AN INFORMATION RETRIEVAL SYSTEM FOR IMAGES FROM THE TRACE SATELLITE

by

Robert Ray Lamb

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

April, 2008

APPROVAL

of a thesis submitted by

Robert Ray Lamb

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Division of Graduate Education.

Dr. Rafal A. Angryk

Approved for the Department of Computer Science

Dr. John Paxton

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfullment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Robert Ray Lamb

April, 2008

ACKNOWLEDGEMENTS

I would like to thank Dr. Rafal Angryk for bringing me onto this project, having the patience to deal with my rough drafts, and giving copious amounts of invaluable advise. I would like to thank Monika Akbar, Shahriar Houssain, and Richard McAllister. We had lots of fun in our office, as we completed our projects. I would also like to thank Sam Gardner and Chad Bohannan for helping me to unwind during summer BBQs. Kathy Hollenback and Jeannette Radcliffe for keeping a constant supply of coffee in the office. I would also like to thank my parents and brother for their support. Finally, I would like to thank Kristin Harbuck for her patience and support though this process.

TABLE OF CONTENTS

1. INTRODUCTION 1

Motivation 2
Scope 2
Image Classification 3
Information Retrieval 6
Phenomena Catalogs 6
Query Interface 6
Background 7
Similar Systems 9
Image Format 12
The TRACE Satellite 12
Classifiers, Evaluations & Measures 13

2. IMAGE PROCESSING 17

Data Set 17
Preprocessing the Images 18
Segmenting the Images 19
Feature Extraction 21
Supporting Image Libraries 23
Sampling 23
Benchmark Creation 24

3. TRAINING CLASSIFIERS 25

Evaluation of Extracted Features 26
Training of Classifiers 27
Conclusions 31

4. INFORMATION RETRIEVAL 33

System Components 33
Evaluation of our Sample-based IR 35

5. CONCLUSIONS AND FUTURE WORK 40

REFERENCES 43

APPENDIX A Sample Query Images 48

LIST OF TABLES

Table Page

1 Minimum and Maximum Values generated by each extracted feature. . 22
2 Average values for detecting the Empty-Sun. 28
3 Average values for detecting the Coronal Loops. 29
4 Average AUC for ROS and RUS. 29
5 Average F-Measure for RUS and ROS. 29
6 Minimum and Maximum AUC values using the RUS technique. 30
7 Minimum and Maximum AUC values using the ROS technique. 30

LIST OF FIGURES

Figure Page

1 Data flow through the image classifying and searching mechanism. 3

2 A sample arrangement of binary classifiers to produce a set of labels. . . 5

3 SnapFind displaying results from interactive search. 11

4 A sample ROC curve 15

5 Examples showing how we are currently segmenting the solar images . 20

6 Average Recall vs. Precision for all of our sample images. 36

7 Results from the first sample query image containing a coronal loop. . . 37

8 Results from the second query image containing the best results. 37

9 Results from the third query image. 38

10 Results the fourth sample query image containing a sun spot. 38

11 The first sample query image. 49

12 The second sample query image. 50

13 The third sample query image. 51

14 The fourth sample query image. 52

ABSTRACT

The ability to identify particular features and structures, such as faces or types of scenery in images, is a topic with many available applications and potential solutions. In this paper we discuss solar images and the results of our preliminary investigation of techniques that can be used to identify solar phenomena in images from the TRACE satellite. Being able to automatically identify various phenomena in solar images is of great interest for scientists studying phenomena on the sun. A set of characteristics that can be quickly extracted from solar images needs to be acquired. These characteristics are used to create classifiers for various phenomena contained in solar images. There are many obstacles that need to be overcome when extracting features and creating these classifiers. These include the inherent unbalanced data sets due to varying rates at which different phenomena appear and multiple phenomenon that could appear in each image. The classifiers that have been generated were used in the creation of an information retrieval system to make finding phenomenon solar images quick and easy.

1

INTRODUCTION

This paper discusses an investigation into creating classifiers for use in an Information Retrieval (IR) system for solar images. To be useful in practice, our classifiers need to be fast and accurate. Currently we are working with images from the Transition Region and Coronal Explorer (TRACE) satellite. This satellite takes images of the sun at regular intervals. The image repository size for TRACE is approximately one terabyte and is growing at the rate of 260 megabytes a day. Future satellites will be capable of taking higher resolution images at a faster rate than the current satellite. We are developing this system with the intent that it will be able to scale with the increased data rate of images in the future.

There are several difficulties that need to be understood with our classification problem. First, for each image there are multiple classes to choose from as opposed to binary classification where only two labels occur. The second issue is that each image may also by labeled multiple times with several different classes, because multiple phenomena can occur in the same image. Also, as with any real world problems, the classes are significantly imbalanced, with some classes appearing more than others.

The paper is broken into four sections. The introduction section will introduce the topic and provide motivation for this research. Similar systems that currently exist along with related research will be discussed. The second section will discuss the images, the file format they are in, and how they are processed to extract features. The third section discusses the creation of classifiers.

Motivation

As noted in the introduction, the current repository of images from TRACE is growing at a rapid pace. With a new image being taken at a rate of nearly one every three minutes, it becomes nearly impossible for humans to analyze every image. This problem will only become worse as new equipment becomes available that can gather data at a faster rate.

To help alleviate this problem an automated computer process will be needed to help analyze the incoming information. This paper discusses our investigation into creating such an Information Retrieval system for images of the sun. The goals of this system are to quickly classify features in new images as they come in and provide a basic search by sample interface that can be used to search a database of solar images for images with similar attributes.

Scope

The scope of this thesis covers three main areas that are separated into chapters. The first chapter covers processing the solar images, extracting attributes from the images, and storing these attributes for later use. The second chapter discusses the creation of classifiers based on the extracted attributes. These classifiers are used for identifying phenomena contained within the images. In the third chapter we discuss the prototype Information Retrieval system created using the classifiers trained in the second chapter. In the final chapter conclusions and ideas for future work are presented.

An overview of our proposed system is presented in Figure 1. The major components of this system are a Query Interface, an Information Retrieval mechanism,

and organization of Virtual Solar Observatory's (VSO) data into catalogs of labeled images.

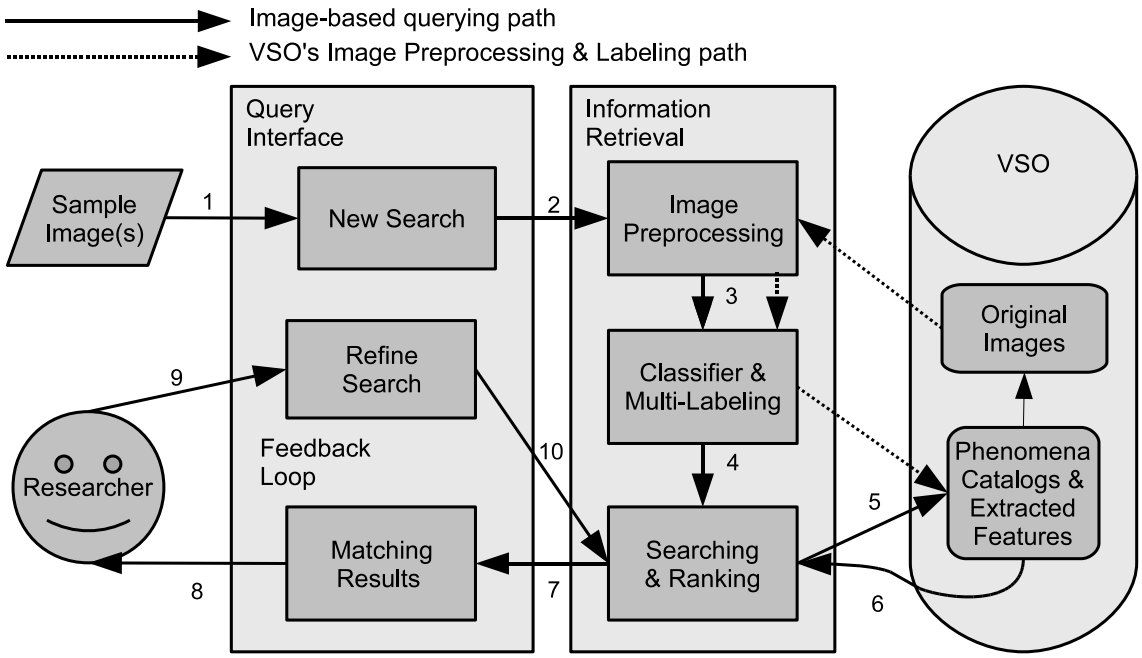


Figure 1: Data flow through the image classifying and searching mechanism.

Image Classification

We take advantage of image classification to increase the speed of processing the example-based queries. By pre-classifying images, we can significantly reduce the number of images we need to look at while responding to the user's query. We will do so by filtering out classes of phenomena that are not relevant to the user's query. This may seem to be restrictive at first glance, but it is the only practical approach when dealing with large data repositories, where brute force comparison of users' image(s) with all the images stored in the database is simply not feasible. The practical solution is to build accurate classifiers and use them to make sure the

VSO's images are labeled properly, and to match the user's images to the appropriate catalogs.

To classify the images we use a multi-level arrangement of binary classifiers similar to the arrangement shown in Figure 2. This "gradual filtering" approach is founded on other well known classifiers' fusion techniques and is proposed here to quickly prune irrelevant images from further consideration. We believe this technique is very effective when dealing with large repositories of data that contain a high percentage of one type of phenomenon. We first apply binary classifiers, focusing on filtering out sections of the images that do not contain any phenomena (i.e. Quiet Sun images).

The following classifiers will be used later to determine what phenomena occur in portions of the images that were recognized as non-Quiet, and provide a proper label. This allows a classifier to specialize in deciding if a single phenomenon exists in an image without having to worry about the classification results being skewed towards more frequent types of phenomena. This is a well known issue and described in [1, 2, 3]. The system should also scale well, as classifiers will be able to run in parallel and in a distributed fashion on a cluster of computers. The modular design also means that individual classifiers can be modified without affecting the rest of our image labeling system.

A set of solar images, that have been labeled by human experts, have been used for training and testing the classifiers. After the classifiers are trained, they are used to label the remaining images in our repository and add the references about the newly labeled images to the appropriate phenomena's catalogs. Each new image that is put into the database will have an attribute vector, $\vec{i}$, created by our image processing module and is to be labeled by our classifiers (see Information Retrieval component in Figure 1). These two crucial pieces of information are then stored in our repository and used while searching for images. They are used to first quickly filter

out segments of images that are not relevant. We then refine our search by ranking based on evaluating the angle between the query vector, $\vec{q}_{avg}$, and the relevant image vectors, $\vec{i}$, in the catalog.

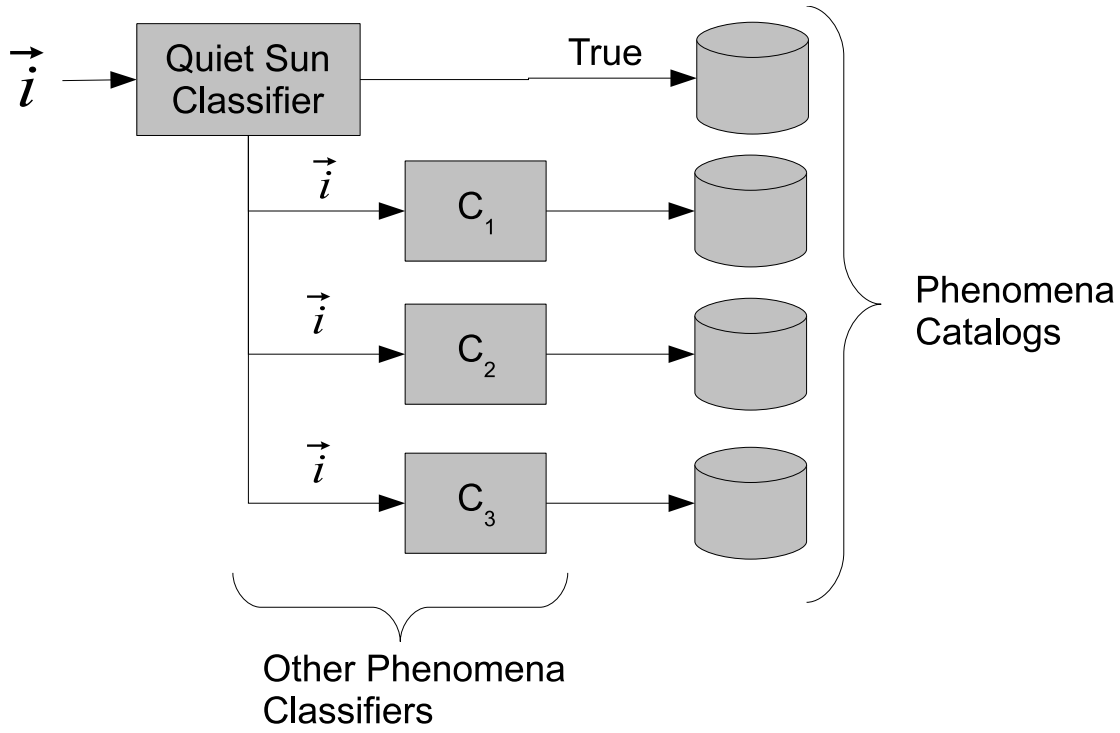


Figure 2: A sample arrangement of binary classifiers to produce a set of labels.

