Design of Automatic Supernova Detection System

Li Jiang[†1†3] Hideyuki Kawashima[†2†3] Osamu Tatebe[†2†3]

This report describes the design of automatic supernova detection system for the cosmological HSC survey. This survey aims to use Hyper Suprime-Cam to shot space images and process analyzing tasks. One important target is to discovery supernovas. In order to get fast supernova detection speed, machine learning methods are used instead of human scanning to classify candidates into specific classes. We design the supernova detection system to fit requirements from cosmologists, supporting efficient data management and accessing, and high performance distributed data storage and processing as well. This report also shows experiments of supernova classification results with attempt of using deep learning, which is novel in this field and get interesting results.

1. Introduction

In Cosmology field, the detection of supernova is an important task. In recent years, by adopting machine learning techniques, cosmologists are expected to be released from heavy tasks of judging candidate objects by human observing, which is time-consuming. Some work [1, 2] attempts to use machine learning techniques in astronomy discoveries.

HSC Survey

Hyper Suprime-Cam(HSC) is a gigantic digital camera built by National Astronomical Observatory of Japan. From March, 2014, a 5-year long space survey project started. One major target of this survey is to discovery supernovas. Generally, from 1 night observation, there can be 20000 objects extracted which are possible to be supernova. Unfortunately, most of them are just noises and real supernovas are very rare. With requirement of real-time supernova detection, multiple machine learning solutions are under attempts and experiments in order to produce accuracy classification results within short period so that the follow-up observation can focus on interesting areas in space and confirm the discovery further.

SciDB

Nowadays, science and industry are growing increasingly data-intensive, efficient analysis of big data is getting more and more important. In many fields, data has multiple dimensions and does not fit in the table data model so well, leading a high cost in data access and analysis tasks. In order to efficiently store and analyze such multi-dimensional data, array database systems appeared, with array as basic data model instead of table. SciDB is a typical array database system [3, 4], designed to store and manipulate big multi-dimensional data. It is a columnar store with the basic data model as array instead of table. This array data model naturally fits in data schema very well in many fields, such as cosmological 2-D images, makes SciDB inherently support efficient storage and analysis over multi-dimensional data. On the other hand, SciDB has a design of shared nothing distributed storage architecture and can process queries in parallel. For storing a large array, SciDB divides it into small chunks and distribute these chunks across severs in a cluster, make it possible to compute queries in parallel for each chunk.

Classification

Classification is a supervised-learning technique, which means a classifier is trained from training dataset first. Then for a new object whose class label is unknown, we can use the classifier to predict its class. In the case of supernova detection, the classifier is trained with a dataset labeled by human scanners. Once the classifier is trained, it can be used to give predictions on whether a candidate is a supernova based on the knowledge it learnt from the training dataset.

This work is to design an automatic Supernova Detection System for the HSC survey in order to support efficient data management and reduce data accessing cost. We choose SciDB as the basic database based on this requirement. Another target is to integrate all the data processing tasks during the classification and provide a convenient and easy-using system.

The rest of this report is structured as following: Section 2 describes the motivation of designing this automatic supernova detection system; Section 3 describes details of the proposed system; section 4 shows some results of classification experiments; section 5 describes related work and section 6 is the summary of this report.

2. Motivation

This section describes the specific requirements for the supernova detection system. Since our cooperating researchers have already developed several functional programs which can analyze telescope image data and produce classification results, we need to show the motivation of designing a new supernova detection system. The current classification framework has several weak points that can be improved, and we discuss them in details as following.

2.1 Data Storage and Management

In the case of HSC, for 1 ‘shot’, it actually takes 104 CCDs. Here one CCD means a subarea frame taken of the space, and we refer it to as a frame for easy-understanding in the rest of this report. Then all these 104 frames are preprocessed by HSC pipeline and combined into a skymap which is a complete observation of HSC, which is shown in figure 1

In current framework, all these images are stored in files, one single file for each frame. For further processing tasks, data is

†1 Graduate School of Systems and Information Engineering, University of Tsukuba.
†2 Faculty of Engineering, Information and Systems, University of Tsukuba
†3 CREST JST CREST

loaded from files when required. It should be mentioned that as the survey moves on, more and more shots will be taken, and for each shot, 104 frame files are created. Eventually, the total images data will get very large.

