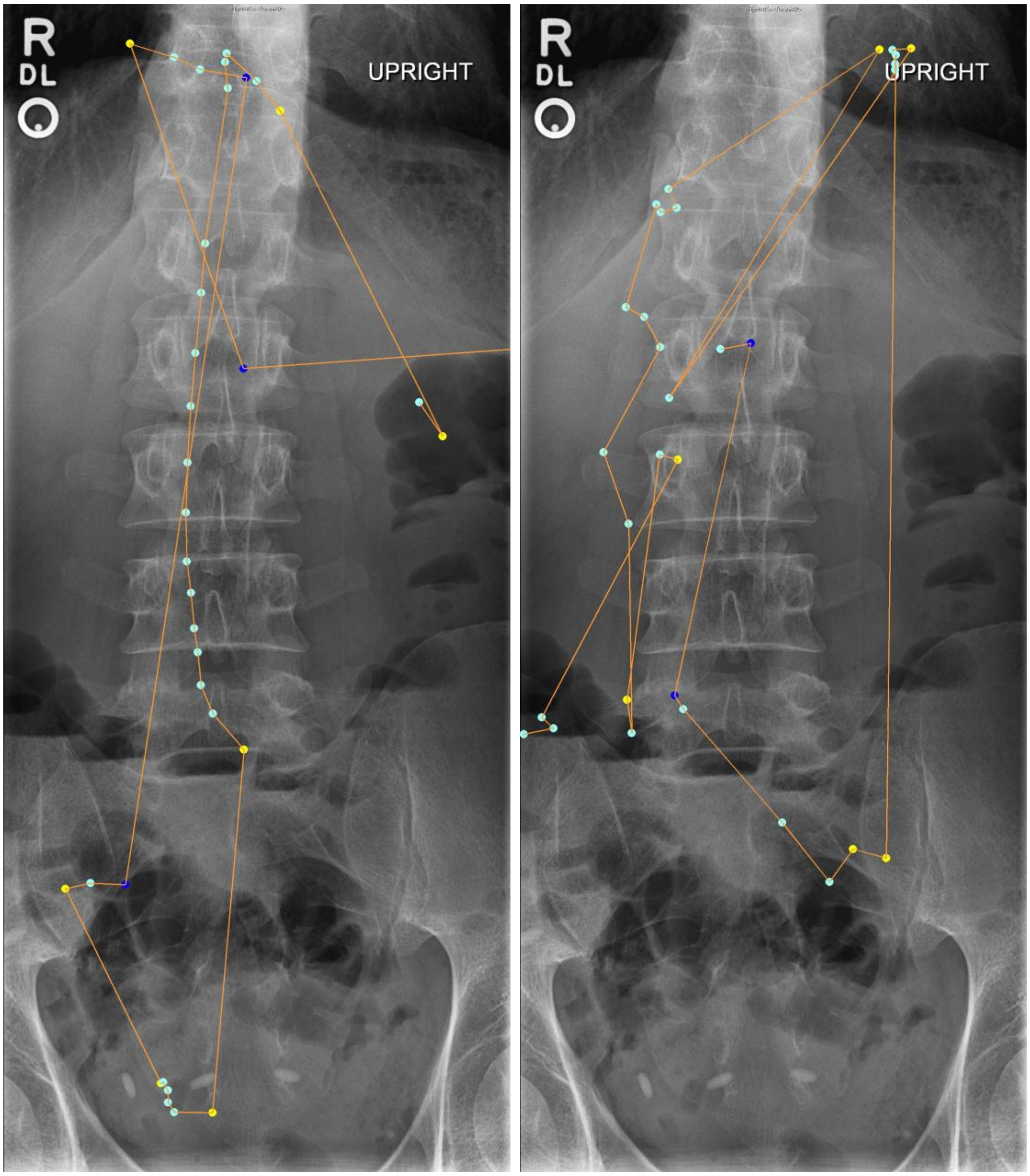


Figure 25: An example of CRF model prediction results on knowledge level 5 – Generic Objects with the question “Does the image demonstrate all of the required anatomy” where junior results are on the left and senior results are on the right. Each predicted visual action is marked as a highlight color. Action focus is light blue, read action is yellow and scan action is blue.

The question for knowledge level 5 - Generic Object is designed as “Does the image demonstrate all of the required anatomy”. To answer this question, students need to have more domain knowledge that when they completed the Level 2 question (above). As we can see in Figure 26, both junior and senior students were tracing on the spine column (L5-T12). This action was found from both junior and senior but with different visual action, junior students scanned through the spine, whereas seniors spent more time using focus and read actions. Moreover, senior student (right) tended to focus on the ala of sacrum and the L5 vertebral body comparing to juniors (left).

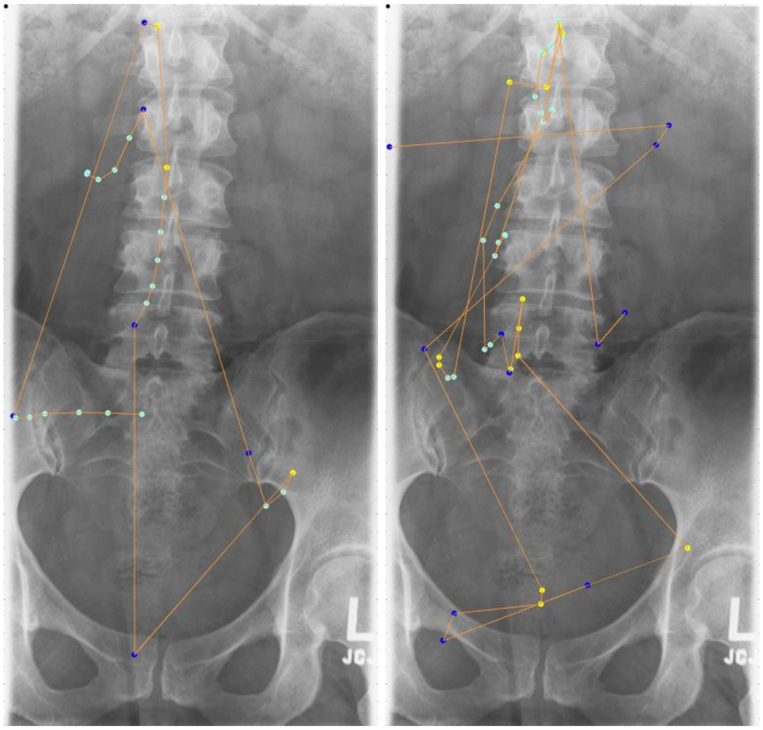


Figure 26: Another example of CRF model prediction results on knowledge level 5 – Generic Objects with the question “Does the image demonstrate all of the required anatomy” where junior results are on the left and senior results are on the right. Each predicted visual action is marked as a highlight color. Action focus is light blue, read action is yellow and scan action is blue.

The question for visual knowledge level 8 – Specific Scene is “Are the relationships between the anatomical structures accurate”. Students are expected to trace and compare the vertebral bodies (L5-T12) and the sacrum in order to answer this high visual level question. Demonstrated in Figure 27, senior students perform close to what was expected, while juniors tended to focus on lower part – Sacrum. From Figure 28, junior students have clear pattern on tracing and comparing vertebral bodies, while senior students tended to compare T12 with L1 and L3 with L4 vertebral bodies. Both students focus on the marker on the lower left corner.

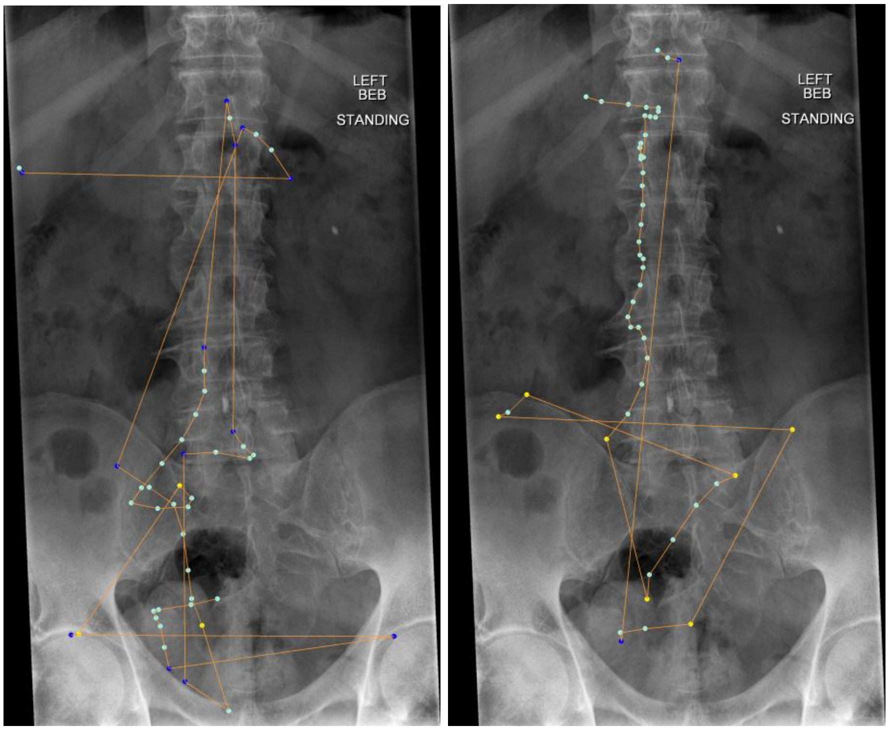


Figure 27: An example of CRF model prediction results on knowledge level 8 – Specific Scene with the question “Are the relationships between the anatomical structures accurate” where junior results are on the left and senior results are on the right. Each predicted visual action is marked as a highlight color. Action focus is light blue, read action is yellow and scan action is blue.