For the development of our prototype we used classifiers present in Weka [4]. Weka is a well known package of data mining tools which provides a variety of known, well maintained classifying algorithms. This allows us to experiment with several kinds of classifiers quickly and easily. Because each image of the sun may contain multiple phenomena, multiple labels could be given to each image by human experts and by our prototype classifier.

Information Retrieval

The Information Retrieval component, as seen in Figure 1 is responsible for analyzing the query image(s) and retrieving similar images from the TRACE image repositories. Distinct features are extracted from the sample image(s) during Image Preprocessing. Classification of the sample image(s) is performed based on the extracted information. After each sample image has been classified, we select similar images from the data catalogs related to the query and rank them from most relevant to least relevant. The ranking function can be modified based on the user's feedback for subsequent queries.

Phenomena Catalogs

The backbone of the system are the Phenomena Catalogs, shown in Figure 1, maintained in our repository. The phenomena catalogs contain collections of pointers to the original images, features that have been extracted from the images, and the results of our classification. The Searching & Ranking component, shown in Figure 1, uses the classification results and extracted attributes from the images to quickly select and rank images that are similar to the query. We decided to utilize pre-classification mechanisms and catalogs to speed up the image retrieval process. Brute force mechanisms, as used in [5], would generate too much of a delay when used with such a large image repository as VSO. The phenomena catalogs will be populated by images from the TRACE mission. As new images are made available, they will be classified and kept in the appropriate catalogs.

Query Interface

The first module, the one users will have direct interaction with, is the Query Interface from Figure 1. The first step is to start a query by providing image(s)

with an example phenomenon. Other constraints, if known to the user, can also be provided, such as dates and wavelengths, to confine the search. After the query has been submitted and results have been returned, the researcher may opt to refine the search. This is done intuitively via interactively selecting images that are interesting and uninteresting to modify the search. The refined query can then be resubmitted and new, more relevant results can be returned for the researcher to further review.

Background

Automatically detecting phenomena in solar images has become a popular topic in recent years. Zharkova *et al.* [6] discuss several methods for identifying features in solar images including Artificial Neural Networks, Bayesian interference, and shape correlation. Five different phenomena are discussed: sunspots, Inference, plage, coronal mass ejections, and flares. Nine techniques for feature detection were used: histograms, Laplacian of Gaussian, Region Growing, Simulated Annealing, Baysian Inference, Artificial Neural Networks, Hough Transforms, Valley Detection, and Multiple Abstraction Level Mining. Each technique was only used to find a specific kind of phenomena. There was no single technique discussed that could find a variety of phenomena.

Zharkov *et al.* [7] did a statistical analysis of sunspots during the years of 1996–2004. Their study showed regular intervals for when sunspots can be expected to appear on the sun. This knowledge could be used for identifying sun spots based on the date the image was taken. During a time when there is higher sun spot activity, there is a greater probability that the phenomena being identified is a sun spot. Likewise during a time of lesser sun spot activity there is a lesser probability that the phenomena is a sun spot.

Turmon *et al.* [8] use a statistical analysis for identifying and labeling active regions of the sun. Three types of structures were analyzed in this study; Umbra and Penumbra from Sun Spots, Faculae, and the background of the sun. Michelson Doppler Interferometer (MDI) pictograms and magnetograms used together can identify phenomena accurately. A Bayesian image-segmentation technique was used for classifying the phenomena. It was trained using pre-labeled images.

Wit [9] used a Bayesian classifier for segmenting solar images and claimed to be able to track structures in near real time. Different features last for different periods of time, such as Solar Flares last for a relatively short period of time compared to Sun Spots. Being able to track structures from image to image could be used in the future as another data point to consider.

Zharkova and Schetinn [10] have trained a neural network to identify filaments within solar images. Unfortunately the neural network was only trained to identify one kind of phenomenon and not many different kinds of phenomena. Training the neural network takes time and it is difficult to know when a neural network has been trained properly, although it runs quickly when identifying filaments. Delouille, De Patoul, and Hochedez [11] used wavelet analysis along with the CLARA clustering algorithm to segment mosaics of the sun.

There have also been many papers from the computer science domain that deal with analyzing and clustering images. Goldberger *et al.* [12] broke down images into features such as texture and shape and modeled these features with a Gaussian Mixture Model (GMM). These GMMs are then grouped together using an Information Bottleneck (IB) method.

Deselars *et al.* [13] used a K-Means clustering algorithm and a Lyman Break Galaxies (LBG) clustering algorithm to group images. For processing the images an integral transformation was performed on the images. This provided a single number

to represent the image. In addition, a histogram of the image was produced as well as a histogram describing the texture of the image.

Hoiem *et al.* [14] broke down images into statistical structures for classification. They used a histogram and texture features from the entire image as well as predefined scalings of the image. The image and down sampled images were also divided into a dynamic grid of cropped images. Histograms and texture features were also extracted from cropped images and used in the classification process. SnapFind [15] took this technique a step farther and integrated it with the Diamond Framework. The Diamond Framework allows interactive searching of images by discarding potential images early in the processing of the images.

Similar Systems

There are only a few image searching systems currently available. Different systems use different approaches. For example, services such Google's Image Search [16] do not have to rely on just the contents of the image, these services can also take advantage of the text content in which the image was found. This can essentially reduce the image search to the use of well known text based search mechanisms where an image is returned instead of text.

Information Retrieval of similar images, based on the features extracted directly from them, as opposed to text-based methods, has also been making progress recently, although data mining of images is a more computationally intense and complex process. Currently research is taking place to help catalog and diagnose medical images as shown in [17, 18, 19, 20]. There has also been a great deal of research dealing with regular photos. This work focuses on extracting features from images and using them for image clustering, classification, and searching purposes [21, 12, 14, 13]. There have also been attempts to apply feature recognition algorithms directly to solar data.

These attempts typically focus on a single type of phenomenon [22, 10, 6] causing them to have limited benefits to the broad community of solar scientists. They have been built by solar physicists to solve their immediate needs and do not focus on a modular, universal approach that scales well to larger data sets and can be improved using feedback. In this research, we want to propose a universal solution, that could help to integrate these efforts.

To the best of our knowledge, there are currently only a few interactive image searching systems available. This leads us to believe that a large scale image searching system is practically achievable, but enough challenges remain to make this a non-trivial research project. Two papers [5, 20] published in 2007 discuss recently developed systems that allow for searching images by example. These two systems extract and analyze information from the images in different ways, suggesting that domain specific approaches are required to achieve the best results.

The first similar image searching system, called MassFind [20], is an interactive image retrieval system developed mainly in Intel Research labs. Its purpose is to help with the analysis of mammograms. A new digitized mammogram, that has not been analyzed, is used as the sample input image. The system returns similar images to the user to allow comparison with the undiagnosed image. The purpose is to allow for better medical analysis of the sample. The process of submitting an image for the retrieval of similar images provided the initial motivation for our work.

The results of this study are evaluated in terms of *precision at n*. This is the number of correct images there are out of n returned images. An image is considered correct if it has the same diagnosis as the test input image. If 20 images are returned and 15 are correct, the *precision at n* would be 0.75. The *precision at n* in this study ranges from 0.59 to 0.68. It should be noted that this system is domain-specific and uses techniques that are not necessarily applicable for dealing with solar images. The

work also clearly shows the practical benefits of building domain-specific systems for this kind of applications.

The second system, known as Diamond SnapFind [23], has been developed at Carnegie Mellon University. This program also allows interactive searching of photo albums. Gibbons *et al.* [5] discuss the latest version for interactive searching using several test scenarios. Some of the scenarios include finding photos containing a sail boat or finding photos from weddings. With the application, users were able to find a desired subset of pictures as requested by the scenarios. Figure 3 shows the querying interface with some results from the wedding scenario.

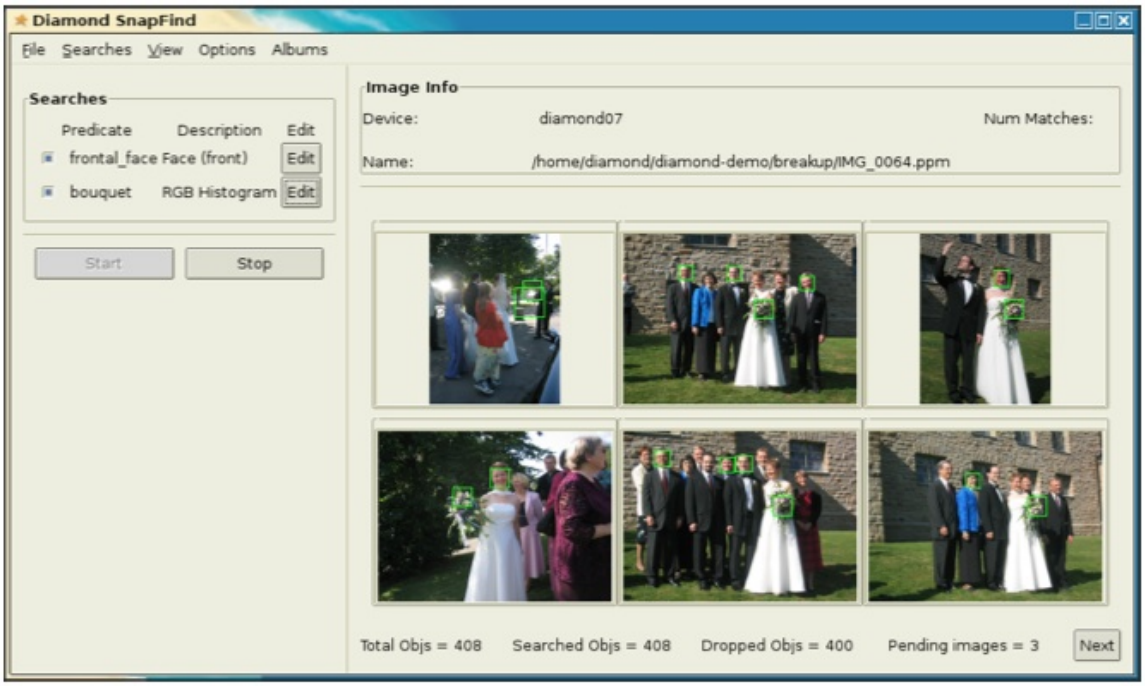


Figure 3: SnapFind displaying results from interactive search.

The scenarios included catalogs of pictures ranging in size from 1072 to 32,796 images. Due to the small size of the data repository, the authors were able to take advantage of some brute force techniques that would not be suitable for our

application. We believe that a well known database searching strategy called "filter and refine" [24] can be developed to handle much larger data-sets such as TRACES's on-line image repositories. To accomplish this we plan to concentrate on characteristics that can be extracted quickly from solar images and distinguish between different types of phenomena well.

Image Format

The images from TRACE are contained in a Flexible Image Transport System (FITS) file. FITS is the data format standard used by astrophysicists from NASA and around the world. While the FITS format has been designed to store any type of information, it is generally used to store images [25]. Each file contains a header section and a section containing the data. The header section many contain extra information about the file such as the time the image was taken and the resolution of the image. Image data is generally in 8, 16, 32, or 64 bit grayscale image [26, 27]. There are several libraries available for reading and writing the FITS file format including libraries for Perl, C/C++, and Fortran [28].

There are also many programs for reading and writing the FITS format. These include the open source FV [29] and DS9 [30] programs. The TRACE group uses SolarSoft [31] for analyzing the images. SolarSoft requires the installation of the IDL language and programming environment [32] to run properly.

The TRACE Satellite

The TRACE [33] satellite telescope is able to take a picture of the sun approximately once every three minutes. The image is in a high resolution with each pixel representing 0.5 arc seconds of the sun. Unfortunately the majority of the images are 1024 pixels wide by 1024 pixels high, so each image only represents a small

portion of the sun. TRACE is also able to take images in many different wavelengths. This is done because different phenomena are visible in different wavelengths. The wavelengths TRACE is able to take images at are: 171Å, 195Å, 248Å, 1216Å, 1550Å, 1600Å, 1700Å, and White Light [34].

Classifiers, Evaluations & Measures

For this project we trained two classifiers, C4.5 and Support Vector Machines (SVM) [1]. These two classifiers were chosen based on preliminary experiments with the labeled dataset. The boosting algorithm, AdaBoost, was also applied to help increase the accuracy of the classifiers. C4.5 is a tree based classifier that uses the attributes in the feature vector to split the dataset into smaller datasets. The attribute to split on is chosen by the amount of information gain from splitting on the particular attribute. The algorithm recurses on the smaller datasets until a tree is built that will properly classify the training data.