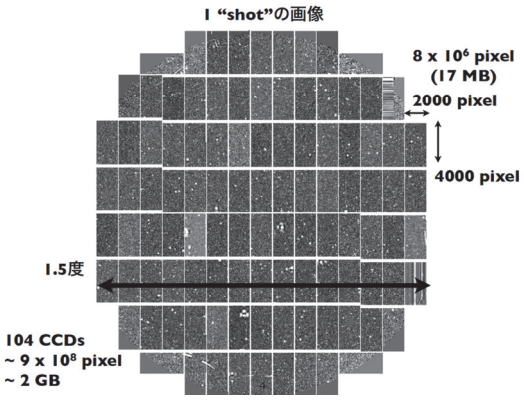

Figure 1 One complete observation of HSC telescope

This scheme as simply store data in files is actually not a well-managed storage solution. For analyze tasks processed by unit of frames, it is acceptable. But for analyze tasks that need to access customized areas (which is a real requirement from cosmologists), files may not work efficiently. When the targeting area contains more than one frame and even the border frames are only required partially, the data accessing cost gets higher. Another important requirement from cosmologists is that they wish to conduct image differencing and further processing of a same area but taken at two different time. It can be painful when accessing data to find out the correct files that should be loaded.

Therefore, a well-manage storage solution for big image data is required. It should be able to efficiently access data from any area of the skymap, based on any time the needed 'shot' is taken.

2.2 System Integration

The supernova detection framework requires several steps of data processing before the final classification. In current framework, different steps are implemented separately and they are not so well-integrated. In order to run a complete classification, multiple calls of different components are required.

The whole workflow can be divided into steps as image subtraction, object extraction, feature calculation and classification.

(1) Image Subtraction

Image Subtraction is the first step for supernova detection. It calculates the image differencing between 2 images, one is a reference image which is taken before, and another one is the new taken image. After the subtraction, supernova or other transient objects get much easier to be detected compared with the background.

State-of-art algorithm of image subtraction was proposed and optimized in work [5, 8]. Current image subtraction implementations are almost all based on this work. In this field, another research about acceleration of subtraction based on GPU named P-HOTPANTS [6] is quite interesting. It argued that the process of image subtraction has considerable computation part that can be parallel processed. Based on this work, we can also achieve parallel processing based on distributed data storage and improve the process of image subtraction.

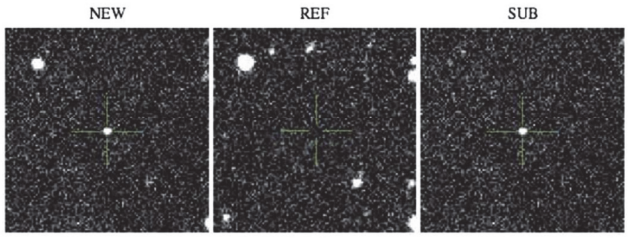

Figure 2 Image subtraction of new and reference image[2]

(2) Object Extraction

After subtraction, an extraction process is executed to produce interesting objects, which are candidates for further analyzed. Then these candidates are stored into a catalog database with center coordinators and other information.

(3) Feature Calculation

Feature is an important concept in machine learning. An object is represented as a set of features which are able to describe the object's attributes. Feature definition and selection is critical for machine learning methods to get good results. However, for specific tasks, feature engineering requires knowledge in the fields and in our case of supernova classification, the machine learning team is still working on figuring out the best features that can distinguish supernova from other objects.

After features are defined, for each candidate extracted in step 2, its features need to be computed, usually based on the sub-image around the candidate object.