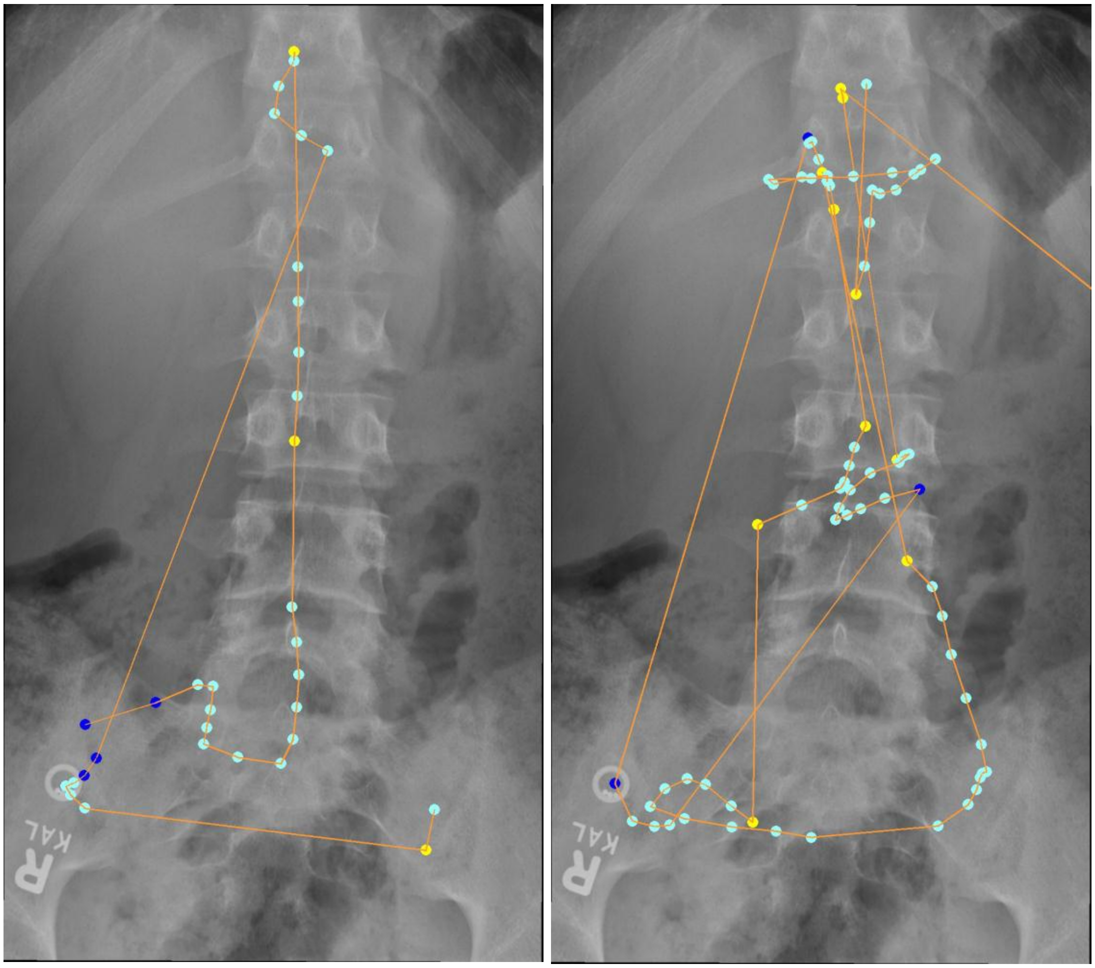


Figure 28: Another example of CRF model prediction results on knowledge level 8 – Specific Scene with the question “Are the relationships between the anatomical structures accurate” where junior results are on the left and senior results are on the right. Each predicted visual action is marked as a highlight color. Action focus is light blue, read action is yellow and scan action is blue.

For further analysis, based on these basic three visual actions, we can posit that more actions represent more complex reasoning, such as tracing action for tracing on lines or comparison action for comparing different anatomies.

6.2.2.2 Markov Decision Process (Reinforcement learning result)

Based on Equations 17 and 18, Markov Decision Process (MDP) models were generated for single gaze tracking case (single user with single image), aggregated user level (single user with all eight images), and aggregated group level (all users in the junior or senior group with all eight images). Figure 29 shows an example of a single gaze tracking MDP model with the order equal to 1 (meaning that the current observation is dependent only on the previous one). From Figure 29, we can see that each state is an anatomic region. In this case, we have total four states: T12 vertebral body, L1 vertebral body, L3 vertebral body, and L4 vertebral body. Additionally, three action nodes are defined: Focus, Read, and Scan. The reward function R(s, a) is calculated using Equation 18 on each 'state->action' edge. For example, on edge 's3->a0', there is '+0.85' reward, which means that the user can gain 0.85 reward if they use a Focus action on state 3 – L4 vertebral body. Moreover, the state-action transition probabilities are calculated by Equation 17. The transitions between states are passing through an action node. Each transition probability is marked on 'action->state' edges. For example, from state s1 to s0 through action a0, the transition probability is 0.67.

To efficiently update and learn this MDP model when dealing with real time streaming data (new gaze tracking fixation comes in), we keep counts for both numerator and denominator terms in Equation 17 and Equation 18. In Spark streaming, we used a buffer RDD (Resilient Distributed Dataset) to keep accumulating these two counts when new knowledge (gaze tracking data) became available. Then, by computing the ratio of these counts, we were able to estimate both transition probability and reward in real time.

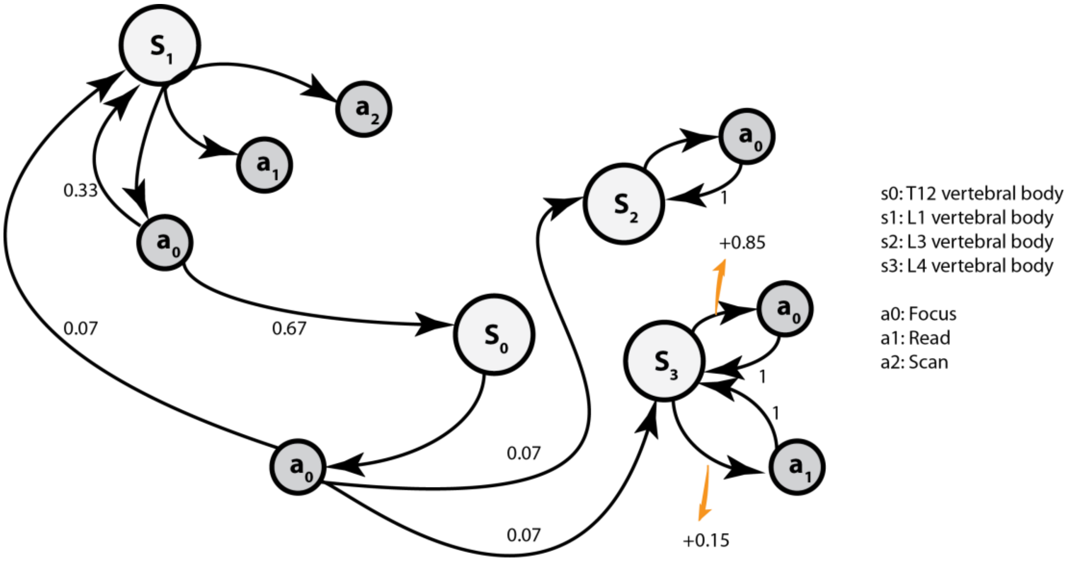


Figure 29: An example of single gaze tracking Markov Decision Process model.

We ran the large scale data analysis on an elven node Spark cluster. Each node has a Xeon CPU – 3.2GHz with 32GB RAM and 6TB of disk spaces. The simulated cases were generated and stored in HBase from the original 200 cases, and for increasing numbers of cases (2,000, 20,000, 200,000, 2,000,000, 20,000,000, 200,000,000, and 2,000,000,000). HBase is a distributed database on top of HDFS. Figure 30 and Figure 31 show the running time and total data storage size for different numbers of cases. From these results, we can see that with the increased number of cases, the running time increased. Despite this, for samples of up to 200,000 cases, the searching can be done in real time (<5s).

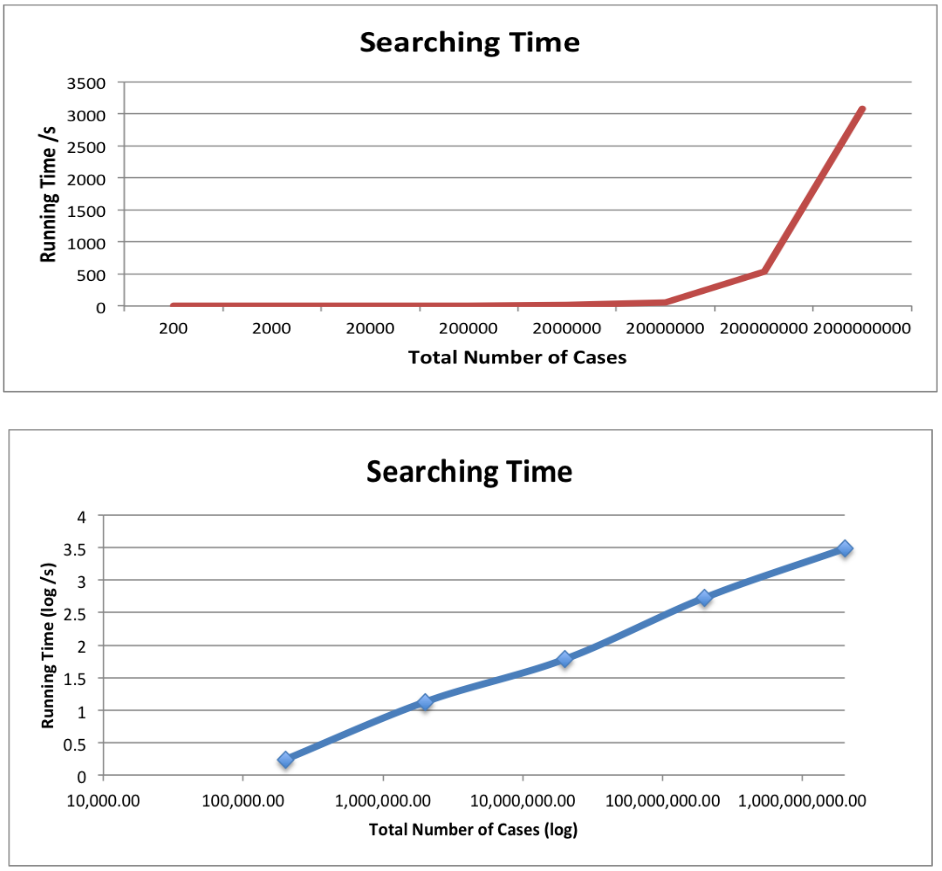


Figure 30: The running time on large scale simulation analysis.

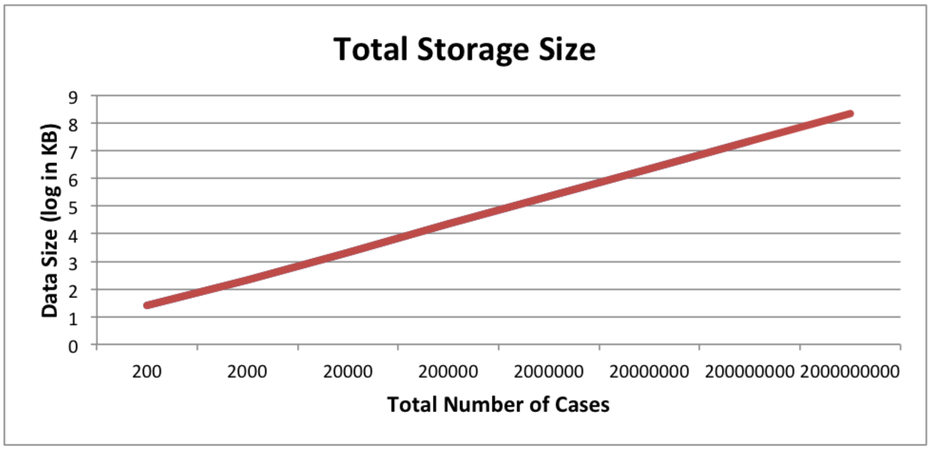


Figure 31: The storage size on large scale simulation analysis.