SVM classifiers produces a hyperplane in n-dimensional space that linearly separates points by their class into 2 regions. The algorithm will attempt to find a boundary between the two classes with the maximum margin or distance between them. The larger the margin, the better the classifier is able to accurately classify future samples.

In addition to using the C4.5 and SVM algorithms, we augmented the creation of the classifiers with AdaBoost. AdaBoost causes classification algorithms to produce different classifiers based on misclassified instances from a previous generation. It does this by modifying weights associated with each instance. For misclassified instances the weights are increased and likewise for correctly identified instances the weights are decreased.

To evaluate the classifiers and IR system we use a selection of measurements that are commonly used to evaluate classifiers and Information Retrieval systems. They are Precision, Recall, F-Measure, and ROC-Curves.

Precision, Recall, and F-Measure are three measures that are used while analyzing the accuracy of our classifiers. Precision, as shown in Equation 1, gives the fraction of items that have been correctly classified. Recall, as shown in Equation 2, is the probability that a given item from that class will be selected. In unbalanced data sets, measures such as Recall alone aren't enough to analyze the quality of a classifier. If a classifier returns every item in the data set, the Recall will be 100%. F-Measure, as shown in Equation 3 is the harmonic mean between precision and recall. Higher values for recall and precision make the f-measure value approach 1.

$$\text{Precision} = \frac{|\{\text{Human Label}\} \cap \{\text{Classifier Label}\}|}{|\{\text{Classifier Label}\}|} \tag{1}$$

$$\text{Recall} = \frac{|\{\text{Human Label}\} \cap \{\text{Classifier Label}\}|}{|\{\text{Human Label}\}|} \tag{2}$$

$$\text{F-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Receiver Operating Characteristic (ROC) curves are a useful tool for determining the accuracy of a binary classifiers. ROC curves were developed to help analyze radar images. The ROC curve plots the true-positive rate on the y-axis and the false-positive rate on the x-axis. The area under the curve represents the accuracy of the classifier overall. The closer the area under the curve is to 1.0, the more accurate the classifier. The accuracy of the classifier decreases as the area under the curve approaches 0.5. At 0.5 the classifier would not be any better then randomly picking a class.

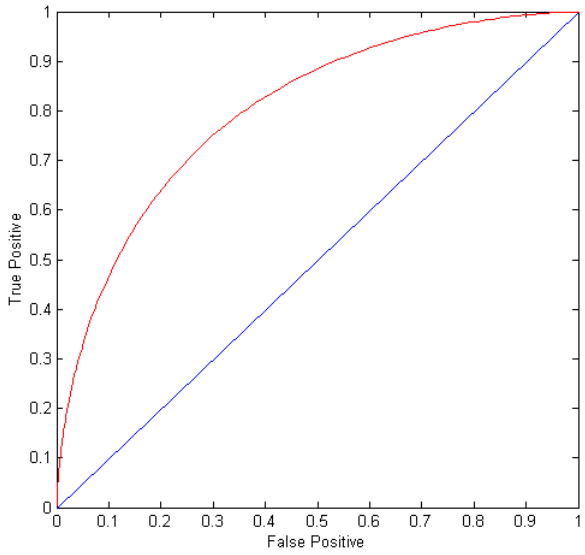


Figure 4: A sample ROC curve

Modifications of Precision and Recall are used for the analysis of the IR system. These modifications are shown in Equations 4 and 5. The modifications take into account the ranking of the returned images. The modified equations return the recall or precision at rank k. The set of relevant images returned by the system is denoted a I_q. A relevance list is created for each position, $(r_1, r_2, \cdots, r_n)$ where $r_i = 1$ if $i_i \in I_q$ or 0 if $i_i \notin I_q$. The set of relevant images out of set I is denoted as I_q. In our case there are 32 images in I, and the number of images in I_q varies depending on the query.

$$\text{Recall}(k) = \frac{1}{|I_q|} \sum_{1 \leq i \leq k} r_i \tag{4}$$

$$\text{Precision}(k) = \frac{1}{k} \sum_{1 \leq i \leq k} r_i \tag{5}$$

Precision for rank k is calculated with Equation 5. Recall for rank k is calculated with Equation 4. Recall is the number of relevant images returned at rank k divided by the total number of relevant images, I_q in the set.

IMAGE PROCESSING

In this section we discuses how the images are processed from the FITS file format into the feature vectors we use. First we discuss the set of images we use for this project. Then we discuss how the images are cleaned up and normalized. Third, the process of extracting information for the creation of our attribute vectors is discussed. Finally we discuss sampling techniques we used for creating the classifiers.

Data Set

The TRACE images were downloaded from the TRACE image repository. The images are from 1999 and 2001, when there was high activity on the sun and from 2007, when there was low activity in the sun. These images have been hand labeled by students who study the sun. We have 232 images labeled with 433 labels. The labeling of images continues to help increase the variety of images for future research projects.

Concerning distribution of the samples, we currently have labeled 232 images whose attributes are analyzed. The phenomena we have identified in these images are: coronal loops, filaments, flares, and sun spots. There are 276 coronal loops, 13 filaments, 7 flares and 17 sun spots labeled. The remaining 120 labels are of regions where no phenomena exist that we are calling the empty-sun.

As for the wavelength distributions, for the labeled images there are 22 in the 171Å spectrum, 43 in the 195Å spectrum, 8 in the 1216Å spectrum, 1 in the 1550Å spectrum, 7 in the 1600Å spectrum, and 5 in the white light spectrum. Due to the way the TRACE satellite takes images, there are many more images in the 171Å and 195Å spectrums.

A large number of the images tend to contain empty-sun, and are not very interesting for further analysis. By identifying what parts of the sun are active, we can reduce the areas that need to be analyzed with more complex methods. To sample images for this analysis, we randomly selected labeled images in a distribution where half of the selected images contained the empty sun. The remaining images selected contained a phenomenon. The samples were normalized with a min-max normalization with the minimum and maximum numbers coming from the entire set of images and not just the sampled images.

Preprocessing the Images

The first piece of software the images are processed with is called SolarSoft. SolarSoft is written in the IDL language and used is by researchers studying TRACE images for analyzing the solar images. While SolarSoft is difficult to learn, the IDL language provides a powerful scripting environment for manipulating data. SolarSoft provides many useful functions for manipulating and viewing TRACE images. We use SolarSoft's built in functions to help remove noise, normalize, and save the images into the TIFF file format.

The command *trace_prep* is used to prepare raw images for further manipulation. To prepare images it fills in pixels with a value of zero with the mean pixel value of the image. Then it replaces saturated pixels with a value less then 4095. Third it subtracts the dark pedestal and current from each image. Optionally *trace_prep* will also call other functions for the images. These functions remove radiation belt spikes and streaks, remove the background diffraction pattern, normalize each image, and correct pointing in the index structure. Included in the header file of the image is calibration information and *trace_prep* can take advantage of this information for

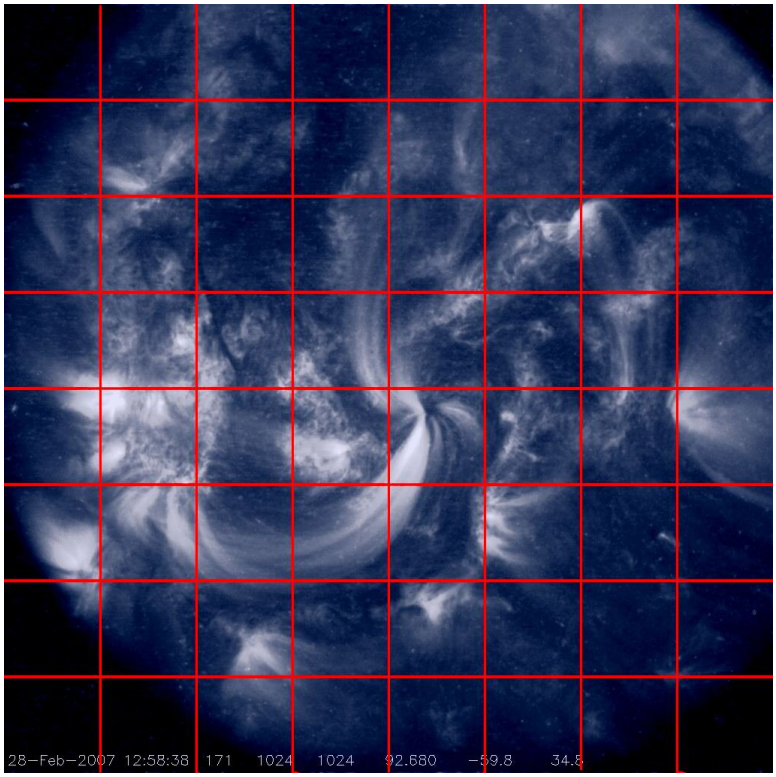
adjusting the colors in the image. The *trace_prep* function is an on going piece of work in SolarSoft and new features are being added as they are required.

trace_read is a method that allows SolarSoft to read in FITS images. When used in combination with a filter, specific images can be read. A filter can be for a certain time frame, a certain frequency, or a certain region of the sun. A corresponding method called *trace_write* allows for writing manipulated images back to disk in a variety of formats including JPEG, FITS, TIFF, and BITMAPS.

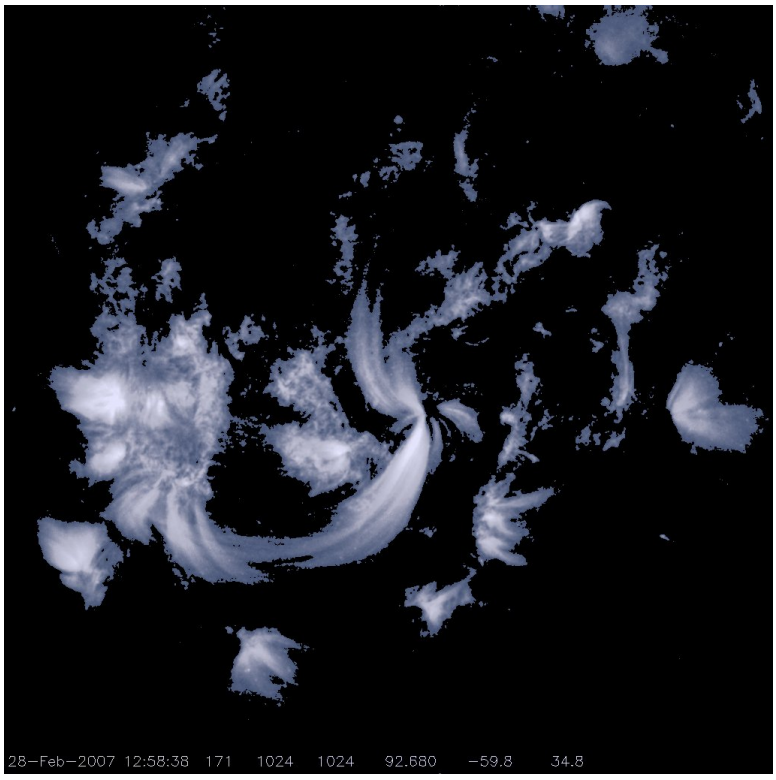
Segmenting the Images

When extracting features from images, what part or parts of the image these features should be extracted from needs to be decided. Features can be extracted from the entire image or from sections of the image. Currently two techniques are used for segmenting images into sub regions. In the first technique a fixed Grid Segmentation (GS) is used. The image is broken up into 128 by 128 pixel non overlapping sub-images. This gives us an 8 by 8 grid over a 1024 by 1024 sized solar image. The grid size is also convenient because it is a powers of two number and the corner grids tend to be filled with black as seen in Figure 5(a). The attributes as described in the Feature Extraction section are extracted from each sub image. A sample image is shown in Figure 5(a).

For the second technique a Thresholding Segmentation (TS) technique is used. All pixels with less then a certain intensity value are marked as black and pixels above that value are marked as white. This gives a segmentation map that can be applied to the original image. Groups of contiguous black or white pixels containing 500 or more pixels have features extracted as described in the Feature Extraction section.



(a) *Grid Segmentation Example.*



(b) *Threshold Segmentation Example.*

Figure 5: Examples showing how we are currently segmenting the solar images

Regions containing less then 500 pixels tended not to be very interesting and were omitted. A sample image is shown in Figure 5(b).

Feature Extraction

To produce $\vec{i}$, the images are first segmented into smaller regions for the extraction of texture information [35]. We have worked with two kinds of image segmentation that is performed on the images. The first is Gridded Segmentation [5, 36] as shown in Figure 5(a). This technique breaks the image into 128 by 128 pixel blocks and is very fast. Another approach we implemented is Threshold Segmentation [37], as shown in Figure 5(b), where the appropriate threshold value was derived by values from the quiet sun. It breaks the image into darker and lighter regions. Adjoining pixels are considered a segment from which to extract texture information.