(4) Classification

With the classifier already trained ahead, by using the features calculated in step 3, predictions on candidates' class are produced. Candidates that get class label as 'supernova' are gathered in the supernova candidate list and finally output as result of the workflow.

Obviously, it is more convenient to develop a system integrates all these processing steps and simply accepts input images and outputs possible supernova candidates. Another disadvantage of current separated workflow is that the intermediate result images are required to be stored in temporary files and normally discarded after the whole process, while in an integrated system, it's easier to store, manage and access such intermediate results in case of further reuse.

2.3 Parallel Processing

After analyze the detailed processing steps, we found that most image processing tasks during the classification workflow can be divided into sub-tasks. For example, for 2 images, divide them each equally into 4 sub-images and conduct subtraction for

every pair of sub-images, then combine the 4 result images together will produce the same result as subtraction of the 2 original images.

Therefore, most image processing tasks can be executed in parallel and performance can be improved in this way.

3. Proposed System Design

This section describes detailed design of the supernova detection system. The system is designed to fit the requirements discussed in Section 2.

3.1 Storage: Array Database System

In order to effectively store and manage the big images, we decide to use a distributed array database system: SciDB.

SciDB is designed to efficiently deal with multi-dimensional data, such as 2-D images. Its array data model makes it able to quickly locate a sub-array and accessing the requested data in a multi-dimensional coordinate system.

To storage one complete shot of the HSC survey introduced in Section 2.1, we plan to store it in 2 ways. First, similar with current file storage, for each frame we create a 2-D array to store the image, with array name clearly recording the area and taken time of the frame. These small arrays are used for tasks focus on single frame and provide fast accessing. Besides this, we will store another huge 3-D overall array, with the first two dimensions represents the coordinators in the space and the last dimension represent the time series. In this array, for each available time point, instead of separated 104 small frames, the whole skymap is stored as an integrated image. Figure 3 shows this overall array.

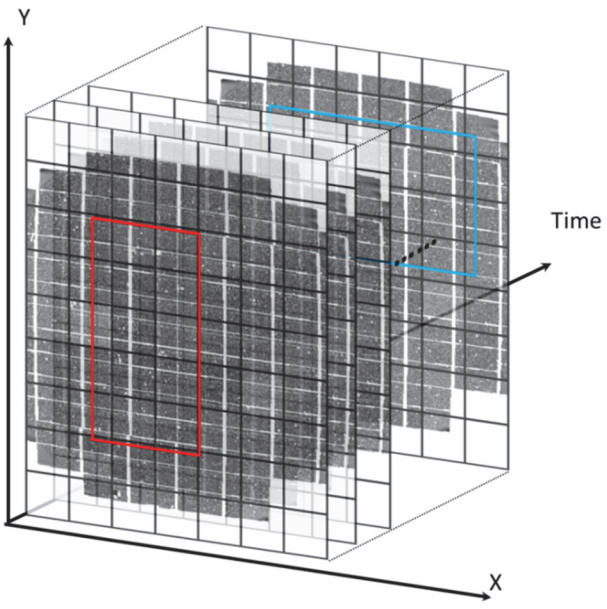

Figure 3 3-dimensional overall array storage and data access

Instead of rigid frame by frame data accessing in file storage, in this way, data can be accessed efficiently with any size of area, and based on any time the shot was taken. As shown in figure 3, the red and blue rectangles represent 2 data accesses targeting with customized areas and different shot taken time. This feature is very useful for supernova historical tracking, confirmation and many other cosmological analyzing tasks.

Another advantage of SciDB is its ability of managing big size of data. SciDB can be built as a shared-nothing distributed cluster. When dealing with a huge array, it divides the whole array into small chunks and distributes these chunks within the cluster. When get data access and process tasks, each node only handles the sub-tasks required for the data stored in that node.