The simulation cases were further analyzed based on proposed similarity measurement. Table 16 shows the accuracy analysis on different variance of large scale simulation cases using proposed MDP similarity measurement. The same results are also shown in Figure 32 which clearly demonstrates that "remove" action's results have much lower accuracy than "add" actions'. From the table, we can see that with increased random difference (adding, removing or combining two), the accuracy decreased. On the large scale 200,000 cases sample, even on top 3000 results, we still can reach more than 90% accuracy. For original 200 cases' variance, the "remove ROI" variance gives the worst accuracy whereas "add state" variance can still reach 90% accuracy. In other word, the proposed similarity measurement is more sensitive on missing salient region than accidental observe new regions. The reason of it can be explained as below. For simplicity, we consider the original case represents as 4 dimension vector $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$, where $x_2 = 0$. The "add state" simulation case is represented as $\begin{bmatrix} x_1 \\ x_2{}' \\ x_3 \\ x_4 \end{bmatrix}$, where $x_2{}' <> 0$. The "remove state" simulation case is represented as $\begin{bmatrix} x_1 \\ x_2 \\ x_3{}' \\ x_4 \end{bmatrix}$, where $x_3{}' = 0$. Assuming "remove state" and "add state" variance generate the same difference comparing to original case, Figure 33 shows the differences of similarity score from two variances. From the figure, we can see that "remove state" variance' distance score has larger changes comparing to "add state" due to the square root function derivative differences (h1>h2). Thus, the "remove state" variance has more impact on changing distance score and has worse accuracy than "add state" variance.

Table 16: The accuracy analysis on controlled variance simulation cases.

	200	2000	20,000	200000	
	Top 10	Top 100	Top 1000	Top 1000	Top 3000
Add One State	0.9	0.9	0.894	1	1
Add Two States	0.9	0.87	0.85	1	0.999333
Add Three States	0.9	0.84	0.809	1	0.996667
Remove One State	0.6	0.71	0.692	1	0.989667
Remove Two States	0.6	0.65	0.639	1	0.938333
Remove Three States	0.5	0.52	0.532	1	0.896667
Combination 1	0.9	0.76	0.733	1	0.939333
Combination 2	0.6	0.61	0.59	1	0.912333
Combination 3	0.7	0.59	0.53	0.997	0.861667

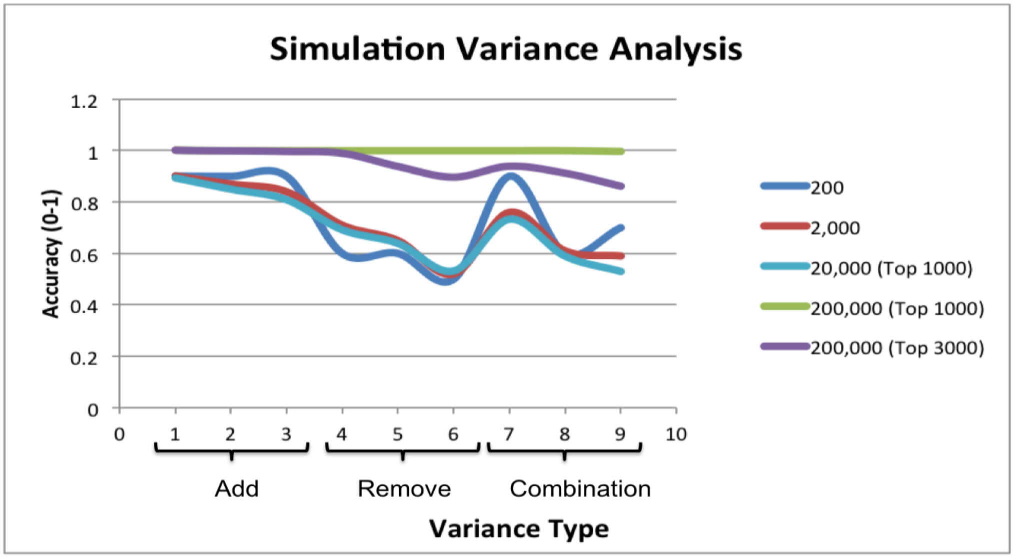


Figure 32: Simulation variance retrieval result analysis.

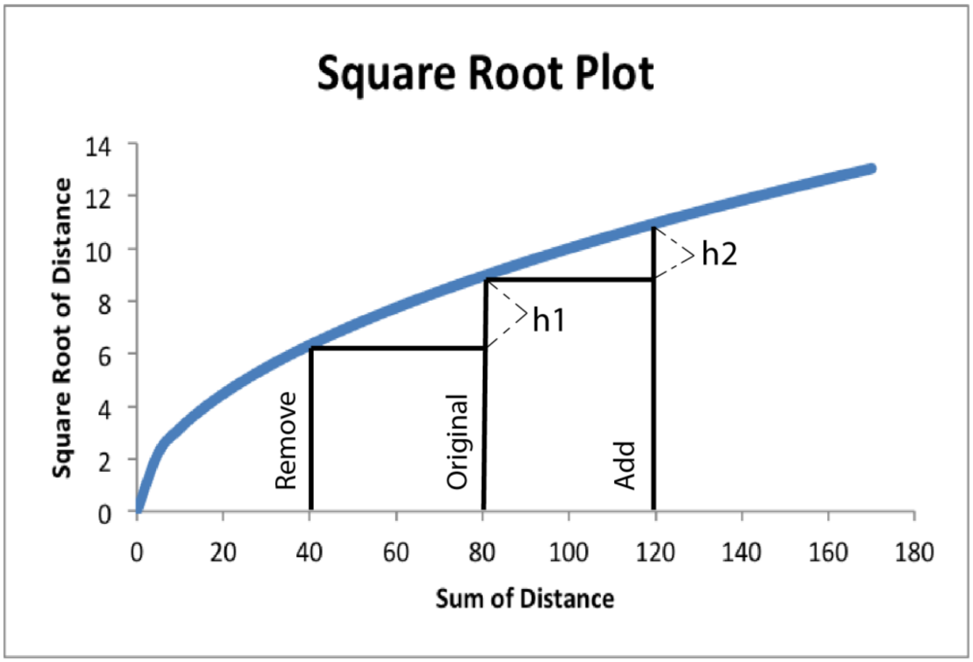


Figure 33: A demonstration on distance score changes.

To better search across large scale cases, we used Mean Shift clustering on the original 200 cases to group them into 4 clusters, with a goodness-of-fit coverage coefficient Rc=0.918766 (maximum 1). The clustering results are used for case retrieval when the new streaming data is available. Since re-ranking results requires shuffling current results based on the clustering, we can search the candidate retrieval cases from the same or neighbor clusters using the current top results. In this way, we can significantly reduce the search time, as shown in Table 17.

Table 17: searching time comparison between clustering results and total running time.

Running Time /s	Cluster 1	Cluster 2	Cluster 3	Total
2000000	2.283	1.729	4.427	13.36033333
20000000	18.435	13.329	39.845	60.92966667
200000000	113.791	61.371	325.817	536.622

6.3 Case Study

As described in Section 4, the CBR system bases the returned cases on the transition matrix that has been refined through the CBR cycle. The transition matrix defines the probability between two states (ROIs) that user will look at two states consecutively. While we have shown that users with different levels of expertise differ significantly in spatial and temporal search features, we cannot evaluate whether the probabilities defined by the transition matrix are significant in terms of semantic knowledge from statistical analysis alone. In order to determine the face validity of the returned results, we have shown the top frequency results by temporal matching and the associated image search task to radiology experts to determine whether the rationale behind the matching gaze sequence patterns was evident. While transitions between two or three ROIs exhibited higher frequencies, the investigators felt that the shorter sequences could more easily be attributed to chance. Therefore, the radiology experts focused on high-frequency sequences of four or more ROIs. The following descriptions detail the rationale for some of the identified sequences. Examples were selected to demonstrate the appropriateness of the frequency results across the ten levels of the visual knowledge pyramid described in section 2.2.

In the lowest level of the visual knowledge pyramid we asked participants, “What is the modality of this image?” The first case, shown in Figure 34, demonstrates two patterns identified in the temporal sequences for this task. The task involves low-level knowledge related to the type of visual information presented (either CT or MR). The matching gaze sequence in the CT image (Figure 34 left image) identified a pattern in which the radiologic technologists look from the spleen to the splenic artery, then to the stomach and on to the liver with 11.1% frequency. In this pattern, they are first comparing structures of differing radiographic density. This CT scan is performed with contrast media and performed during the arterial phase of contrast enhancement. Contrast media has a higher atomic number and will block more x-rays than the soft tissue structures, causing it to be displayed as a brighter structure. In MR, the motion of the blood results in an increased signal, so there is no contrast difference between the arteries and veins. Since the blood in the splenic artery is displayed as bright and the vessels in the liver are not, this is an indicator that the image was produced by computed tomography. This is then confirmed by comparing structures of similar subject density. In CT, structures of similar subject density will exhibit similar optical brightness due to similar attenuation of the x-ray beam. This is not necessarily the case in MR. In this image, the stomach and the liver are displayed with similar levels of brightness, confirming the idea that the image was produced by CT. Participants who viewed an MR image (Figure 34 right image) for this question exhibited similar comparing behavior. The gaze pattern in the MR image has 4.5% frequency. Participants first compared abdominal organs to the fat seen posterior to the spine. In CT, fat will always allow more transmission of x-rays through the area than the surrounding soft tissue, resulting in a darker image. In MR, fat is frequently brighter than the soft tissue, although the brightness is dependent upon the sequence used. They then compare the inferior vena cava to the surrounding abdominal structures. In CT, structures of similar subject density will exhibit similar optical brightness due to similar attenuation of the x-ray beam. In the case of the IVC, it will appear slightly darker than the surrounding liver tissue due to the slightly decreased subject density between the solid liver and the liquid blood. In MR, the movement of the blood typically results in increased signal and a bright image. In this way, they determine the modality of the image.