Every image that is put into the system is processed and an attribute vector, $\vec{i}$, is created to represent the image, as reflected in the Image Processing module in Figure 1. These techniques are commonly used in search engines [38, 39] and will increase the search speed [21, 40] in the case of large data repositories such as TRACE. Representing the images in the form of attribute vectors also allows us to compute the similarity of two images by calculating the angle between the two image vectors. The attribute vector, $\vec{i}$, is used as input for training the classifiers, labeling new images, analyzing queries, and ranking the relevance of returned images. The vectors are stored in the database associated with the original images. An attribute vector is used instead of the entire image because it is much smaller and easier to manipulate then an image.

$$\mu_n = \sum_{i=0}^{L-1}(z_i - m)^n p(z_i) \quad (6)$$

$$R = 1 - \frac{1}{1+\sigma^2(z)} \quad (7)$$

$$e = -\sum_{i-0}^{L-1} p(z_i)\log_2 p(z_i) \quad (8)$$

$$U = \sum_{i=0}^{L-1} p^2(z_i) \quad (9)$$

The values in our attribute vector $\vec{i}$ reflect different types of texture information extracted from the intensity of the images and sub-images. The attributes extracted so far are the mean intensity ($\bar{z}$), the standard deviation of the intensity (σ_z), the Third Moment (μ_3) and Fourth Moment (μ_4) as in Equation 6, Uniformity as in Equation 9, Entropy as in Equation 8, and Relative Smoothness as in Equation 7. We have chosen these characteristics for our preliminary investigation because the values they produce are not influenced by different orientations of the same kinds of phenomenon in different images. These attributes can also be extracted from the images quickly, an important aspect when dealing with large sets of VSO images. Table 1 shows the minimum and maximum values generated from our training data.

	For Gridded Segmentation:			For Threshold Segmentation:		
Image Feature:	Min	Max	Δ	Min	Max	Δ
Mean ($\bar{z}$)	0	253.0	253.0	87.846	236.773	148.926
Std. Dev. (σ_z)	0	108.067	108.067	2.067	74.308	72.241
Entropy (e)	0	7.225	7.225	0.816	6.906	6.090
3^{rd} Moment (μ_3)	-1,348,610	790,790	2,139,400	-204,512.0	144,632.0	349,144
4^{th} Moment (μ_4)	0	$2.473 * 10^7$	$2.473 * 10^7$	56.698	$3.486 * 10^7$	$3.486 * 10^7$
Rel. Smoothness (R)	0	0.153	0.153	$6.67 * 10^{-5}$	$7.94 * 10^{-2}$	$7.93 * 10^{-2}$
Uniformity (U)	$7.07 * 10^{-3}$	1.0	0.992	$9.55 * 10^{-3}$	0.637	0.627

Table 1: Minimum and Maximum Values generated by each extracted feature.

To improve our search capabilities, we decided to extend our $\vec{i}$ with additional non-texture features extracted from the headers of the images in the FITS file format. These include items such as the wavelength in which the image was taken, the arc second of the pixels and the center of the sun relative to the image. These features along with extracted texture features, that were mentioned earlier, are used to create an $\vec{i}$ for each solar image.

Supporting Image Libraries

Several approaches have been evaluated for producing attributes from TRACE images for proper phenomenon recognition. These approaches include Gabor filters [19, 41] and Gaussian Mixture Models [42]. OpenCV [43] includes many different algorithms for transforming and analyzing the structure of images. OpenCV also provides many of these methods for analyzing images. Some of the methods provided by OpenCV include an algorithm to automatically extract blobs or connected components from the images. The blobs contain interesting objects from the image that could be further processed. Another method that produces Eigenvectors and Eigenvalues for images was also investigated. A third method that was investigated from OpenCV is a pyramid segmentation algorithm.

Sampling

Sample datasets were created to train the classifiers. The sample datsets have an equal distribution between the label we are training to identify and all of the other labels. Two different techniques were used in order to minimize the effects of our imbalanced dataset. These techniques are Random Under Sampling (RUS) and Random Over Sampling (ROS) as described in [44].

RUS is sampling without replacement. This method ensures that a data sample is never represented twice in our sample dataset. It is used because we want the classes to be equally represented. This technique focuses on labels that are well represented in our dataset such as the empty-sun and coronal loops.

ROS is the process of oversampling our labeled data. We use a random selection with replacement, and ensure that 50% of our sampled images contain the label of interest. This guarantees that the label in question is equally represented, although many of the selected elements may be duplicates. This technique focuses on labels that are underrepresented in our dataset such as flares and filaments.

For the two labels we have the most, Empty-Sun and Coronal Loop, we performed RUS selecting 10% of the instances with an equal distribution between the two labels. For each label, 10 sample sets were generated for the classifiers to be trained and evaluated. With the remaining labels we limited our random sample size to 50 because there are a fewer number of data instances with these other types of solar phenomenon. In this case we used both sampling techniques. Once again 10 sample sets were generated for each sampling technique and each label.

Benchmark Creation

Part of this work is to create a set of standardized solar images that can be used as a benchmark for solar image IR systems. This is similar to the TREC[45] dataset of text documents that is commonly used to benchmark text based IR systems. The TREC dataset contains a set of documents and a standard set of queries for evaluating the IR system. Our dataset contains about 1600 images along with 6 sample query images. This paper provides results from our IR system using the set of images we have selected.

TRAINING CLASSIFIERS

There are several difficulties that need to be understood with our classification problem so the correct decisions can be made. First of all there are multiple classes to choose from for each image. The second issue is that each image may also by labeled multiple times with several different classes. Having multiple classes isn't necessarily a problem. Zhang *et. al* [46] state that multi-class problems can always be reduced to multiple binary classification problems. Our solution for this is laying out the binary classifiers in a way that will give images their proper label.

Another issue is that training data sets are inherently imbalanced and this imbalance needs to be understood and compensated. The labels in the data sets are imbalanced because the phenomena in the images appear over time at different rates. AdaBoost is one technique that can be used to help compensate for imbalanced data sets. We are also using two sampling techniques as described in Chapter 2.

For the first version of our system, several types of classifiers have been trained for each potential label. These classifiers are C4.5, AdaBoost C4.5, SVM and AdaBoost SVM. In prior investigations, C4.5 has been useful as a binary classifier for determining areas of images that contain the empty sun or not. AdaBoost has been shown to increase the accuracy of many classifiers, and were used in an attempt to improve these classifiers while training them. A SVM classifier is used as part of the work in [47, 46, 48] for classifying medical images and texture features, therefore we believe that SVM classifiers can also be useful for classifying phenomena in our images.

Evaluation of Extracted Features

First we evaluate features we have extracted to see how well they work for being able to identify phenomenon in an image. We use *ReliefFAttribue* with *Ranker* search and *CFsubSet* with a *BestFirst* search using 10-fold cross validation.

ReliefFAttribute gives each attribute a merit in the range -1 and 1. A positive merit means the attribute is a good selector. A negative merit means the attribute is not a good selector [49]. This algorithm works by first randomly selecting an instance. It then finds the k nearest neighbors that have the same label and the k nearest neighbors that have different labels. The weights for each attribute are then adjusted based on the difference between the selected instances and the nearest neighbors.

CFsSubSet evaluates sets of attributes instead of attributes individually. It will first compute a correlation matrix, then it uses a heuristic to find a set of attributes that have high merit. Attributes that are highly correlated with the class and have low inter-correlation have higher merit [49].

With 10 fold cross validation, the set of data is split into 10 equal sized partitions with the classes represented with approximately the same ratios as the entire set. The evaluation is run 10 times, each time a different partition is used as the test data and the rest of the data is used as the classification data [4].

The results from *ReliefFAttribute* with *Ranker* search from Table **??** show us that the mean and wavelength attributes are good at determining if a phenomena exists in a particular region. The results from *CFsSubset* with *BestFirst* search from Table **??** show us that the mean and third moment attributes are good at determining if a phenomena exists in a particular region or not using the Threshold Segmented images. For both Grid Segmented images and Threshold Segmented images, using

ReliefFAttribute ranking give wavelength as the attribute with the most merit. In both cases the entropy attribute is ranked second.

By evaluation of the features we can conclude that the mean is a good indicator to identify if a phenomenon exists in a region or not. We believe this is because regions where phenomena exits tend to have a significantly different average then the surrounding areas. The wavelength and entropy attributes are good indicators to identify what phenomenon is in the region of the image. The wavelength attribute is ranked highly because different phenomena tend to show up in certain wavelengths, but not others.

Rank (i)	Attribute	Merit	Δ i-(i+1)	Weighted Average
1	Wave Length	0.235	0.157	1.099
2	Mean	0.078	0.016	0.468
3	Std. Dev.	0.062	0.011	0.310
4	Entropy	0.051	0.006	0.204
5	R	0.045	0.007	0.135
6	Fourth Moment	0.038	0.014	0.076
7	Third Moment	0.024	0.009	0.024
8	Uniformity	0.015		0.000

Table 2: Comparing phenomena to the empty sun.

Rank (i)	Attribute	Merit	Δ i-(i+1)	Weighted Average
1	Wave Length	0.360	0.161	2.520
2	Mean	0.199	0.136	1.194
3	Std. Dev.	0.063	0.009	0.315
4	Entropy	0.054	0.008	0.216
5	R	0.046	0.028	0.138
6	Third Moment	0.018	0.002	0.036
7	Uniformity	0.016	0.007	0.016
8	Fourth Moment	0.009		0.000

Table 3: Comparing phenomena to the empty sun.

Attribute	Sample 1	Sample 2	Sample 3	Sample 4
Wave Length	10	9	10	10
Mean	10	10	10	10
Std. Dev.	9	0	10	10
Entropy	1	0	0	0
Third Moment	0	0	1	1
Fourth Moment	0	0	0	0
R	0	0	0	0
Uniformity	0	5	4	8

Table 4: Comparing phenomena to the empty sun.

Attribute	Sample 1	Sample 2	Sample 3	Sample 4
Wave Length	0	0	0	0
Mean	10	10	10	0
Std. Dev.	0	0	0	0
Entropy	0	0	0	0
Third Moment	10	10	10	10
Fourth Moment	0	0	0	0
R	0	0	0	0
Uniformity	0	0	0	8

Table 5: Comparing phenomena to the empty sun.

Training of Classifiers

When creating a sample data set for a particular label, we create a new temporary data set for that label. For example, when creating classifiers for the Empty-Sun, we first give all of the data that is not labeled with the empty sun the label of 'other'. This would cause the 'other' label to reflect the frequency of phenomenon occurrences. From this new data set we randomly sample and create a sample data set for training.

Two series of tests were performed. First, for the two labels we have the most of, Empty-Sun and Coronal Loop, we performed RUS selecting 10% of the instances

with an equal distribution between the two labels. For each label, 10 sample sets were generated for the classifiers to be trained and evaluated. The average AUC for these tests are presented.

C4.5 and Support Vector Machine were used for classifiers. AdaBoost was also applied to these two classifiers. These 2 classifiers performed the best based on our previous investigations where we also analyzed ID3, and Bayesian classifiers. A 10-fold cross validation was used when generating the results. These two classifiers were chosen based on knowledge gained from prior experiments. The average AUC and F-Measure values are presented in Tables 2 and 3.

	C4.5	SVM	AdaB C4.5	AdaB SVM
AUC	0.915	0.921	**0.963**	0.900
Precision	0.916	0.902	**0.910**	0.910
Recall	0.930	0.941	**0.935**	0.902
F-Measure	0.922	0.921	**0.922**	0.920

Table 6: Average values for detecting the Empty-Sun.

Table 2 shows the results for the Empty-Sun labeled data. Overall, the classifiers we produce have a large AUC. For the C4.5 classifier, AdaBoost tended to help increase the accuracy of the classifier. With the SVM classifier AdaBoost tended to decrease the accuracy of the classifiers. Overall, these results show that for recognizing the Empty-Sun, the C4.5 classifier with AdaBoost gave the best results.

Table 3 shows the results for the Coronal-Loop labeled data. The sample data sets and results were generated in the same manner as the Empty-Sun data. Once again the classifiers generated have high accuracy. While SVM with AdaBoost does not lose accuracy over the standard version like the Empty-Sun labeled data, C4.5 with AdaBoost once again performs the best overall. Initially we had very high hopes for SVMs, as these classifiers tend to be fast and accurate. Now, we believe that

our results show that sampling, forced by our unbalanced data, can cause removal of instances along the maximum margin hyperplanes, which resulted in a decrease of an average SVM's accuracy.

	C4.5	SVM	AdaB C4.5	AdaB SVM
AUC	0.943	0.947	**0.976**	0.969
Precision	0.954	0.981	0.958	**0.963**
Recall	0.931	0.910	**0.942**	0.931
F-Measure	0.942	0.945	**0.950**	0.946

Table 7: Average values for detecting the Coronal Loops.