3.2 Classification Process: Integrated inside SciDB

Because most of the supernova classification process involved image processing, it can improve the performance to implement these tasks into SciDB and execute them within SciDB, taking advantage of its multi-dimensional feature and natively supported distributed parallel processing. Also an integrated system can provide more convenient usage and reduce the cost of intermediate temporary file creation and accessing.

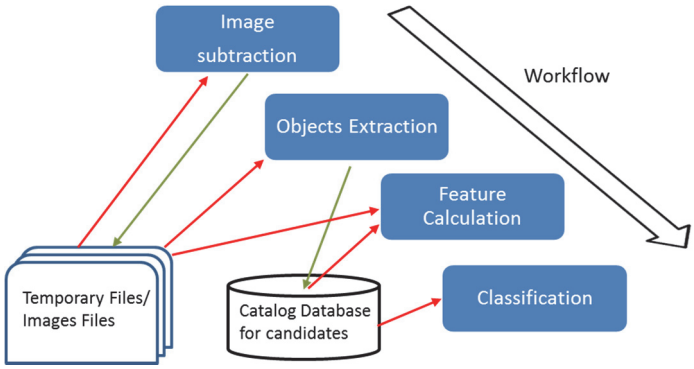

Figure 4 Current supernova detection workflow

Figure 4 above shows the current workflow of supernova detection with separated 4 steps. The red lines represent for data access and green lines represent for data storage.

Compared with this separated workflow, figure 5 briefly shows the integrated supernova detection system designed with a distributed storage array Database – SciDB. By this design, the whole large skymaps are stored in distribute, so that any sub-area from the skymap can be efficiently accessed for analyze usages.

Another benefit of distribute storage is that SciDB support parallel processing by chunk, which is the unit of array distribute storage. This means, if we can implement all processing steps inside SciDB and execute them chunk by chunk, the performance can be improved as parallel processing is achieved.

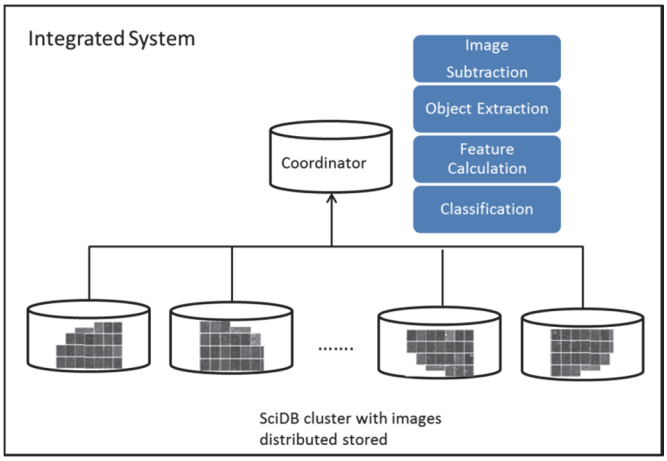

Figure 5 Designed Integrated System with SciDB as database

However, in order to implement all the steps into SciDB, we need to study the detailed algorithms of image subtraction, object extraction and feature calculation from our cooperating cosmologists. Unfortunately, in current stage, information of most processing algorithms is not well-provided so far.

The HSC survey project is still in its early stage and we have only been provided with the detailed image subtraction algorithm. The implementation of it into SciDB is still undergoing.

4. Classification Stage

Classification stage is the final step of the system to produce supernova candidates. In the system, the classifiers trained ahead are already prepared to process class prediction on new coming objects. Generally, for most classification methods, the training stage is time-consuming but the prediction (classification) stage is always much faster and can be treated as real-time response.

The selection of specific suitable machine learning method for supernova detection is very important. Thanks to the cosmology team's hard work on scanning and labeling the data, a set of valuable labeled objects is provided in February which can be used for classification training and testing.

Several different classification methods are tested by different teams. From their results, random forest seems to be the most stable and accurate one. To verify this result, we work out our own random forest implementation and the experiment result is very similar to the other teams. On purpose of seeking other machine learning methods that work well in supernova detection, we also tested convolutional neural network which is a type of deep learning with the same data and exactly the same experiment process as others so that the results are comparable.