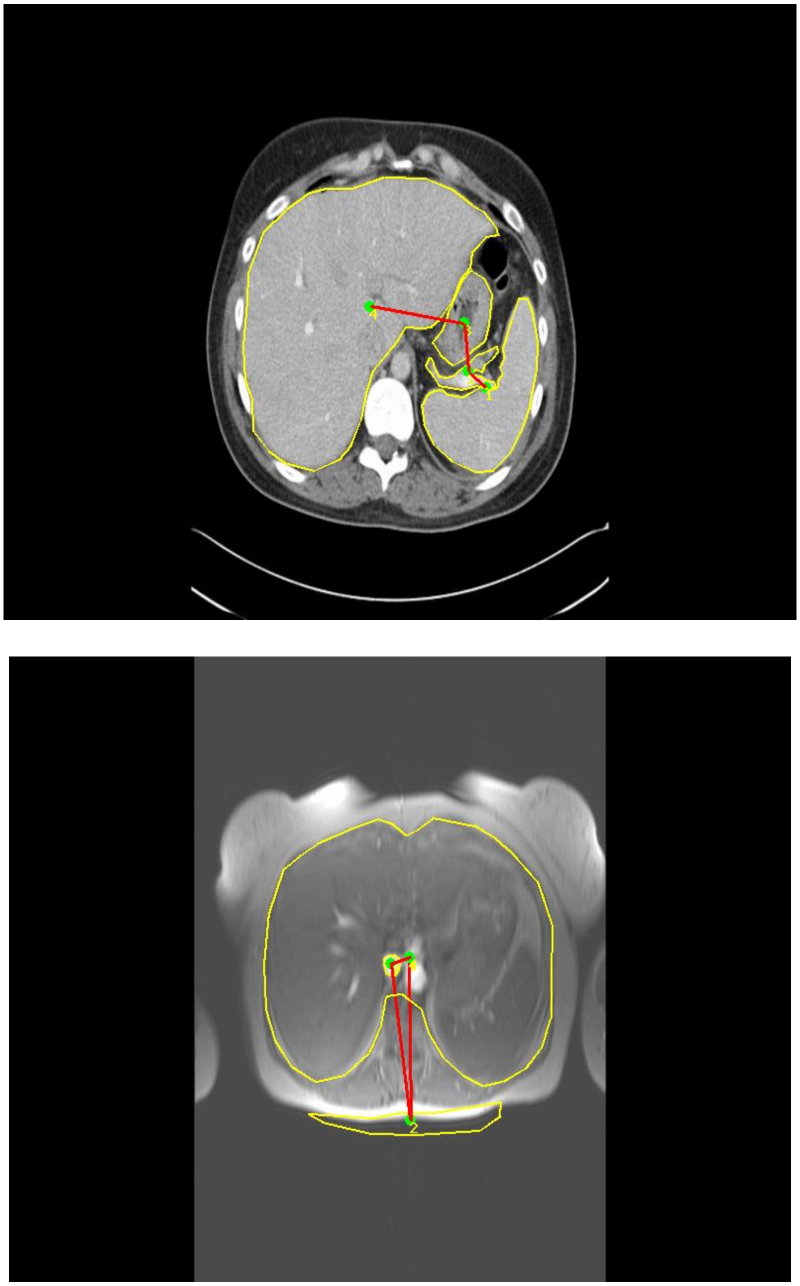


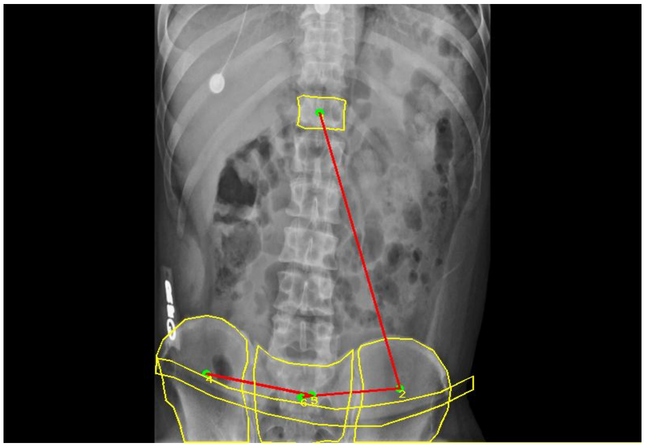
Figure 34: One example of case studies with highlight matching gaze pattern. CT image is at top. MR image is at bottom.

In comparison to the relatively simple level 1 task, we looked for the rationale in the temporal patterns for level 8, where we asked participants to evaluate the positioning demonstrated on the image. The pattern in Figure 35 left, which appears with 1.28% frequency within the data for this image, demonstrates an evaluation of the rotational aspects of the positioning. Rotation in an abdominal image may occur in the upper body, the lower body or both. The technologists begin the evaluation of rotation by looking at the relative position of the spinous process within the vertebral body, as a centered vertebral body in the upper lumbar spine indicates that the upper body is not rotated. They progress to the left ilium and trace to the right ilium along the line formed by the abdominal pannus, ending at the sacrum. This series of fixations constitutes a comparison of the symmetry formed by the ilium and an evaluation of centering of the sacrum between the iliac wings. When the pelvis is rotated the iliac wings are no longer symmetrical and equidistant from the center of the sacrum. The evaluation of pelvic rotation was also seen in another retrieval result (Figure 35 right), with a frequency of 1.28%.

We continued to see explainable patterns in the frequency results for level 9. In this level, we asked the participants to describe the patient based on the visual information provided. In Figure 36 (left), case three demonstrates a complex sequence, seen with 4.5% frequency, in which the technologists look at the right breast and a skin fold on the right side, followed by the right lung, left lung base, abdomen and heart. From this sequence, they can describe the patient as female, lacking muscle tone with clear lungs, decreased bone density, indicating possible osteoporosis, and normal heart size. The decreased bone density and lack of muscle tone indicate that the patient is likely older.

Finally, explainable patterns are seen in the highest level of visual knowledge as well. This question is in level 10 which requires them have more knowledge in order to answer. Participants were asked to determine the problems the patient in the image is exhibiting. In this case, shown in Figure 36 (right), the technologists look from the oval artifact caused by the backboard handle to the humeral shaft, then up to where the humeral shaft is impact fractured into the humeral head and over to the scapula. In this sequence of fixations, the radiologic technologist can determine that the patient has been involved in a traumatic event, like a fall or a motor vehicle accident, the patient has a fractured humerus of a type consistent with putting your arm forward to catch yourself, and the humeral head is not laterally displaced from the glenoid fossa of the scapula. This sequence of fixations occurred with 14.1%, which is much higher than the others.

Overall, the complex sequences with the highest frequencies demonstrated explainable patterns in relationship to the task undertaken. While the frequencies reported are relatively low, they do not account for instances of the same pattern with gaps or interruptions in the sequence, nor do they take into account instances where the sequence might occur in the reverse order. As the system is improved to account for gaps and reverse sequences, the frequencies of these identifiable patterns are likely to improve.



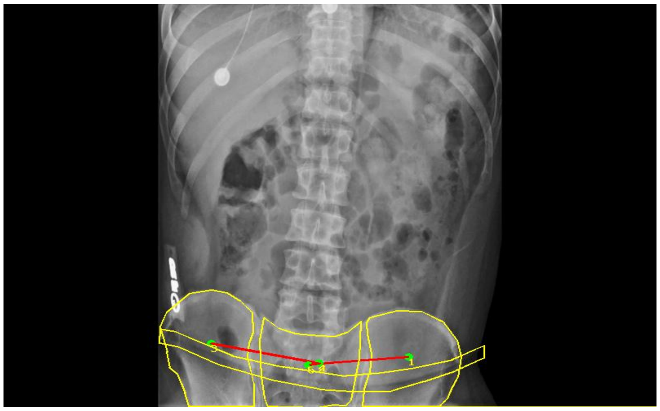


Figure 35: Second and third case study with highlight matching gaze pattern.

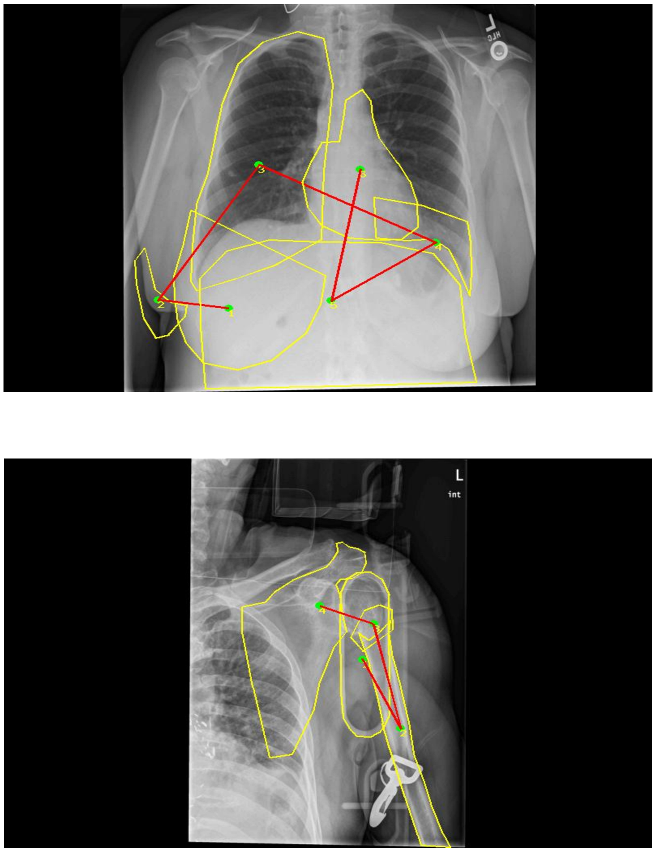


Figure 36: Fourth case study with highlight matching gaze pattern.

CHAPTER SEVEN

SUMMARY AND FUTURE WORK

7.1 Summary of Existing Work

This dissertation work proposes a new computational method to index and model the tacit and hard-won visual knowledge, and provides a framework for retrieving large scale cases in real time for computer science community. The visual activities are first addressed in the dissertation, especially for auto-segmentation and annotation. The proposed framework can be scaled to hundreds of millions of cases, as reported in the experimental results, using commodity hardware. Using real-time gaze tracking framework, we can computationally capture and index human visual knowledge in real time. This framework can be also used to assist image search such as satellite image and biomedical image retrieval.

This dissertation presents a case-based reasoning system for visual knowledge gaze tracking and image-based anatomical information. We have developed a new approach to represent an eye-tracking case using graph model and Markov Chain (MC) model to capture both spatial and temporal features. A graph distance function was used in a case-based reasoning (CBR) system to retrieve similar cases from the case library. By spatial and temporal pattern mining and comparisons, we find that experts (radiological practitioners and professionals) and novices (junior and senior-year students) tend to use different visual patterns to solve the same high-level tasks. For most low-level tasks, however, there are no significant differences between junior and senior students compared to experts' visual patterns. In the novice group, we observed that junior and senior students have no significant differences when dealing with high-level tasks. One of reasons could be that both subgroups lack of needed knowledge to answer the questions.

We tested our proposed framework on 200 human visual reasoning cases from 12 junior and 13 senior students from MU Radiography training program. An innovative approach was proposed and implemented to analyze human gaze patterns spatially (Sub-graph Matching) and temporally (Markov Decision process) in an attempt to model human cognitive process. To computationally mimic human reasoning, in addition to the analysis of low level gaze features (fixation, duration, saccades) in our previous work, we proposed three visual actions (*focus*, *read* and *scan*) for temporal representation of the human gaze. These three visual actions were then segmented and annotated using a linear-chain conditional random field model.

Finally, employing a big data ecosystem, we developed a novel indexing and retrieval framework for real-time case ranking over hundreds of millions of visual knowledge cases. The proposed framework was implemented on a Hadoop Distributed File System - HDFS, Spark (Streaming, SQL and GraphX) and HBase. To analyze the scalability of the proposed framework, we randomly generated up to 200 million simulation cases with controlled variation. The results show both the accuracy and the speed of our proposed Spark framework.

7.2 Potential Applications

The traditional text-based or content-based search cannot take users' visual reasoning into consideration when they search on those media. With the recent availability of inexpensive gaze trackers, more and more gaze tracking streaming data are expected to be collected in the near future. Our proposed framework is designed not only for similar gaze tracking searches, but also for the following applications: **1. Quality control** – during domain experts' image analytic processes, the proposed real-time gaze matching system can evaluate practitioners' gaze tracking performance and remind them if they miss a salient region in the image; **2. Training** – the proposed gaze tracking framework can also be used as training system for novices, who can learn from the domain experts' gaze pattern and summarize this tacit visual strategy; **3. Searching** – users can query similar gaze tracking visual activity from the case library in real time. This visual knowledge search can assist them in analyzing the image and help them interpret the domain image.

7.3 Future work

7.3.1 Development of Retrieval Algorithms for Other Domain

We expect to see multiple follow-up research endeavors and studies using the proposed framework. One potential future goal is to apply the existing computing framework to other image domains, such as geospatial intelligence, fine arts, etc. The challenge here would be how to define salient regions for each domain, and how to design questions capable of testing various levels of visual knowledge in a way that best allows us to encode human visual strategies. Moreover, some other domains, such as journalism and education may require multimedia data for study (video, flash animation, cartoons, etc.). Thus, for those domains, challenges arise from the dynamic nature of content from videos, animations, and zoom in/out operations.