Table 4 shows us the average AUC for both ROS and RUS for every label. Table 5 shows us the average F-Measure for both ROS and RUS for every label. Overall these classifiers are quite good with a majority of them having an AUC of over 0.90.

	C4.5		SVM		AdaB C4.5		AdaB SVM	
Phenomenon	RUS	ROS	RUS	ROS	RUS	ROS	RUS	ROS
Empty Sun	0.795	0.872	**0.940**	0.923	0.920	0.912	0.939	0.920
Coronal Loop	0.897	0.905	**0.932**	0.922	0.911	0.917	0.912	0.912
Sun Spot	0.890	0.917	0.922	**0.958**	0.901	0.932	0.944	0.943
Filament	0.838	**0.960**	0.832	0.848	0.875	0.783	0.898	0.897
Flare	0.977	0.970	**0.988**	0.980	0.977	0.967	0.977	0.976

Table 8: Average AUC for ROS and RUS.

	C4.5		SVM		AdaB C4.5		AdaB SVM	
Label	RUS	ROS	RUS	ROS	RUS	ROS	RUS	ROS
Empty Sun	0.873	0.885	**0.936**	0.883	0.900	0.862	0.929	0.925
C. Loop	0.880	0.883	**0.932**	0.917	0.886	0.900	0.927	0.914
Sun Spot	0.875	0.924	0.914	**0.956**	0.891	0.923	0.933	0.941
Filament	0.840	0.847	**0.864**	0.853	0.840	0.837	0.845	0.814
Flare	0.975	0.967	0.987	0.979	0.967	0.955	**0.988**	0.979

Table 9: Average F-Measure for RUS and ROS.

With the second technique we created classifiers for each type of label. Because there are fewer numbers of data instances with these other types of solar phenomenon, we limited our random sample size to 50, with an equal distribution between the label in question and the 'other' label. Two different sampling techniques, ROS and RUS, were also used. Once again 10 sample sets were generated for each sampling technique and each label. The average AUC and F-Measure are presented in Tables 4 and 5. The minimum and maximum AUC are presented in Tables 6 and 7.

	C4.5		SVM		AdaB C4.5		AdaB SVM	
Phenomenon	Min	Max	Min	Max	Min	Max	Min	Max
Empty Sun	0.795	0.965	0.838	1.0	0.835	0.972	0.830	1.0
Coronal Loop	0.837	0.972	0.883	0.960	0.846	0.973	0.851	0.978
Sun Spot	0.793	0.966	0.890	0.975	0.834	0.949	0.944	0.974
Filament	0.700	0.909	0.799	0.870	0.803	0.975	0.840	0.955
Flare	0.958	1.0	0.978	1.0	0.958	1.0	0.958	1.0

Table 10: Minimum and Maximum AUC values using the RUS technique.

	C4.5		SVM		AdaB C4.5		AdaB SVM	
Phenomenon	Min	Max	Min	Max	Min	Max	Min	Max
Empty Sun	0.800	0.946	0.876	0.960	0.881	0.949	0.835	0.972
Coronal Loop	0.802	0.977	0.845	1.0	0.872	0.976	0.800	1.0
Sun Spot	0.858	1.0	0.917	1.0	0.842	1.0	0.878	1.0
Filament	0.780	0.954	0.746	0.904	0.793	0.936	0.942	0.942
Flare	0.881	1.0	0.934	1.0	0.903	1.0	0.952	1.0

Table 11: Minimum and Maximum AUC values using the ROS technique.

In general using the second technique of selecting smaller sample sets, generates less accurate classifiers for the Empty-Sun and Coronal Loop labeled data. This is probably due to the lower sample size of 50 instances instead of using a larger 10% of the labeled data set per random sample.

The Filament labeled data had the least accurate classifiers generated. All of the classifiers generated for the filament data had an average AUC of under 0.90, except for the C4.5 classifier using RUS. The same sample data sets also created classifiers with the lowest AUC when AdaBoost was applied to C4.5. The Flare labeled data had the best classifiers generated. The RUS technique generated the best classifiers as opposed to the ROS technique for this label. The classifiers for the Coronal Loops are also quite good. Overall the values are quite high. The AUC is highest for the AdaBoost C4.5 at 0.976 and lowest for C4.5 at 0.943. The remaining tables show the minimum and maximum AUC and F-Measure values produced.

The highest AUC of 0.982 was achieved by the AdaBoost C4.5 classifier, while the lowest of 0.971 was given by the C4.5 classifier. For the Empty-Sun classifiers, the lowest average AUC was 0.900 for the AdaBoost SVM classifier. The highest average AUC achieved was 0.963 for the AdaBoost C4.5 classifier.

Our results show the average values produced by the classifiers are quite good overall. The expert originated choice of the labeled images could be part of the reason these results are so good. We expect these numbers to change as the data set increases in size and more randomly chosen images are labeled.

Conclusions

Overall the classifiers generated by our labeled images have a high accuracy. By fusing classifiers, we hope to increase the accuracy of identifying phenomena in images. We can conclude, by comparing the classifiers generated for the Empty-Sun and Coronal-Loop in the two tests, that having a larger sample data set will give us better classifiers. As the number of labeled images increases, we should be able to generate better classifiers for all of the labels.

There are instances where using AdaBoost helps and instances where AdaBoost will hinder the creation of the classifier. With the larger labeled data sets of the Empty-Sun and Coronal-Loop, AdaBoost helps to increase the accuracy of the classifier. In the case of the Flare data, AdaBoost tended to decrease the accuracy of the classifiers generated.

The next step is creating an information retrieval system based around these classifiers. We plan on using the information gained from this work to create a set of cascading classifiers, where the results of one stage will determine what classifiers are used next on the image in question. Because the Empty-Sun represents a majority of the images and the generated classifiers have a high accuracy, this would be a good first classifier for processing the images. This will reduce the amount of work the other classifiers need to do in terms of number of images to process and the variety of data they will need to use to determine what the image contains.

For further development of our IR system, are incorporating the C4.5 classifiers with AdaBoost for identifying the Empty-Sun and Coronal Loops. This particular classifier was able to best identify these portions of the images in our current data set. To identify Sun Spots, using a SVM classifier using ROS produced the best result. To identify Flares, we will use a SVM classifier with RUS. Our results show us that Filaments will be the most difficult to identify reliably, though the C4.5 classifier using ROS has produced the best results so far, and is used in our IR system.

INFORMATION RETRIEVAL

The Information Retrieval (IR) component of the system allows a user to submit a sample image containing a phenomenon and have similar images returned to the user. The image based searching mechanism will provide a convenient way for solar scientists to search for images by using an image that contain an event of interest. This chapter discusses the components of the IR system, the procedure we used for testing the system, and evaluation of the search results.

System Components

For searching the database a researcher can submit sample image(s) in a FITS, JPEG, or TIFF image format, along with other search criteria, to the system. If the uploaded image is a JPEG or TIFF file, we assume that the file has already been processed by *trace_prep* from SolarSoft. If the submitted image is in the FITS format, the system runs *trace_prep* on the image and internally saves it as a TIFF file. The uploaded image is then displayed to the user so a region of interest can be selected.

The sample images are pre-processed in the same way as discussed in chapter 2, we denote an individual feature vector for our query as $\vec{q}$. After $\vec{q}$ labels for the images are produced, using our classifiers discussed in chapter 3 (Classifier & Multi-labeling module in Figure 1). The extra constraints can be items listed in the headers of FITS images, such as wavelength or date ranges the search should focus on.

To improve our search capabilities, we decided to extend our $\vec{i}$ with additional non-texture features extracted from the headers of the images in the FITS file format commonly used by NASA. These include items such as the wavelength the image was taken in, the arc second of the pixels and the center of the sun relative to the image.

These features along with extracted texture features that were mentioned earlier, are used to create an $\vec{i}$ for each solar image.

A common query vector, denoted by $\vec{q}_{avg}$, with the same features as in $\vec{i}$, is produced by calculating an average sum of the image vectors as shown in Equation 10, where Q is the set of sample images provided by the user. Now the query vector represents an average of the user's sample image(s) shown by arrows 4–5 in Figure 1. To limit the amount of original images that need to be processed, the labels produced by the classifiers for querying images are used for filtering. Collections of labels generated by our classifiers, extended with labels generated for our "average sample" vector, $\vec{q}_{avg}$ are used to limit catalogs to only those which have the same labels. Images in these catalogs that have the most similar $\vec{i}$'s are returned.

$$\vec{q}_{avg} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \vec{q}_i \tag{10}$$

This is the time where the similarity based ranking happens. For each catalog recognized as relevant to the user's sample, an angle between $\vec{q}_{avg}$ is calculated and the $\vec{i}$'s stored in the catalog. The ranking is based on the degree of its angle to $\vec{q}_{avg}$. The user can also limit the number of results that are displayed, or retrieve the next set of relevant results.

The ordering function is based on a cosine similarity function. This function was chosen for our preliminary analysis because it is simple to implement, and it is commonly used in text based IR systems. One of the major benefits of using a cosine similarity function is it measures the similarity on the angle between two vectors instead of the distance between two points. This has the benefit of making two vectors that point in the same direction have a small angle, even though the vectors may have very different lengths.

Evaluation of our Sample-based IR

The images were submitted to the system and the region in question was selected. The top 32 images were verified by eye to determine if they contained a similar phenomenon to the sample image or not. For example, if the sample image contained a coronal loop and the returned image in question contained an coronal loop, this result was given a 1. If the returned image did not contain a coronal loop, the result was given a 0. The results were used as input into Equations 4 and 5. Recall vs. Precision graphs were created by plotting points where the input for k was in the range 1 to 32.

The images used for our results are presented in the Appendix in Figures 11–14. The first image was from February 28, 2007. The wavelength the image was taken in is 171Å, and the image contains a Coronal Loop. The next two images were taken on the same day of March 26, 2000 and are both of the same region. The wavelength the images were taken in are 195Å and 171Å. The final image was also taken on March 26, 2000 and this image was taken in the White Light spectrum. This image contains a sun spot.

Initially recall is a low number, because we are looking at the total number of accurately returned images up to 32 instead of just up to k. A perfect graph would have precision with a value of 1 all for every corresponding recall along the x-axis.

The graphs we present in our results show the recall versus the precision for the queries. First we show the average of the queries in Figure 6. This graph shows all of the queries averaged together, giving the average precision of 0.75 overall. The result of the best query, from the second query image, is shown in Figure 8. The results for the third query image, shown in Figure 9, are similar to the results from the second query image. This is not surprising as both images come from the same

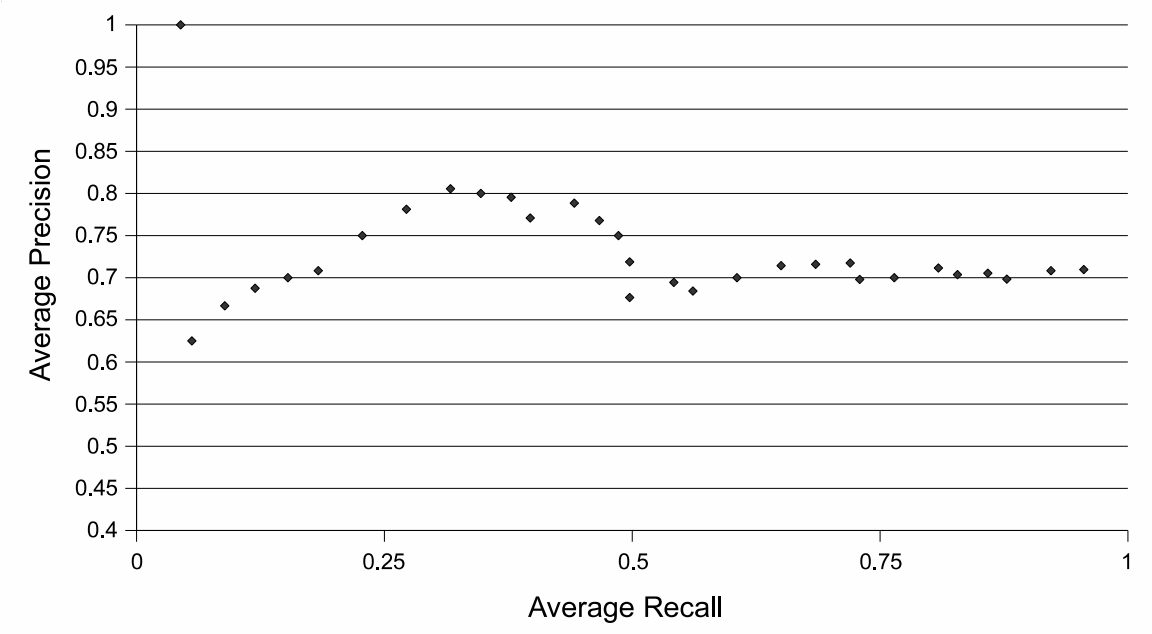


Figure 6: Average Recall vs. Precision for all of our sample images.

day and region of the sun. The largest difference is the wavelength the two images were taken.