4.1 Random Forest

Random forest is learning method for classification and regression. It constructs a forest of multiple decision trees at training stage and predicts the final result by voting among all the trees. By selection of a random subset of training data for each decision tree and selection of a random subset of features in each decision node, random forests correct the overfitting problem of single decision trees and turns out to be one of the top machine learning methods.

4.2 Deep Learning – Convolutional Neural Network

Unlike most machine learning methods represent objects with features and process analysis based on feature values, deep learning processes on the raw data and automatically figure out the most suitable features itself.

Convolutional Neural Network (CNN) is a type of deep learning which is very powerful in image recognition field. It is comprised of one or more convolutional layers and followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image. This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units.[7]

To the best of our knowledge, there are no deep learning attempts in supernova detection field before. Considering the candidates can be represented as small images with object in the center, it's exactly an image recognition task which convolutional neural network is good at.

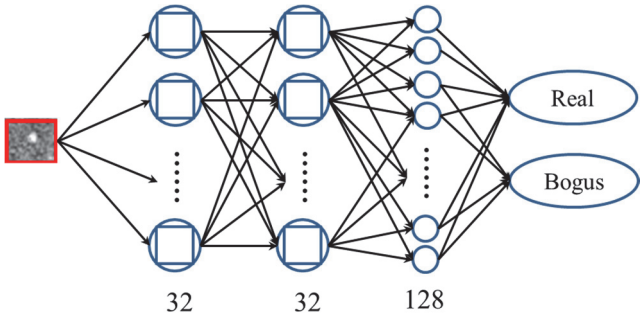

Figure 6 Our convolutional neural network model used in the Experiment

Figure 6 above show the detailed model of our convolutional neural network test. The first 2 layers are convolutional layers and followed by 2 dense layers and finally project into 2 classes.

4.3 Experiment Results

Current machine learning experiments are focusing on the first stage of real/bogus classification. Here real(positive) objects stand for candidates for possible varying stars, supernovas, asteroids or other transient objects; bogus(negative) objects stand for meaningless noises or objects that are not interesting.

The tested dataset contains 15810 objects, including 61 real objects and 15749 bogus objects. In the experiment, the whole dataset is evenly divided into 5 stacks, 4 stacks are combined as training dataset and the rest one stack is used for test to check the classification accuracy.

In machine learning, FNR and FPR are 2 important metrics to evaluate how accuracy is a method. FNR is short for 'False Negative Ratio', represents the ratio of real objects misclassified

as bogus ones. FPR, short for False Positive Ratio, represents the ratio of bogus ones misclassified as real ones.

In order to compare two classifiers, a common strategy is to adjust threshold so that both methods get similar FNR and compare the FPR, or in reverse. In the following tables we show our convolutional neural network results comparing with random forest which shows the best result in the experiment so far.

Table 1 Accuracy Comparison (FPR fixed around 0.01)

	CNN	Random Forest (23 features)	Random Forest (6 features)
FPR	0.011	0.010	0.010
FNR	0.325	**0.278**	0.328

From table 1, we can see that when modify threshold to fix FPR around 1%, random forest with 23 features gets the best accuracy (FNR), but the convolutional network also produced comparable results.

Table 2 Accuracy Comparison (FNR fixed around 0.1)

	CNN	Random Forest (23 features)	Random Forests (6 features)
FPR	**0.055**	0.091	0.133
FNR	0.114	0.149	0.132

In the case of fixing FNR around 0.1, convolutional network gets the best FPR.

Because current dataset is still small and lacking of real objects, result of the classification methods are all lack of accuracy. Still, based on the experiment with current available data, the behavior of random forest is the most stable and accurate. However, from our experiment of convolutional network, it also shows comparable results. Besides that, it has the advantage of no requirement of feature engineering, which can save lots of effort and time cost on feature selection and tuning.