7.3.2 Expanded Existing Visual Action Dictionary

Though the proposed computation model for visual knowledge was developed in this dissertation, research on adding more visual activities such as tracing or comparison based on the existing three basic actions {*Focus, Read, Scan*} could still be performed. First, the new computational approach could be utilized to model more complex visual strategies through the proposed searching framework. Second, for certain domains, we can design specific visual actions designed to most accurately and efficiently analyze domain images.

7.3.3 Large Scale Streaming and In Memory Computing and Indexing

Today we live in a big data world. It is not sufficient to store, index and search large scale images, text, video, audio or other type of media data. Using this proposed big data ecosystem environment, we can easily scale database or streaming data to the GB or even TB level. Thus, approaches allowing system better load and balance the data caching are still needed. We can expand our retrieval engine in multiple ways. First, we could design better streaming modules and scale the retrieval cases in real time. Second, a load balance algorithm can be used to preload candidate cases into memory for faster searching. Third, in situations with limited RAM space, we can expand the caching area to the hard disk using an off-heap caching Tachyon. Finally, in addition to proposed indexing method – Mean Shift clustering – we can extend the indexing structure to tree indexes in future, such as Metric Tree [103, 104] or kd-Tree [105]. Using the Spark GraphX component, it is possible to build and customize the indexing tree in a distributed computing framework. This new tree indexing structure can be preload into memory and distributed across cluster for large scale retrieval.

BIBLIOGRAPHY

[1] M. Carrasco, "Visual attention: The past 25 year," *Vision Research,* vol. 51, pp. 1484–1525, 2011.

[2] N. Kanwisher and E. Wojciulik, "Visual attention: Insights from brain imaging," *Nature Reviews Neuroscience* vol. 1, pp. 91-100, 2000.

[3] S.-K. Chang, "Visual reasoning for information retrieval from very large databases," *Journal of Visual Languages & Computing,* vol. 1, pp. 41–58, 1990.

[4] S.-K. Chang and Y. Deng, "Intelligent database retrieval by visual reasoning," *Proc. 1990 IEEE COMPSAC Conf.,* pp. 459-464, 1990.

[5] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology,* vol. 4, pp. 219-227, 1985.

[6] T. C. Kübler, E. Kasneci, and W. Rosenstiel, "Gaze guidance for the visually impaired," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 385-386.

[7] L. Dupont and V. Van Eetvelde, "The use of eye-tracking in landscape perception research," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 389-390.

[8] O. K. Oyekoya and F. W. M. Stentiford, "Exploring human eye behaviour using a model of visual attention," *Proceedings oof the 17th International Conference on Pattern Recognition (ICPR'04),* pp. 945-948, 2004.

[9] A. Egawa and S. Shirayama, "A method to construct an importance map of an image using the saliency map model and eye movement analysis," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, Santa Barbara, California, 2012, pp. 21-28.

[10] H. Igarashi, S. Suzuki, T. Sugita, M. Kurisu, and M. Kakikura, "Extraction of visual attention with gaze duration and saliency map," in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, 2006, pp. 562-567.

[11] Z. Liang, H. Fu, Y. Zhang, Z. Chi, and D. Feng, "Content-based image retrieval using a combination of visual features and eye tracking data," *ACM Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications,* 2010.

[12] A. T. Duchowski, J. Driver, S. Jolaoso, W. Tan, B. N. Ramey, and A. Robbins, "Scanpath comparison revisited," *ACM Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications,* p. 8, 2010.

[13] T. C. Kübler, E. Kasneci, and W. Rosenstiel, "Subsmatch: Scanpath similarity in dynamic scenes based on subsequence frequencies," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 319-322.

[14] N. Chrobot, "The role of processing fluency in online consumer behavior: evaluating fluency by tracking eye movements," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 387-388.

[15] S. P. Marshall, "The Index of Cognitive Activity: measuring cognitive workload," in *Human Factors and Power Plants, 2002. Proceedings of the 2002 IEEE 7th Conference on*, 2002, pp. 7-5-7-9.

[16] S. Marshall, "Method and apparatus for eye tracking and monitoring pupil dilation to evaluate cognitive activity," *U.S. Patent 6090051,* 2000.

[17] S. P. Marshall, C. W. Pleydell-Pearce, and B. T. Dickson, "Integrating psychophysiological measures of cognitive workload and eye movements to detect strategy shifts," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, 2003, p. 6 pp.

[18] D. D. Salvucci and J. R. Anderson, "Automated eye-movement protocol analysis," *Human-computer interaction,* vol. 16, pp. 39-86, 2001.

[19] S. L. Groll and J. Hirsch, "Two-dot vernier discrimination within 2.0 degrees of the foveal center," *Journal of Optical Society of America,* vol. 4, p. 8, 1987.

[20] T. Blascheck and T. Ertl, "Towards visualizing eye movement data from interactive stimuli," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 391-392.

[21] K. Holmqvist, M. Nystrom, and F. Mulvey, "Eye tracker data quality: what it is and how to measure it," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, Santa Barbara, California, 2012, pp. 45-52.

[22] D. Akkil, P. Isokoski, J. Kangas, J. Rantala, and R. Raisamo, "TraQuMe: a tool for measuring the gaze tracking quality," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 327-330.

[23] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System," *AI Communication,* vol. 7, pp. 39-59, 1994.

[24] D. W. Aha, L. A. Breslow, and H. Munoz-Avila, "Conversational Case-Based Reasoning," *Applied Intelligence,* pp. 1-25, 1999.

[25] K.-D. Althoff, "Case-Based Reasoning," *Handbook on Software Engineering and Knowledge Engineering,* 2001.

[26] D. B. Leake, "CBR in context The present and future," *Case-Based Reasoning: Experience, Lessons & Future Directions, AAAI Press/ The MIT Press,* pp. 3-30.

[27] R. L. D. Mantaras, D. Mcsherry, D. Bridge, D. Leake, B. Smyth, S. Craw*, et al.*, "Retrieval, reuse, revision and retention in case-based reasoning," *The Knowledge Engineering Review,* vol. 20, pp. 215-240, 2006.

[28] P. Cunningham, "A Taxonomy of Similarity Mechanisms for Case-Based Reasoning," *IEEE Transaction on knowledge and data engineering,* vol. 21, pp. 1532-1543, 2009.

[29] R. Bergmann and K.-D. Althoff, "Methodology for Building CBR Applications," pp. 299-326, 1998.

[30] R. Bergmann and W. Wilke, "On the Role of Abstraction in Case-Based Reasoning," pp. 28-43, 1996.

[31] R. Shank, "Dynamic memory; a theory of reminding and learning in computers and people," *New York: Cambridge University Press,* 1982.

[32] A. Aamodt, "A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning," *PhD thesis, University of Trondheim,* 1991.

[33] P. Perner, "Case-Based Reasoning and the Statistical Challenges," *Journal Quality and Reliability Engineering,* vol. 24, pp. 705-720, 2008.

[34] M. M. Richter, "Knowledge Containers," *Readings in Case-Based Reasoning,* 2005.

[35] W. He, S. Erdelez, F.-K. Wang, and C.-R. Shyu, "The effects of conceptual description and search practice on users' mental models and information seeking in a case-based reasoning retrieval system," *Information Processing and Management,* vol. 44, pp. 294-309, 2008.

[36] B. Smyth, M. T. Keane, and P. Cunningham, "Hierarchical Case-Based Reasoning Integrating Case-Based and Decompositional Problem-Solving techniques for Plant-Control Software design," *IEEE Transaction on knowledge and data engineering,* vol. 13, 2001.

[37] P. Perner, "An architecture for a CBR image segmentation system," *Engineering Applications of Artificial Intelligence,* vol. 12, p. 11, 1999.

[38] T. White, *Hadoop: The definitive guide*: " O'Reilly Media, Inc.", 2012.

[39] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, 2010, pp. 1-10.

[40] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster Computing with Working Sets," *HotCloud,* 2010.

[41] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed GraphLab: a framework for machine learning and data mining in the cloud," *Proceedings of the VLDB Endowment,* vol. 5, pp. 716-727, 2012.

[42] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, "Graphx: A resilient distributed graph system on spark," in *First International Workshop on Graph Data Management Experiences and Systems*, 2013, p. 2.

[43] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters," in *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing*, 2012, pp. 10-10.

[44] C.-Y. Lin, C.-H. Tsai, C.-P. Lee, and C.-J. Lin, "Large-scale logistic regression and linear support vector machines using Spark," in *Big Data (Big Data), 2014 IEEE International Conference on*, 2014, pp. 519-528.

[45] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows*, et al.*, "Bigtable: A distributed storage system for structured data," *Seventh Symposium on Operating System Design and Implementation,* 2006.

[46] A. Khetrapal and V. Ganesh, "HBase and Hypertable for large scale distributed storage systems," *Dept. of Computer Science, Purdue University,* pp. 22-28, 2006.

[47] J. Sun and Q. Jin, "Scalable rdf store based on hbase and mapreduce," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, 2010, pp. V1-633-V1-636.

[48] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony*, et al.*, "Hive: a warehousing solution over a map-reduce framework," *Proceedings of the VLDB Endowment,* vol. 2, pp. 1626-1629, 2009.

[49] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM - 50th anniversary issue: 1958 - 2008,* vol. 51, pp. 107-113 2008.

[50] J. Dean and S. Ghemawat, "MapReduce: a flexible data processing tool," *Communications of the ACM,* vol. 53, pp. 72-77, 2010.

[51] C. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng*, et al.*, "Map-reduce for machine learning on multicore," *Advances in neural information processing systems,* vol. 19, p. 281, 2007.

[52] D. Borthakur, "The hadoop distributed file system: Architecture and design," *Hadoop Project Website,* vol. 11, p. 21, 2007.

[53] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans*, et al.*, "Apache hadoop yarn: Yet another resource negotiator," in *Proceedings of the 4th annual Symposium on Cloud Computing*, 2013, p. 5.

[54] S. S. Hacisalihzade, L. W. Stark, and J. S. Allen, "Visual perception and sequences of eye movement fixations: a stochastic modeling approach," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 22, pp. 474-481, 1992.

[55] K. Preston White, T. L. Hutson, and T. E. Hutchinson, "Modeling human eye behavior during mammographic scanning: preliminary results," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 27, pp. 494-505, 1997.

[56] S.-C. Zhu, "Statistical Modeling and Conceptualization of Visual Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, p. 22, 2003.

[57] K. Barnard and D. Forsyth, "Learning the semantics of words and pictures," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2001, pp. 408-415 vol.2.

[58] H. Eidenberger and C. Breiteneder, "Semantic feature layers in content-based image retrieval: implementation of human world features," in *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, 2002, pp. 174-179 vol.1.

[59] L. G. Shapiro, I. Atmosukarto, H. Cho, H. J. Lin, and S. Ruiz-Correa, "Similarity-based retrieval for biomedical applications," *Case-based Reasoning on Signals and Images, Perner P (ed.). Springer,* p. 34, 2007.