The results in the graphs show a range of precision varying between 0.5 and 1.0. The 1.0 value happens when k is 1, meaning that the first result returned for the query is accurate. The average precision for our best query is 0.83. The lowest average precision value for our queries is 0.59. The average precision averaged over all of our queries is 0.75, meaning that on average, we can expect 75% of the returned images in our queries to be relevant.

These results show us that for a given search, we can expect more then half of the returned images to be relevant. In these results for the best case we can expect 83% of the returned images to be relevant. The images containing coronal loops also had

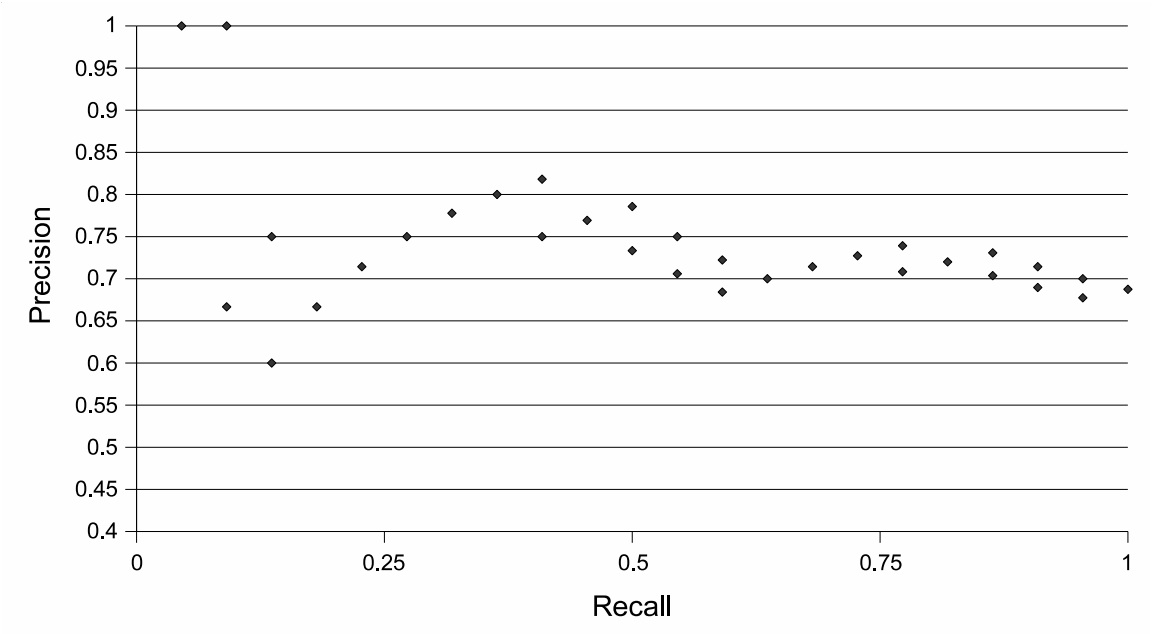


Figure 7: Results from the first sample query image containing a coronal loop.

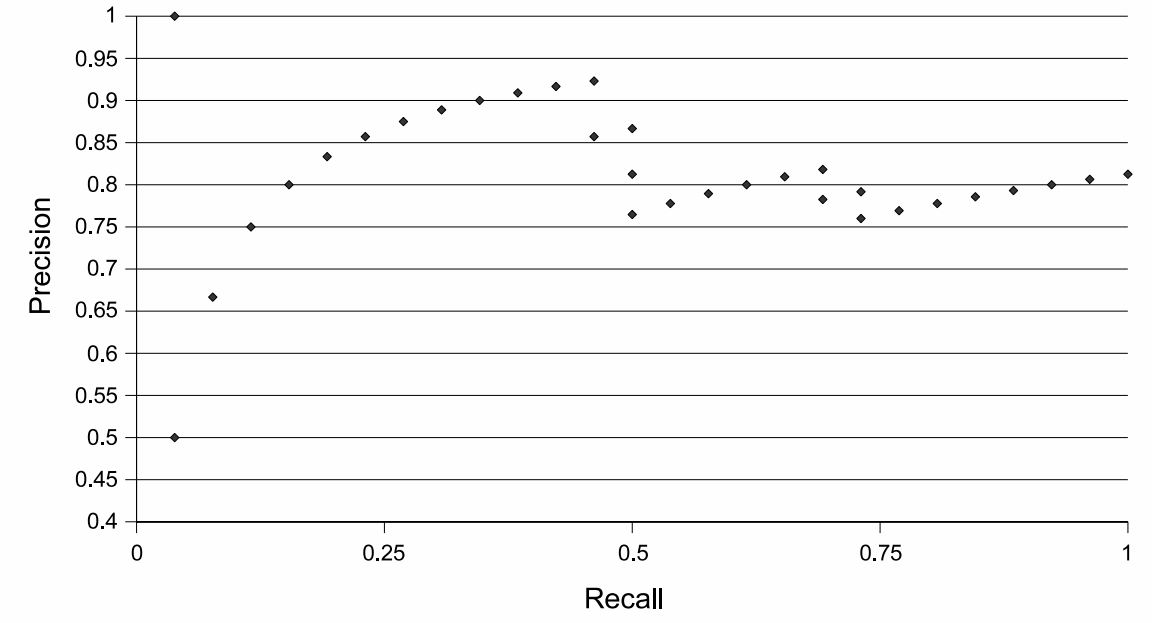


Figure 8: Results from the second query image containing the best results.

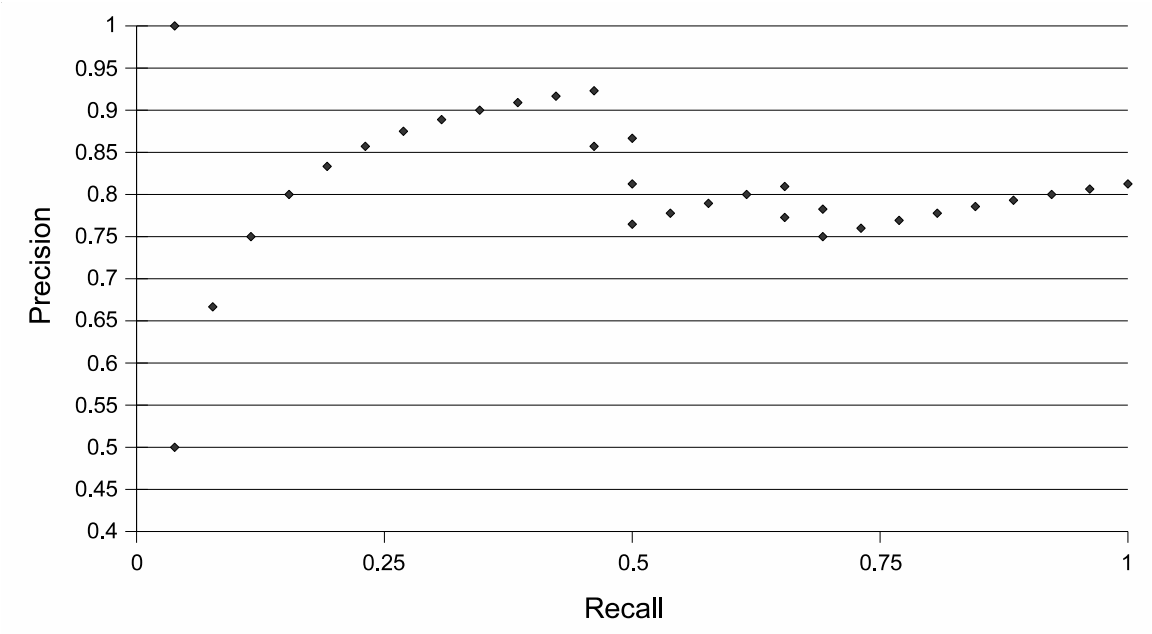


Figure 9: Results from the third query image.

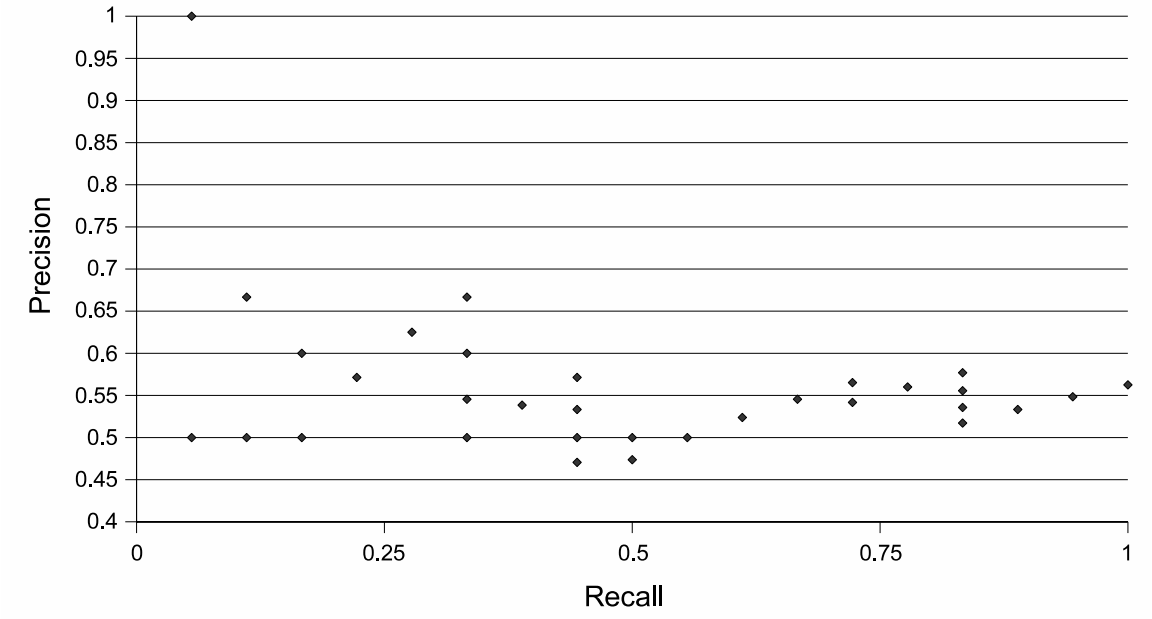


Figure 10: Results the fourth sample query image containing a sun spot.

a higher average precision than the image containing the sun spot. Part of the reason for this could be because there are more images in the database containing coronal loops than sun spots. Overall this IR system works and returns relevant images, but there is room for improvement.

Experimenting with different ranking algorithms for this application is the next step. While we are currently using a cosine similarity ordering that provides adequate results, there is obviously room for improvements. Now that this system is in place, it should be much simpler to add extra new features to the system to see how they effect the performance of the system.

CONCLUSIONS AND FUTURE WORK

Throughout this project we have created a dataset of images that can be used as a benchmark, extracted information from the images, built classifiers from the extracted information, and built a prototype IR system. The dataset, combined with the results from our prototype IR system gives researchers a baseline system to compare with future results. It is our hope that the benchmark dataset we have created will be used in the future by other researchers.

We have been able to process the solar images, taking them from a raw FITS format and represent them as a feature vector denoted as $\vec{i}$. This step involved cleaning up the images using SolarSoft, and then extracting attributes from the cleaned up images using OpenCV. We discovered that the extracted attributes of mean and standard deviation have a high merit for differentiating between various phenomena, and the attribute wavelength, that is present in the headers of the files, had the most merit.

The features we extract from the solar images, while simplistic, have proven useful in creating classifiers that can identify phenomena in solar images. Specifically, the classifier for the empty-sun is highly accurate with an AUC of 0.982. This is significant because this classifier can be used to discard portions of the image that contain no phenomena, reducing the number of comparisons that need to be made. The coronal loop classifier also has a high AUC of 0.976. While the classifiers for other labels do not have as high of accuracy, it is our hope that as the benchmark dataset matures and new techniques are used, more accurate classifiers can be produced for these labels.

The classifiers we created also provide the basis of a prototype IR system. The results from this system show that when searching for a coronal loop or sun spot, we

can expect more then half of the returned images to be relevant. In our results in the best case we can expect 83% of the returned images to be relevant. The images containing coronal loops also had a higher average precision than images containing the sun spots. Overall this IR system works, returns relevant images, and provides a baseline for other systems to compare.

Concerning future work, every component in the system could use improvements through more research. First of all new attributes should be investigated to add to the feature vector $\vec{i}$. Wavelet, Fourier transforms, and fractal signatures [50, 51] are all potential attributes that should be investigated. More image segmentations techniques can be investigated. These include techniques such as Normalized Cuts [52] and techniques that take advantage of domain specific knowledge, such as the curvature of the Sun in the case of solar repositories. OpenCV along with other image processing libraries should also be revisited to better understand what feature extracting methods they could provide.