At this point, we think convolutional neural network can be a good option for the supernova detection task. We plan to execute more experiments with more data to check whether it can actually work well in this cosmology task.

4.4 Results Integration of Multiple Classification Methods

Although random forest and CNN has the most accurate results so far, other methods are also potential options and may produce better predictions on specific situations.

Therefore, instead of selecting a single final method for classification step in the system, we decide to integrate multiple classification results. Because the predictions time are always very short, it won't make a big difference predicting candidate class based on one single classifier or based on multiple classifiers.

By integrate different methods' classification results, we can gather the supernova candidates more completely and combine the strength of all methods. One solution is that as long as one method judges a candidate as supernova, this candidate is added to the final results. Although this can prevent most supernova candidate from mislabeled, it will also increase the false positive rate, meaning too much bogus ones in the final list as the result of loose standard and increase the human confirm workload. Another solution is to vote with all the methods. Those methods with better accuracy in the experiment can be assigned with heavier weight in voting. In this way multiple methods make contribution in the final decision making and too low standard is avoided. Related Works

Some work exists of using machine learning methods in astronomical discovery [1, 2]. They describe more details on the data processing strategy and feature selection in the cosmology field. About data efficient storage and management, parallel task processing which this report focus on, these works barely discussed.

About image subtraction, the state-of-art algorithm is proposed and optimized in work [5, 8]. Current image subtraction implementations are almost all based on this work. In this field, another research is P-HOTPANTS, an acceleration of subtraction based on GPU [6]. Although the main ideas to improve performance are similar between ours and this work which is to process the image subtraction in parallel, our method is to achieve distributed data storage and processing by a cluster of servers instead of GPU acceleration.

5. Summary

This report introduces our design of the supernova detection system which aims to automatically detect supernovas with input as telescope images. Our design supports effective big image data storage and management, as well as efficient data processing tasks required for classification in parallel. The design is exactly based on real requirements from our cooperating cosmologists.

On the other hand, an attempt of deep learning in supernova detection task is shown in the report. Not like other classification methods, deep learning doesn't require feature engineering and directly train and classify over image data, which saves lots of effects on feature selection and tuning. From the experiment, it shows comparable or even better accuracy. Therefore, deep learning is a potential nice option for the task of automatic supernova detection.

Acknowledgments

This work is partially supported by JST CREST "System Software for Post Petascale Data Intenseive Science" and JST CREST "Extreme Big Data (EBD) Next Generation Big Data Infrastructure Technologies Towards Yottabyte/Year". It is also supported by JSPS KAKENHI Grant Numer 25280043HA and JST CREST "Statistical Computational Comsmology with Big Astronomical Imaging Data".

Reference

1) Henrik Brink, J.W.Richards etc: Using machine learning for discovery in synoptic survey imaging data, MNRAS 435, 1047-1060, Advance Access publication 2013

2) J.S.Bloom, J.W.Richards, P.E.Nugent etc: Automating Discovery and Classification of Transients and Variable Stars in the Synoptic Survey Era, Publications of the Astronomical Society of the Pacific, 124:1175-1196, 2012
3) P. Cudre-Mauroux, H. Kimura, K.-T. Lim etc: A Demonstration of SciDB: A Science-Oriented DBMS, VLDB'09 Volume 2, Number 1, 1534-1537, Lyon, France, 2009
4) Paul G. Brown: Overview of SciDB, Large Scale Array Storage, Processing and Analysis, SIGMOD conference, 2010
5) C. Alard: Image Subtraction Using A Space-varying Kernel, Astronomy & Astrophysics Supplement Series, June 2000
6) Yan Zhao, Qiong Luo: Accelerating Astronomical Image Subtraction on Heterogeneous Processors, IEEE 9th International Conference on eScience, 2013.
7) http://ufldl.stanford.edu/tutorial/ Stanford online UFLDL Tutorial
8) C. Alard, RH. Lupton:[A Method for Optimal Image Subtraction, The Astrophysical Journal, August, 1998