[60] F. Shic, K. Chawarska, J. Bradshaw, and B. Scassellati, "Autism, eye-tracking, entropy," in *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, 2008, pp. 73-78.

[61] F. Shic, B. Scassellati, D. Lin, and K. Chawarska, "Measuring context: The gaze patterns of children with autism evaluated from the bottom-up," in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, 2007, pp. 70-75.

[62] E. A. Krupinski, A. A. Tillack, L. Richter, J. Henderson, A. K. Bhattacharyya, K. M. Scott*, et al.*, "Eye-movement study and human performance using telepathology virtual slides. Implications for medical education and differences with experience," *Human Pathology,* vol. 37, p. 14, 2006.

[63] F. Galgani, S. Yiwen, P. L. Lanzi, and J. Leigh, "Automatic analysis of eye tracking data for medical diagnosis," in *Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on*, 2009, pp. 195-202.

[64] M. S. Kim, A. Burgess, A. J. Waters, G. P. Reece, E. K. Beahm, M. A. Crosby*, et al.*, "A Pilot Study on Using Eye Tracking to Understand Assessment of Surgical Outcomes from Clinical Photography," *Journal of Digital Imaging,* vol. 24, pp. 778-786, 2011.

[65] S.-K. Chang and S.-H. Liu, "Picture indexing and abstraction techniques for pictorial databases," *IEEE Transactions on Pattern analysis and machine intelligence,* vol. 4, pp. 475-484, 1984.

[66] S.-K. Chang and Y. Deng, "Intelligent database retrieval by visual reasoning," in *Computer Software and Applications Conference Proceedings., Fourteenth Annual International. IEEE*, 1990.

[67] S.-K. Chang, "Visual reasoning for information retrieval from very large databases," *Journal of Visual Languages & Computing* vol. 1, pp. 41-58, 1990.

[68] A. Jaimes and S.-F. Chang, "A Conceptual Framework for Indexing Visual Information at Multiple Levels," *IN PROCEEDINGS OF SPIE INTERNET IMAGING,* vol. 3964, pp. 2-15, 2000.

[69] X. Wang, S. Erdelez, C. Allen, B. Anderson, H. Cao, and C.-R. Shyu, "The Role of Domain Knowledge in Developing User-Centered Medical Image Indexing," *Journal of the American Society for Information Science and Technology (JASIST),* vol. 63, pp. 225-241, 2012.

[70] I. Watson, *Applying case-based reasoning: techniques for enterprise systems*: Morgan Kaufmann Publishers Inc., 1998.

[71] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D. W. Hansen*, et al.*, "Evaluation of a low-cost open-source gaze tracker," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, 2010, pp. 77-80.

[72] M. Barrett, H. Skovsgaard, and J. San Agustin, "Performance evaluation of a Low-Cost gaze tracker for eye typing," in *Proc. Conf. Commun. Gaze Interact. Lyngby, Denmark: COGAIN*, 2009, pp. 13-17.

[73] Y. Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel, "SAGA: a subgraph matching tool for biological graphs," *Bioinformatics,* vol. 23, p. 8, 2007.

[74] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*, 2001.

[75] M. L. Puterman, "Markov decision processes: discrete stochastic dynamic programming," vol. 414, 2009.

[76] P. Blunsom, "Hidden markov models," *Lecture notes,* vol. 15, pp. 18-19, 2004.

[77] L. R. Rabiner and B.-H. Juang, "An introduction to hidden Markov models," *ASSP Magazine, IEEE,* vol. 3, pp. 4-16, 1986.

[78] M. K. Cowles and B. P. Carlin, "Markov chain Monte Carlo convergence diagnostics: a comparative review," *Journal of the American Statistical Association,* vol. 91, pp. 883-904, 1996.

[79] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," in *International Conference on Machine Learning (ICML)*, 2000, pp. 591-598.

[80] T. Hara, D. Mochihashi, Y. Kano, and A. Aizawa, "Predicting word fixations in text with a CRF model for capturing general reading strategies among readers," in *Proceedings of the First Workshop on Eye-Tracking and Natural Language Processing*, 2012.

[81] T. M. T. Do and T. Artières, "Conditional Random Field for tracking user behavior based on his eye's movements," in *Workshop at NIPS 2005*, Whistler, BC, Canada, 2005.

[82] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, "Discretized streams: Fault-tolerant streaming computation at scale," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, 2013, pp. 423-438.

[83] H.-c. Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker, "Map-reduce-merge: simplified relational data processing on large clusters," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007, pp. 1029-1040.

[84] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley*, et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, 2012, pp. 2-2.

[85] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, ed, 2000, pp. 71–78

[86] P. Dobruschin, "The description of a random field by means of conditional probabilities and conditions of its regularity," *Theory of Probability & Its Applications,* vol. 13, pp. 197-224, 1968.

[87] *Solr - near real time indexing search engine*. Available: http://lucene.apache.org/solr/

[88] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, 2000, pp. 39-48.

[89] I. Greenberg Ronald, "Bounds on the Number of Longest Common Subsequences'," *The Computing Research Repository cs. DM/0301030,* pp. 1-13, 2003.

[90] D. D. Salvucci, "Mapping Eye Movements to Cognitive Processes," *Doctoral Dissertation, Department of Computer Science, Carnegie Mellon University,* 1999.

[91] P. Majaranta, U.-K. Ahola, and O. Špakov, "Fast gaze typing with an adjustable dwell time," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 357-360.

[92] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale CRFs," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 504-513.

[93] T. Michael and I. Jordan, "Reinforcement learning algorithm for partially observable Markov decision problems," *Proceedings of the Advances in Neural Information Processing Systems,* pp. 345-352, 1995.

[94] A. Yarbus, *Eye Movements and Vision*: Springer, 1967.

[95] T. Betz, T. C. Kietzmann, N. Wilming, and P. König, "Investigating task-dependent top-down effects on overt visual attention," *Journal of Vision,* vol. 10, 2010.

[96] M. M. Hayhoe, A. Shrivastava, R. Mruczek, and J. B. Pelz, "Visual memory and motor planning in a natural task," *Journal of Vision,* vol. 3, 2003.

[97] J. D. Nelson, G. W. Cottrell, J. R. Movellan, and M. I. Sereno, "Yarbus lives: a foveated exploration of how task influences saccadic eye movement," *Journal of Vision,* vol. 4, 2004.

[98] K. M. Martensen, *Radiographic image analysis*: Saunders Elsevier, 2011.

[99] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. v. d. Weijer, *Eye tracking: a comprehensive guide to methods and measures*, 2011.

[100] K. A. Ericsson, N. Charness, R. R. Hoffman, and P. J. Feltovich, *The Cambridge handbook of expertise and expert performance*: Cambridge University Press, 2006.

[101] P. Zieliński, " Opengazer: open-source gaze tracker for ordinary webcams (software)," *Samsung and The Gatsby Charitable Foundation.,* 2008.

[102] H. Mann and D. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *Annals of Mathematical Statistics,* vol. 18, pp. 50-60, 1947.

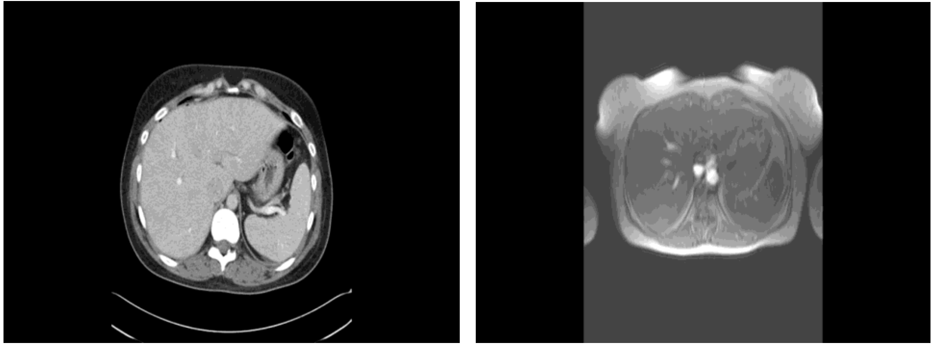
[103] J. K. Uhlmann, "Satisfying general proximity/similarity queries with metric trees," *Information processing letters,* vol. 40, pp. 175-179, 1991.

[104] S. Gibson and K. Gibson, "MTree data structure for storage, indexing and retrieval of information," ed: Google Patents, 1996.

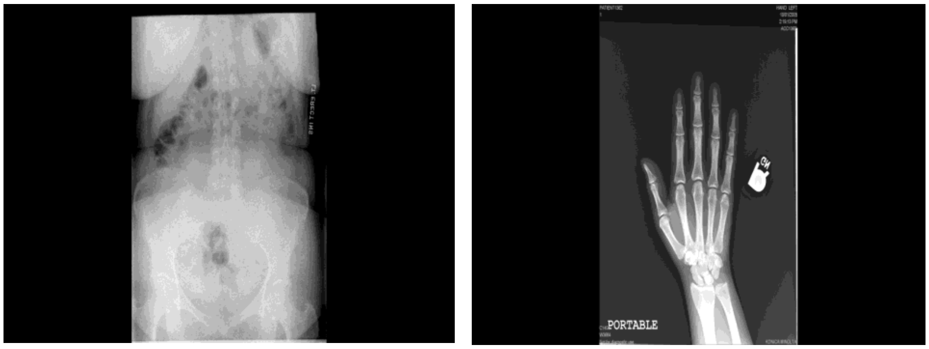
[105] G. J. Scott and C.-R. Shyu, "EBS kd tree: An entropy balanced statistical kd tree for image databases with ground-truth labels," in *Image and Video Retrieval*, ed: Springer, 2003, pp. 467-477.

Appendix A

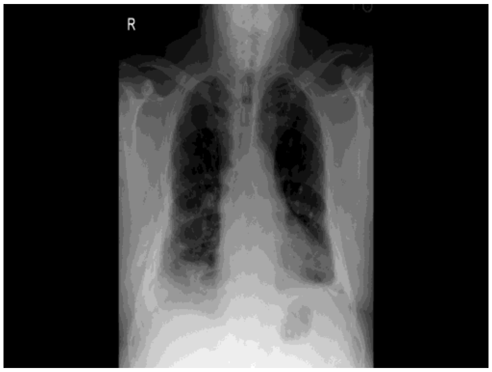
A.1 Ten Level Images



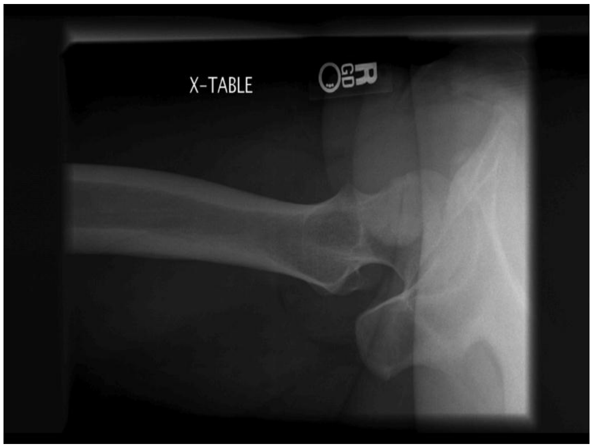
(a) Level 1 - What is the modality of this image?