Different classifiers can also be investigated. With the development of new features extracted from the images, the current classifiers might not work. Different classification algorithms will need to be investigated that would work better with different attributes contained within the feature vector $\vec{i}$. A different combination of the cascading classifiers can also be investigated. Currently we are using one binary classifier for each label. Future work could include creating an ensemble of classifiers for each label. One could also take the idea further, creating sub-label variations on the same type of phenomena. In this case a classifier could be created for each sub-label that could give a more precise idea of what was contained in an image.

Finally there are modifications to the IR system. Experimenting with different ranking algorithms for this application is the next step. While we are currently using a cosine similarity ordering that provides adequate results, it remains to be seen how

different ordering algorithms could affect the outcome.

While this project has produced a working prototype system, more research can be done on every component of the system. Now that a prototype system has been created, future research can focus on improving this system instead of implementing the components from scratch.

REFERENCES

[1] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, M. Press, Ed. 400 Sansome Street, Suite 400, San Francisco, CA 94111: Morgan Kaufmann, 2006.

[2] U. Fayyad and K. Irani, "On the handling of continuous-valued attributes in decision tree generation," *Machine Learning*, vol. 8, no. 1, pp. 87–102, 1992.

[3] U. M. Fayyad and K. Irani, "The attribute selection problem in decision tree generation," *Proceedings of the Tenth National Conference on Artificial Intelligence*, vol. 104, pp. 104–110, July 1992.

[4] I. H. Witten and E. Frank, *Data Mining:: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[5] P. Gibbons, L. Mummert, R. Sukthankar, M. Satyanarayanan, and L. Huston, "Just-In-Time Indexing for Interactive Data Exploration," Tech Report CMU-CS-07-120, Carnegie Mellon University, School of Computer Science, Tech. Rep., 2007.

[6] V. Zharkova, S. Ipson, A. Benkhalil, and S. Zharkov, "Feature recognition in solar images," *Artif. Intell. Rev.*, vol. 23, no. 3, pp. 209–266, 2005.

[7] S. Zharkov, V. V. Zharkova, and S. S. Ipson, "Statistical properties of sunspots in 1996â2004: I. detection, north-south asymmetry and area distribution," *Solar Physics*, vol. V228, no. 1, pp. 377–397, 2005. [Online]. Available: http://dx.doi.org/10.1007/s11207-005-5005-7

[8] M. Turmon, J. Pap, and S. Mukhtar, "Statistical pattern recognition for labeling solar active regions: application to SOHO/MDI imagery," *Astrophysical Journal*, vol. 568, no. 1 part 1, pp. 396–407, 2002.

[9] T. D. de Wit, "Fast segmentation of solar extreme ultraviolet images," *Solar Physics*, vol. 239, no. 1, pp. 519–530, 2006.

[10] V. V. Zharkova and V. Schetinin, "Filament recognition in solar images with the neural network technique," *Solar Physics*, vol. V228, no. 1, pp. 137–148, 2005. [Online]. Available: http://dx.doi.org/10.1007/s11207-005-5622-1

[11] V. Delouille, J. D. Patoul, J. F. Hochedez, L. Jacques, and J. P. Antoine, "Wavelet spectrum analysis of eit/soho images," *Solar Physics*, vol. V228, no. 1, pp. 301–321, 2005. [Online]. Available: http://dx.doi.org/10.1007/s11207-005-5620-3

[12] J. Goldberger, H. Greenspan, and S. Gordon, "Unsupervised Image Clustering Using the Information Bottleneck Method," *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pp. 158–165, 2002.

[13] T. Deselaers, D. Keysers, and H. Ney, "Clustering visually similar images to improve image search engines," *Informatiktage 2003 der Gesellschaft fr Informatik, Bad Schussenried, Germany*, 2003.

[14] D. Hoiem, R. Sukthankar, H. Schneiderman, and L. Huston, "Object-Based Image Retrieval Using the Statistical Structure of Images," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. 490–497, June 2004.

[15] L. Huston, R. Sukthankar, D. Hoiem, and J. Zhang, "SnapFind: Brute Force Interactive Image Retrieval," *Proceedings of International Conference on Image Processing and Graphics*, vol. 00, pp. 154–159, 2004.

[16] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1816–1823 Vol. 2, 2005.

[17] F. Florea, A. Rogozan, A. Bensrhair, J. Dacher, and S. Darmoni, "Modality Categorization by Textual Annotations Interpretation in Medical Imaging," *Medical Informatics Europe (MIE 2005)*, pp. 1270–1275, 2005.

[18] F. Florea, C. Vertan, A. Rogozan, A. Bensrhair, and S. Darmoni, "Comparison of histogram-based feature sets for medical image modality categorization," *Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on*, vol. 1, pp. 47–50 Vol. 1, 2005.

[19] F. Florea, A. Rogozan, A. Bensrhair, and S. Darmoni, "Comparison of feature-selection and classification techniques for medical images modality categorization," INSA de ROUEN, Perception Systemes Information FRE CNRS 2645, INSA de Rouen, Tech. Rep. FFI no.01, 2004. [Online]. Available: http://asi.insa-rouen.fr/~fflorea/documents/rapport_intern_PSI_FFI.pdf

[20] L. Yang, R. Jin, R. Sukthankar, B. Zheng, L. Mummert, M. Satyanarayanan, M. Chen, and D. Jukic, "Learning distance metrics for interactive search-assisted diagnosis of mammograms," *Proceedings of SPIE*, vol. 6514, p. 65141H, 2007.

[21] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *J. Intell. Inf. Syst.*, vol. 3, no. 3-4, pp. 231–262, 1994.

[22] S. W. McIntosh and J. B. Gurman, "Nine Years Of Euv Bright Points," *Sol. Phys.*, vol. 228, pp. 285–299, May 2005.

[23] (2007, November) Example diamond applications. [Online]. Available: http://diamond.cs.cmu.edu/applications.html

[24] S. Shekhar and S. Chawla, *Spatial Databases: A Tour*. Upper Saddle River, New Jersey 07458: Prentice Hall, 2003.

[25] (2007, June) Fits overview. [Online]. Available: http://heasarc.gsfc.nasa.gov/docs/heasarc/fits_overview.html

[26] (2007, June) Definition of the flexible image transport system (fits). [Online]. Available: http://archive.stsci.edu/fits/fits_standard/

[27] (2007, Jume) The fits support office at nasa/gsfc. [Online]. Available: http://fits.gsfc.nasa.gov/

[28] (2007, June) Fits libraries guide. [Online]. Available: http://fits.gsfc.nasa.gov/fits_libraries.html

[29] (2007, June) Fv: The interactive fits file editor. [Online]. Available: http://heasarc.gsfc.nasa.gov/ftools/fv/

[30] (2007, June) Saoimage ds9: Astronomical data visualization application. [Online]. Available: http://hea-www.harvard.edu/RD/ds9/

[31] (2007, January) Solarsoft. [Online]. Available: http://www.lmsal.com/solarsoft/

[32] (2007, June) The idl computing environment for data visualization analysis from itt. [Online]. Available: http://www.ittvis.com/idl/

[33] (2007, July) Transition region and coronal explorer. [Online]. Available: http://trace.lmsal.com/

[34] L. Golub and J. M. Pasachoff, *Nearest Star The Surprising Science of Our Sun*. Cambridge, Massachusetts: Harvard University Press, 2001.

[35] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, A. Dowrkin, Ed. Upper Saddle River, New Jersey 07458: Prentice Hall, 2002. [Online]. Available: http://www.prenhall.com/gonzaalezwoods

[36] M. Boutell, J. Luo, and X. Shen, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, September 2004.

[37] M. Johnson and R. Cipolla, "Improved Image Annotation and Labelling through Multi-label Boosting," *Proceedings of 2005 British Machine Vision Conference (BMVC '05)*, vol. 5, 2005.

[38] T. Joachims, "Optimizing search engines using clickthrough data," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2002, pp. 133–142.

[39] T. Joachims and F. Radlinski, "Search engines that learn from implicit feedback," *Computer*, vol. 40, no. 8, pp. 34–40, August 2007. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4292009

[40] Y. Rui, T. Huang, and S. Chang, "Image retrieval: current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 4, pp. 39–62, 1999.

[41] R. Chora, T. Andrysiak, and M. Chora, "Integrated color, texture and shape information for content-based image retrieval," *Pattern Analysis Applications*, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10044-007-0071-0

[42] S. Gordon, H. Greenspan, and J. Goldberger, "Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations," 2003, pp. 370–377 vol.1.

[43] (2007, Feburary,March,April) Opencv library wiki. [Online]. Available: http://opencvlibrary.sourceforge.net/

[44] N. Japkowicz, "Learning from imbalanced data sets: a comparison of various strategies," 2000. [Online]. Available: citeseer.ist.psu.edu/japkowicz00learning.html

[45] (2008, April) Text retrieval conference (trec) homepage. [Online]. Available: http://trec.nist.gov/

[46] P. Zhang, J. Peng, and B. Buckles, "Learning optimal filter representation for texture classification," *icpr*, vol. 2, pp. 1138–1141, 2006.

[47] P. Bhattacharya, M. Rahman, and B. C. Desai, "Image representation and retrieval using support vector machine and fuzzy c-means clustering based semantical spaces," *ICPR*, vol. 2, pp. 1162–1168, 2006.

[48] C. Barbu, K. Zhang, J. Peng, and B. P. Buckles, "Boosting in classifier fusion vs. fusing boosted classifiers," in *IRI*, 2005, pp. 332–337.

[49] M. Hall and G. Holmes, "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining," *Benchmarking*, vol. 15, no. 6, pp. 1437–1447, 2003.

[50] M. Mehrubeoglu, N. Kehtarnavaz, G. Marquez, and L. Wang, "Characterization of skin lesion texture in diffuse reflectance spectroscopic images," *Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium*, pp. 146–150, 2000.

[51] A. Piantanelli, P. Maponi, L. Scalise, S. Serresi, A. Cialabrini, and A. Basso, "Fractal characterisation of boundary irregularity in skin pigmented lesions," in *Medical Biological Engineering Computing*, vol. 43, 2005, pp. 436–442.

[52] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 22, no. 8, pp. 888–905, 2000.

APPENDIX A

SAMPLE QUERY IMAGES

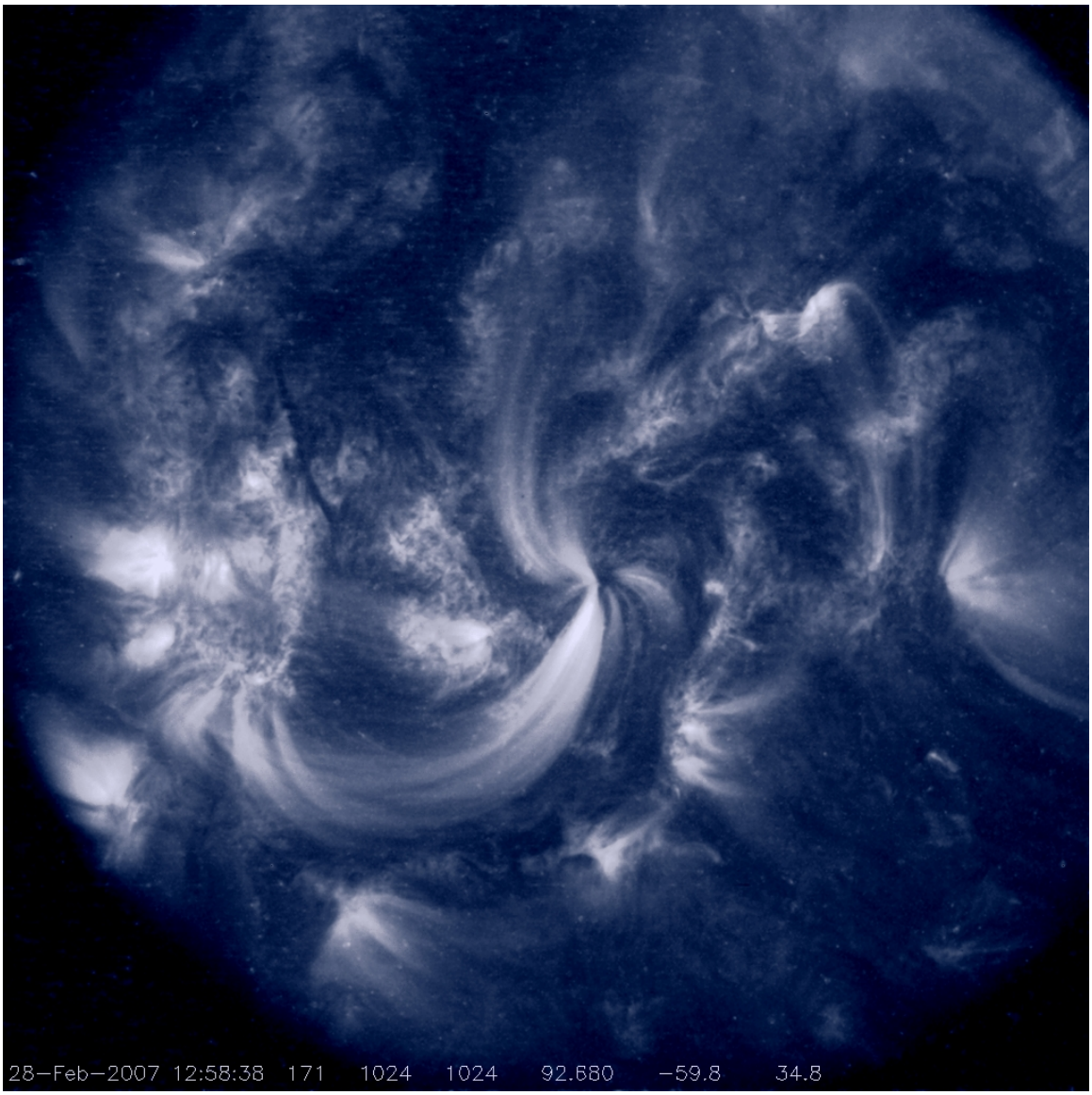


Figure 11: The first sample query image.