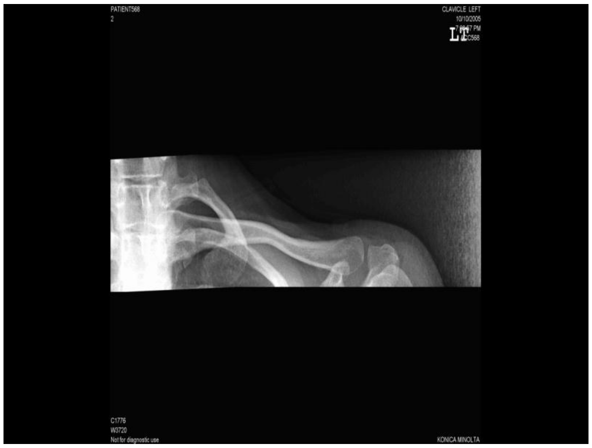
(b) Level 2 - Describe the overall photographic properties of this image.



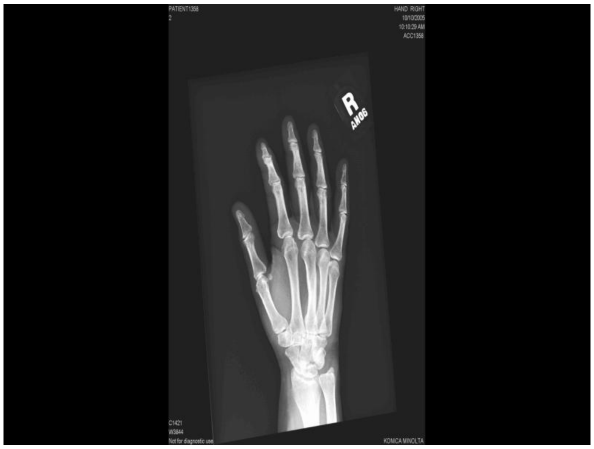
(c) Level 3 - What basic textual elements do you identify on this image?



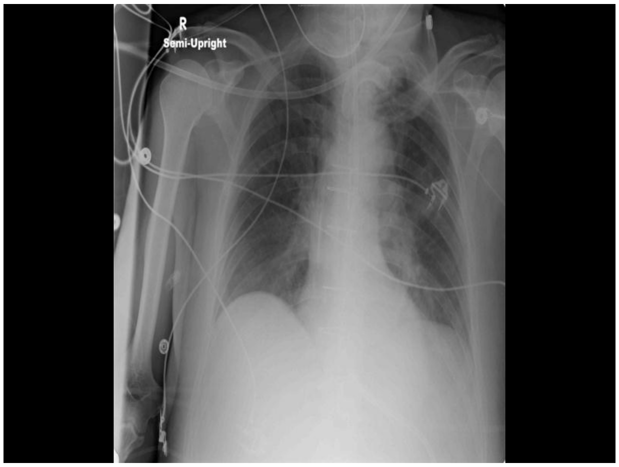
(d) Level 4 - How do you orient yourself to this image?



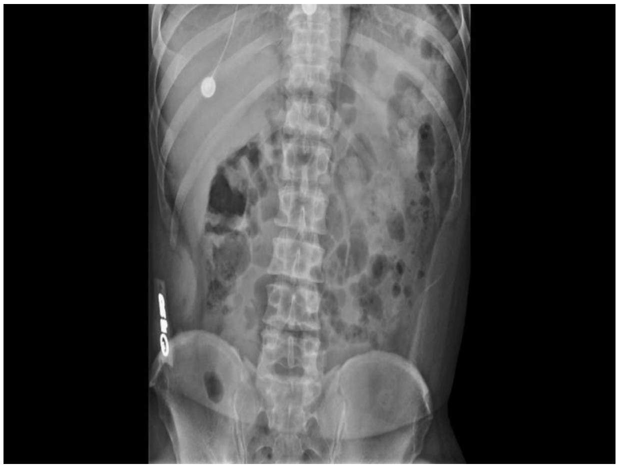
(e) Level 5 - What body part does this image demonstrate?



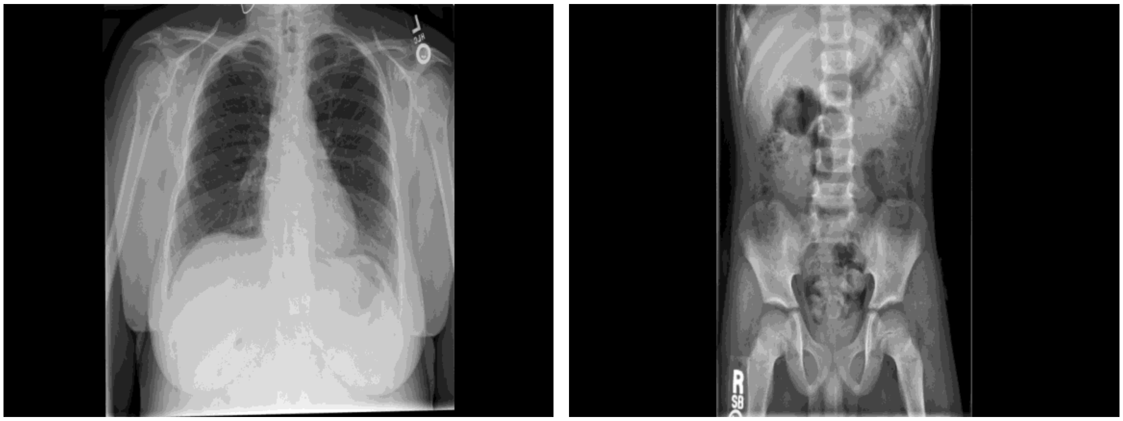
(f) Level 6 - What is the projection of this image?



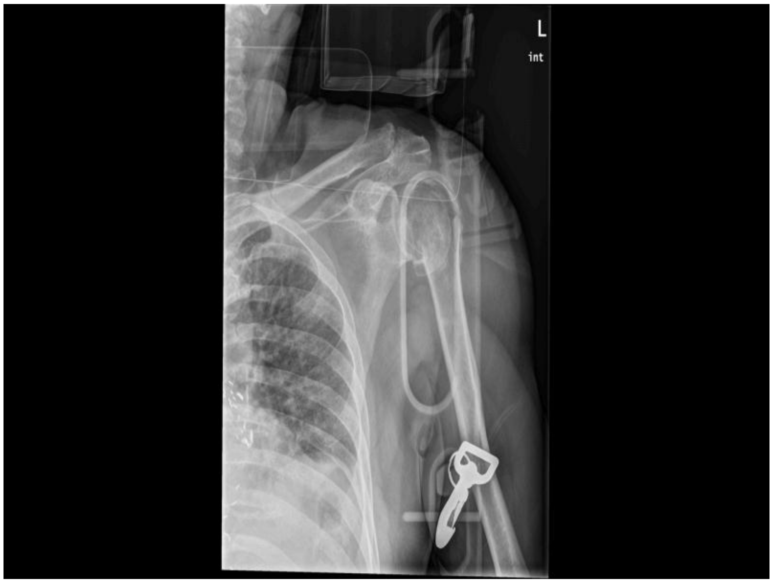
(g) Level 7 - Identify 3 foreign objects on this image.



(h) Level 8 - Evaluate the positioning of this image.

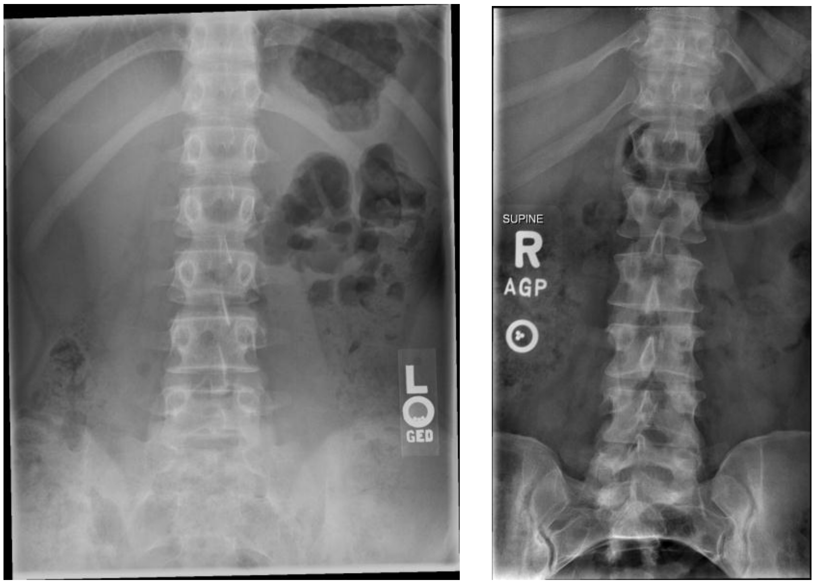


(i) Level 9 - Describe this patient based on what you see in this image.

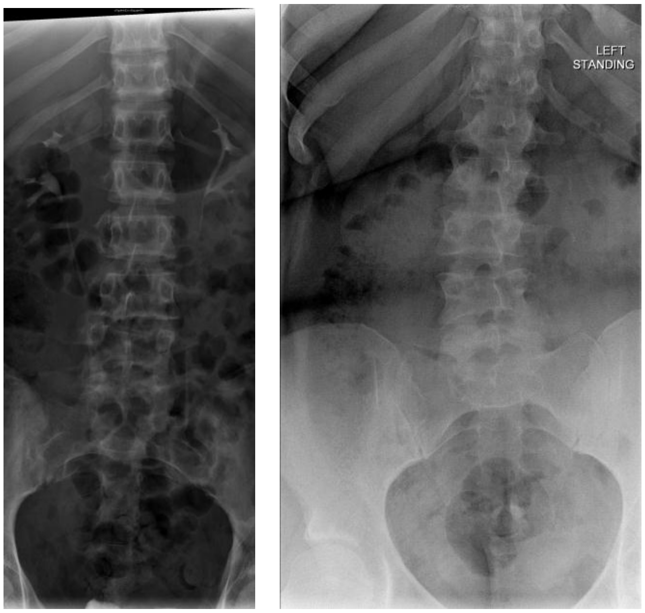


(j) Level 10 - What problem(s) do you think this patient has?

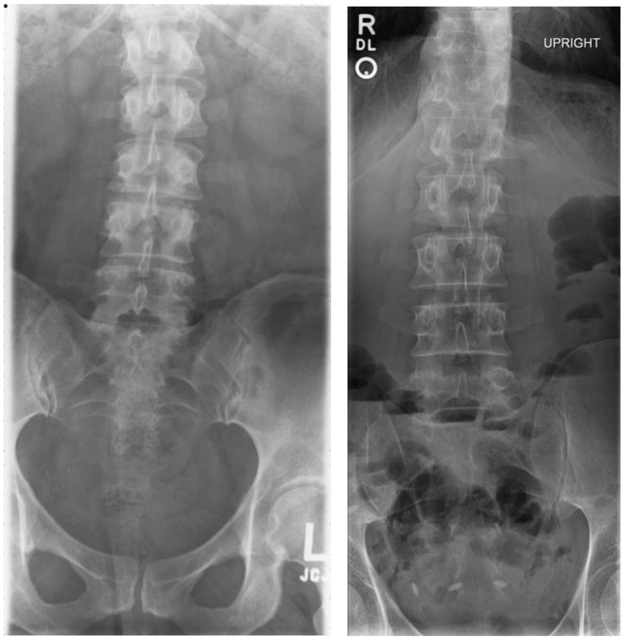
A.2 Three Level Images



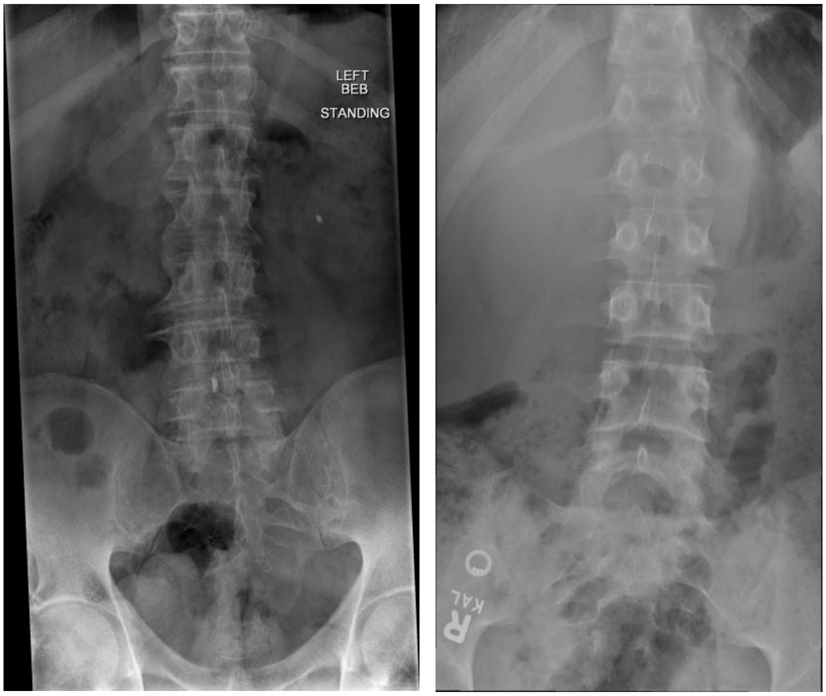
(a) Please evaluate the positive and negative aspects of this image.



(b) Level 2: Was the amount of x-rays used for this image adequate?

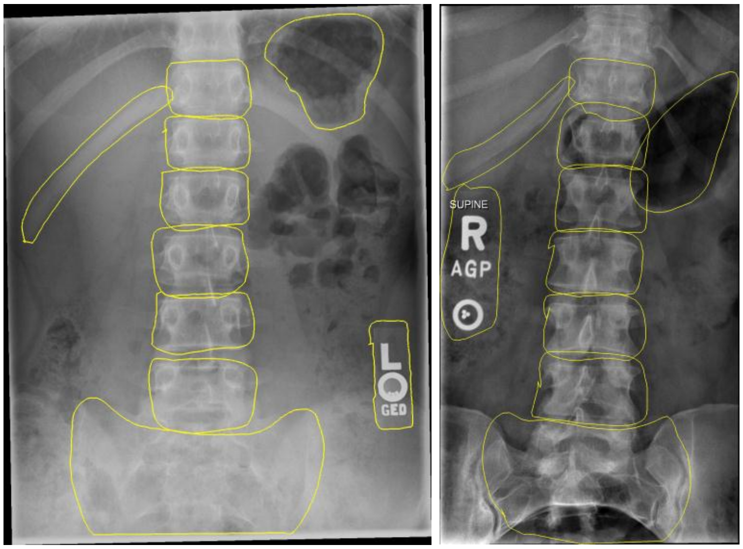


(c) Level 5: Does the image demonstrate all of the required anatomy?

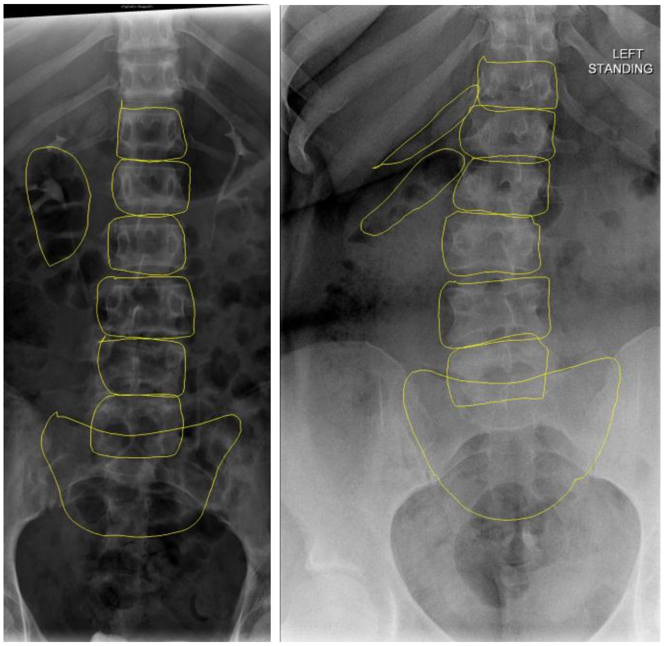


(d) Level 8: Are the relationships between the anatomical structures accurate?

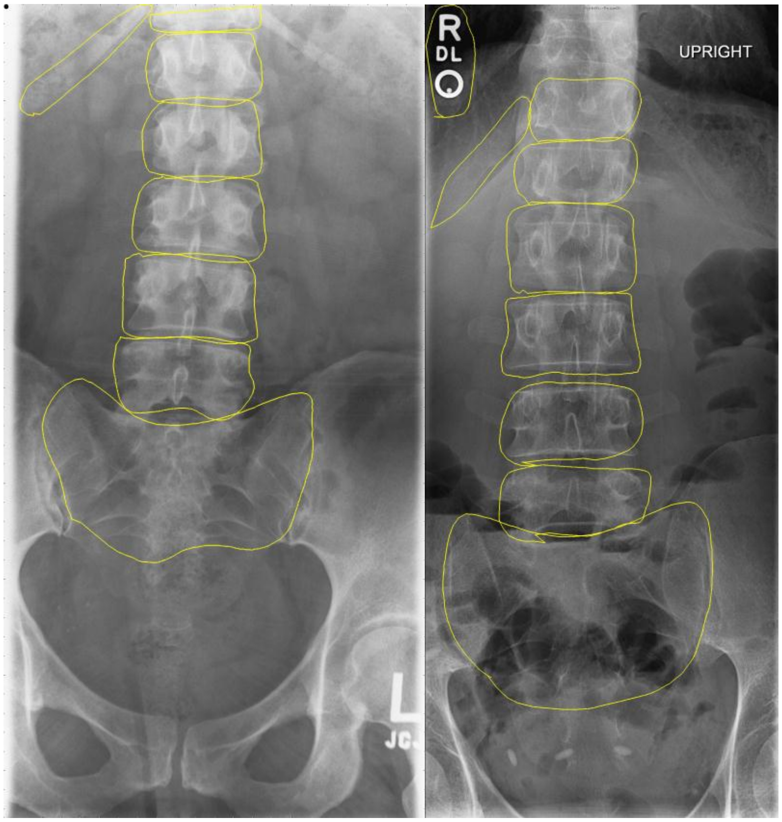
A.3 Domain Anatomy Marker Images



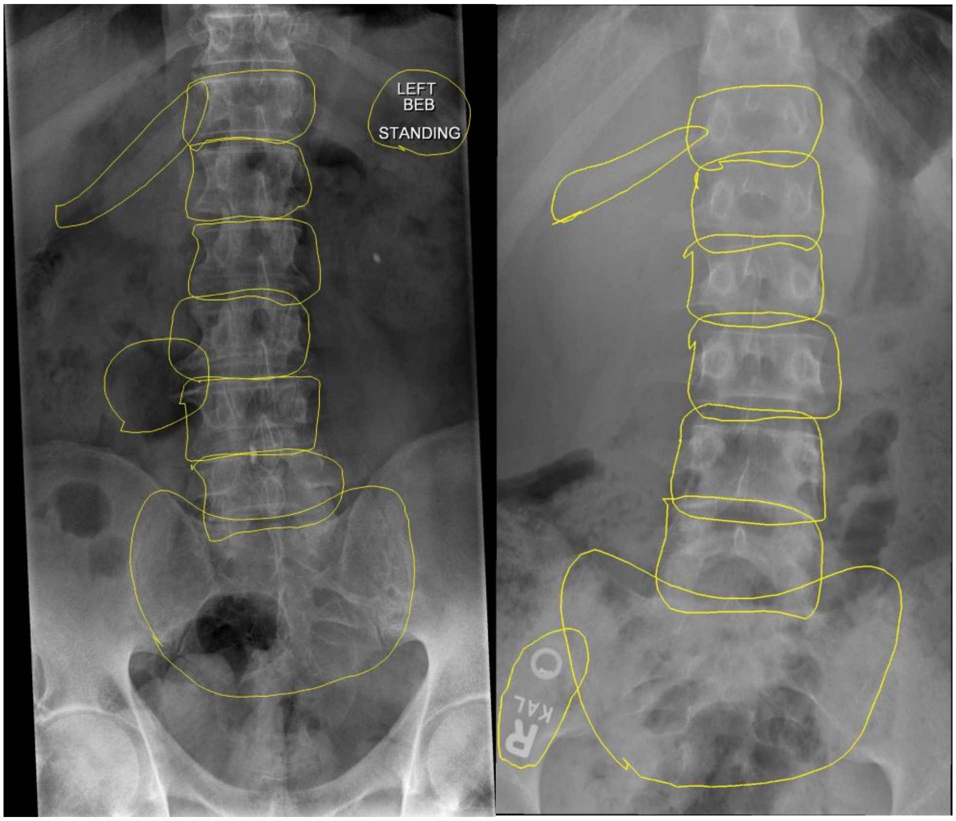
(a) Domain anatomies on images for overall question: Please evaluate the positive and negative aspects of this image.



(b) Domain anatomies on images for Level 2: Was the amount of x-rays used for this image adequate?



(c) Domain anatomies on images for Level 5: Does the image demonstrate all of the required anatomy?



(d) Domain anatomies on images for Level 8: Are the relationships between the anatomical structures accurate?

VITA

Hongfei Cao received his Ph.D. degree in computer science from the University of Missouri in 2015. He obtained his B.E. degree in computer science, University of Science & Technology of China.

Since 2008, he has worked as a Graduate Research Assistant in the Inter-discipline Data Analytics and Search Center under the direction of Dr. Chi-Ren Shyu at the University of Missouri. During this time, he received 2015 Outstanding Graduate Student in the Computer Science Department as part of the annual Missouri Honor Awards. And in 2014, he was selected as top 12 finalists from world-wide big data competition TEXATA 2014 over more than 2000 participants. In 2013, his Big Data project won IBM Smarter Planet Big Data Student Project Award. He also serves as a Teaching Assistant for senior level Relational Database Management Courses for the Computer Science Department.

During his graduate studies in the University of Missouri, Hongfei conducted research on several collaborative projects that included whole genomic sequence analysis in bioinformatics, human visual reasoning analysis with medical experts, database indexing and retrieval for Bio-Image. In his dissertation, he analyzed high-throughput visual knowledge in Big Data distributed paradigm. These works resulted in several publications, conference posters, and presentations.