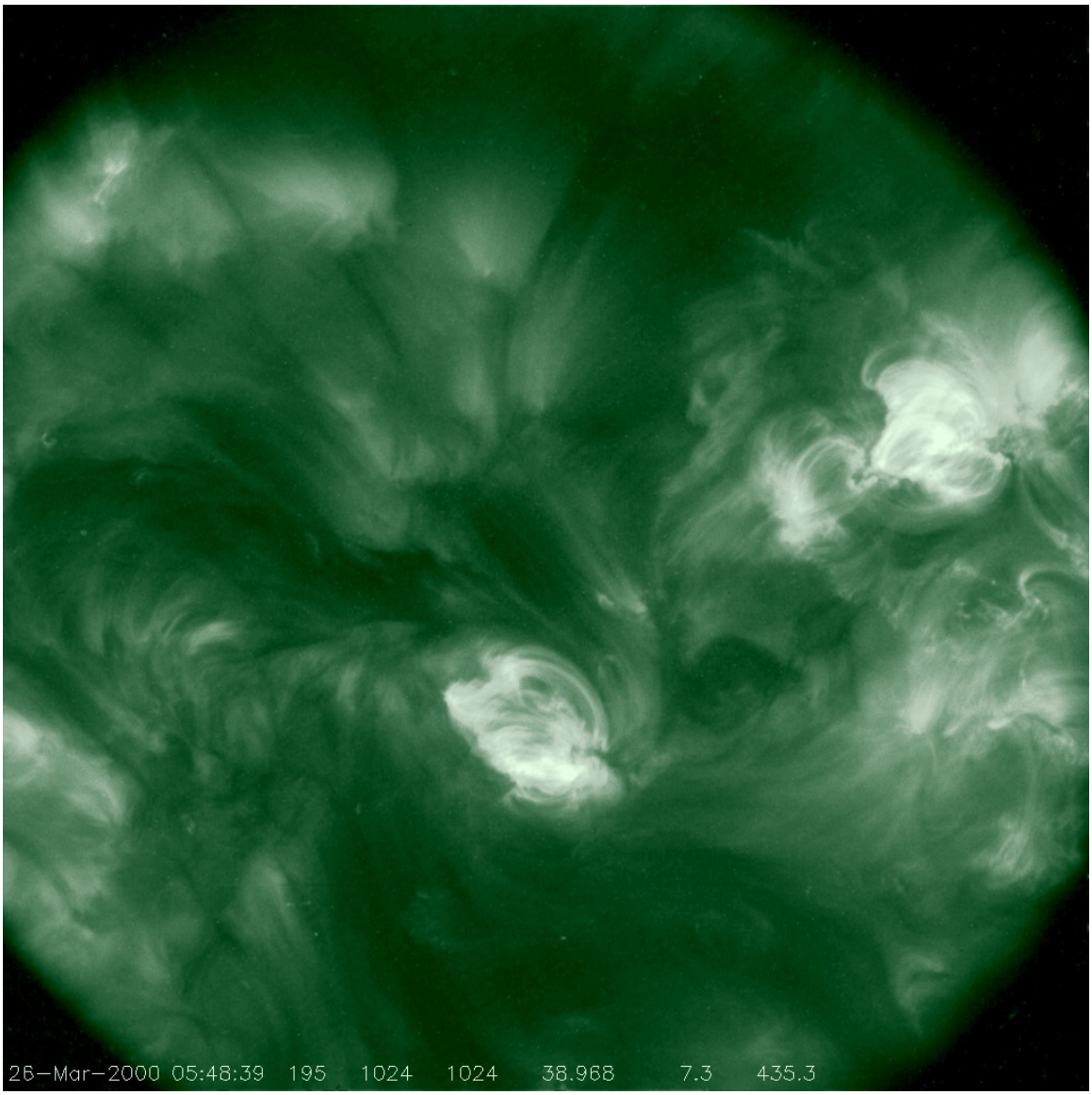


Figure 12: The second sample query image.

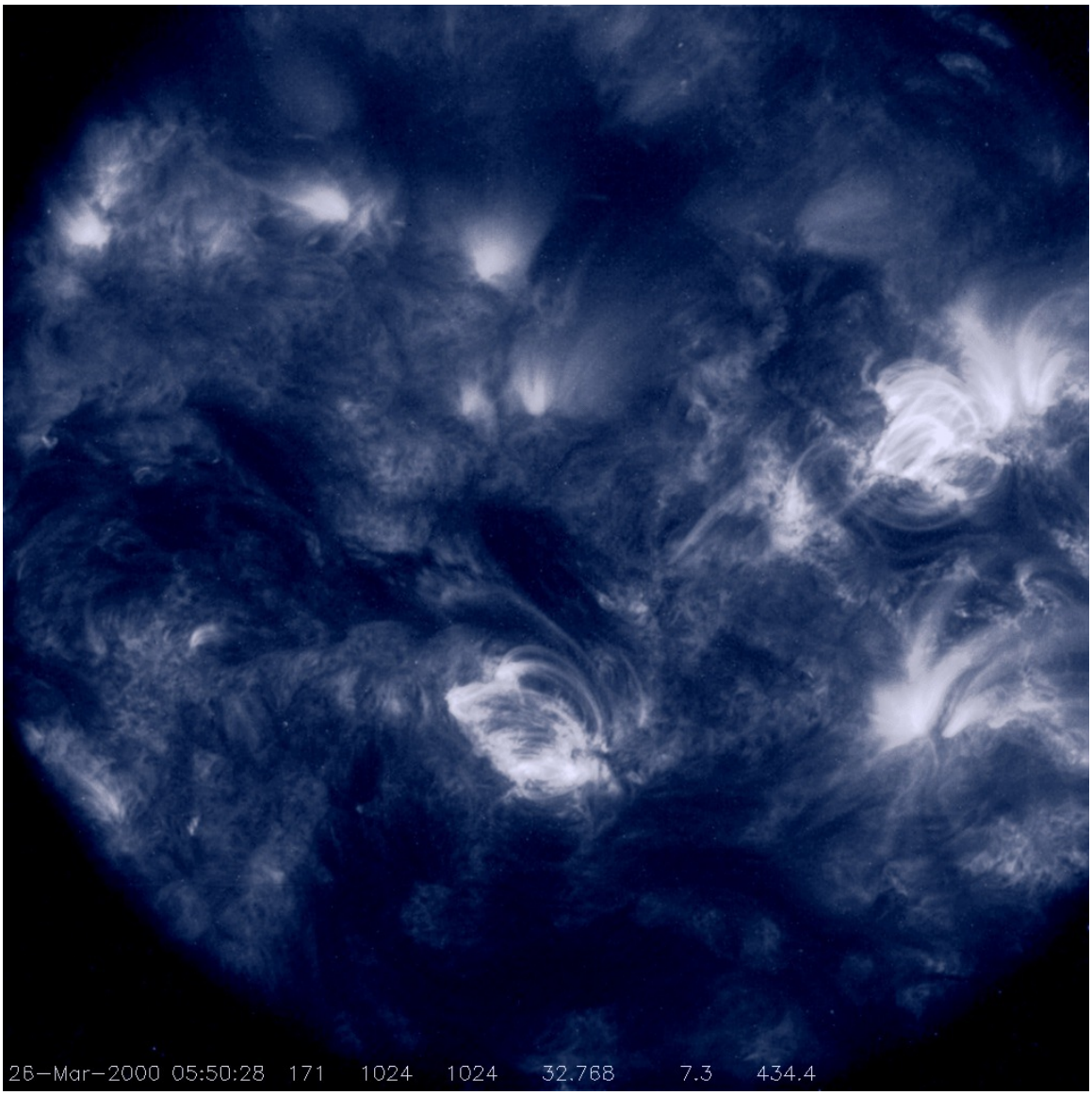


Figure 13: The third sample query image.

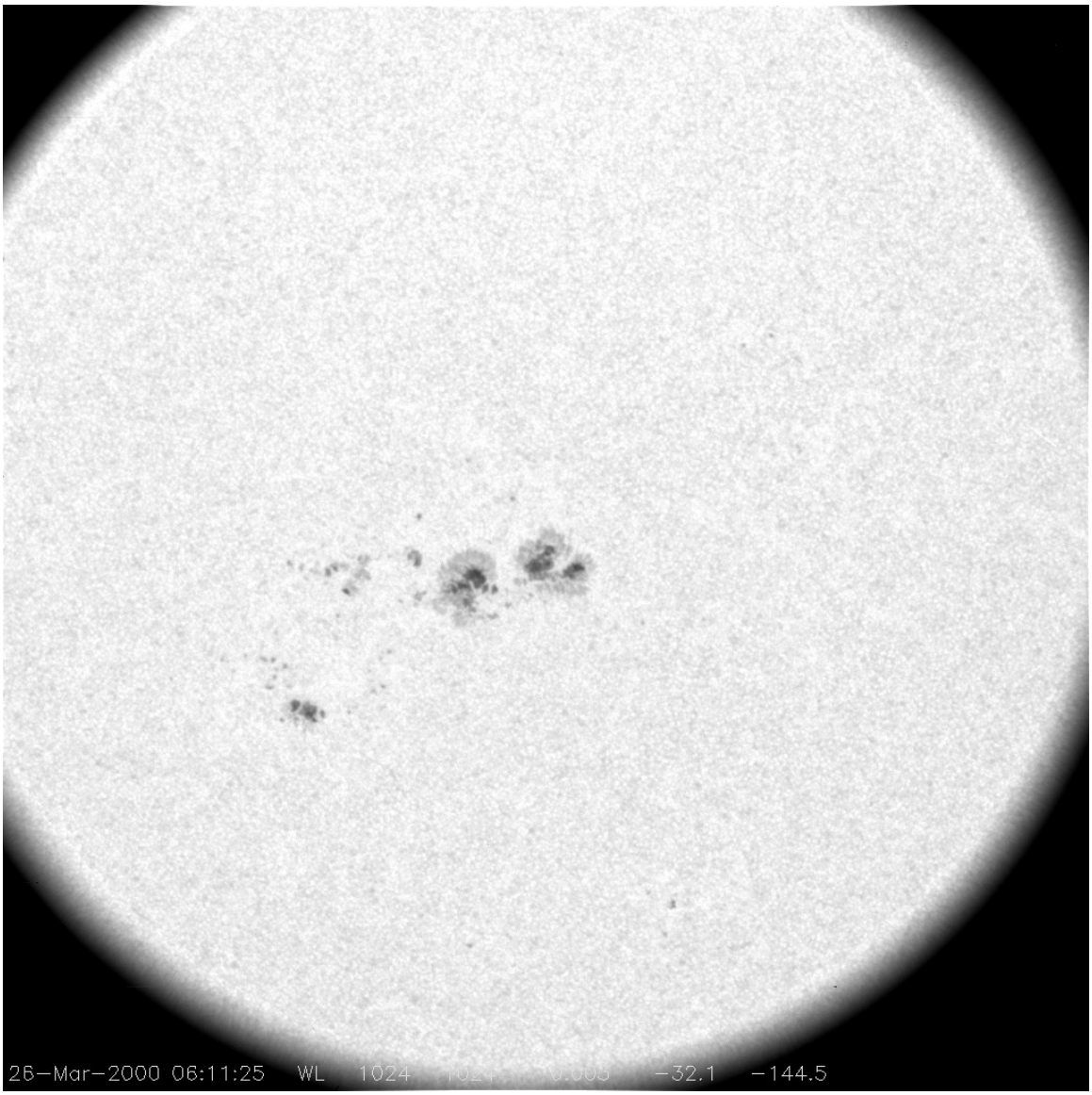


Figure 14: The fourth sample query image.