

A Machine Learning Based Framework for Verification and Validation of Massive Scale Image Data

Junhua Ding, *Member, IEEE*, Xin-Hua Hu, and Venkat Gudivada, *Member, IEEE*

Abstract—Big data validation and system verification are crucial for ensuring the quality of big data applications. However, a rigorous technique for such tasks is yet to emerge. During the past decade, we have developed a big data system called CMA for investigating the classification of biological cells based on cell morphology that is captured in diffraction images. CMA includes a group of scientific software tools, machine learning algorithms, and a large scale cell image repository. We have also developed a framework for rigorous validation of the massive scale image data and verification of both the software systems and machine learning algorithms. Different machine learning algorithms integrated with image processing techniques were used to automate the selection and validation of the massive scale image data in CMA. An experiment based technique guided by a feature selection algorithm was introduced in the framework to select optimal machine learning features. An iterative metamorphic testing approach is applied for testing the scientific software. Due to the non-testable characteristic of the scientific software, a machine learning approach is introduced for developing test oracles iteratively to ensure the adequacy of the test coverage criteria. Performance of the machine learning algorithms is evaluated with the stratified N-fold cross validation and confusion matrix. We describe the design of the proposed framework with CMA as the case study. The effectiveness of the framework is demonstrated through verifying and validating the data set, software systems and algorithms in CMA.

Index Terms—Big Data, Diffraction Image, Machine Learning, Deep Learning, Metamorphic Testing.

I. INTRODUCTION

VOLUME, velocity, variety, and value are the four characteristics that differentiate Big Data from other data [1]. Volume and velocity refer to the unprecedented amount of data and the speed of its generation. Big Data is complex and heterogeneous (variety). To extract value from the data, special tools and techniques are needed. New algorithms, scalable and high performance processing infrastructure, and analytics tools have been developed to support big data research. For example, deep learning algorithms have been widely adopted for analyzing big data [2]. Hadoop provides a scalable and high performance infrastructure for running big data [3], and NoSQL databases are used for storing and retrieving big data [4]. To ensure reliability and high availability, big data applications

and infrastructure have to be validated and verified. However, the four characters of big data create new challenges for the validation and verification tasks. For example, data selection and validation are critical to the effectiveness and performance of big data analysis, but large volume and varieties of big data create a grand challenge for the selection and validation of big data. Existing work has shown that abnormal data existing in datasets could substantially decrease the accuracy of data analysis [5].

Many data analytics tools are complex and are difficult to be tested due to the absence of test oracles. Other approaches for verifying complex software systems are either impractical or infeasible. The machine learning algorithms used for processing big data are also difficult to be validated given the volume of data and unknown expected results. Although there are significant work on the quality assurance of big data, verification and validation of machine learning algorithms and "non-testable" scientific software, few work has been done on systematic validation and verification of a big data system as a whole. The focus of research presented in this paper is on the validation and automated selection of big data as well as verification and validation of analytics software and algorithms. To achieve the best data analysis performance, feature representation, feature extraction and feature selection for machine learning are also discussed. The verification and validation framework proposed in this paper is illustrated in Fig. 1, which includes tasks in three layers – the foundation layer is the technique for automated selection and validation of big data, the middle layer is an approach for verification and validation of the machine learning algorithms including feature representation, extraction and optimization, and the top layer is an approach for testing domain modeling systems, data analytics tools and applications. At the system level, a big data system can be verified and validated using regular software verification and validation techniques. However, regular system verification and validation approaches are not good enough for ensuring the quality of the essential components in a big data system: big data, machine learning algorithms and "non-testable" software components. The framework covers the essential verification and validation tasks needed for any big data application, and the techniques and tools proposed in this paper can be easily extended for verification and validation of other big data systems.

We demonstrate our approach on the classification of the diffraction images of biology cells. The 3D morphological features of a cell captured in the diffraction image can be used for

J. Ding is with the Department of Computer Science, East Carolina University, Greenville, NC, 27858 USA e-mail: dingj@ecu.edu.

X. Hu is with the Department of Physics, East Carolina University, Greenville, NC, 27858 USA e-mail: hux@ecu.edu.

Venkat Gudivada is with the Department of Computer Science, East Carolina University, Greenville, NC, 27858 USA e-mail: gudivadav15@ecu.edu.

Manuscript received April 15, 2016; revised December 2, 2016.

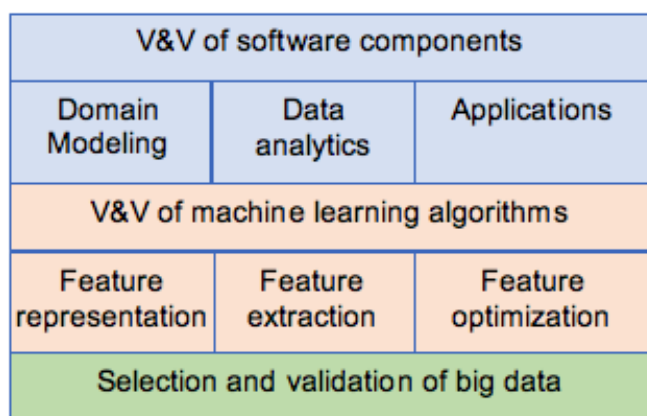


Fig. 1. A schema of V&V of Big Data Systems

accurately classifying cell types. Cells are basic elements of life. They possess highly varied and convoluted 3D structures of intracellular organelles to sustain their phenotypic variations and functions. Cell assay and classification are central to many branches of biology and life science research. While genetic and molecular assay methods are widely used, morphology assay is more suitable for investigating cellular functions at single-cell level. Significant progress has been made over the last few decades on fluorescent-based non-coherent imaging of single cells. Such techniques are used in immunochemistry for the study of molecular pathways and phenotypes and morphological assessment. However, microscopy based non-coherent image data is labor-intensive and time-consuming to analyze because they are 2D projections of the 3D morphology with objects too complex for automated segmentation in nearly all cases. For example, despite the availability of various open-source software systems for pixel operations, much of object analysis of cell image data relies heavily on manual interpretation [6]. 3D cell morphology provides rich information about cells that is essential for cell analysis and classification. Co-authors Ding and Hu have been studying cell morphology assay and classification for over a decade and developed a big data system called Cell Morphology Assay (CMA). This system is used for modeling and analyzing 3-dimensional (3D) cell morphology and to identify and extract morphology patterns from diffraction images of biological cells. These patterns can be viewed as morphology fingerprints and are defined based on the correlations between the 3D morphology and diffraction patterns of light scattered by cells of different phenotypes.

Diffraction images of single cells are acquired using a polarization diffraction image flow cytometer (p-DIFC), which was invented and developed by co-author Hu to quantify and profile 3D morphology [7]. CMA tools can rapidly analyze large amount of diffraction images and obtain texture parameters in real-time. Machine learning algorithms such as Support Vector Machine (SVM) [8] and texture features are used to optimize and identify a set of parameters (i.e., morphology fingerprints) to perform cell assay directly on diffraction images. CMA is innovative in that it provides a means for rapid assay of single cells without the need to stain them with fluorescent reagents.

It also provides researchers a significant source of big data and tools to conduct research and develop big data applications. CMA adopts big data techniques to implement data management, analysis, discovery, applications and sharing into the development of morphology based cell assay tools. It includes a group of scientific software tools for processing, analyzing and producing image data, machine learning algorithms for feature selection and cell classifications, and a database for managing the data.

The verification and validation framework of CMA supports the validation of the image data, evaluation of the effectiveness of the machine learning algorithms, and testing of the scientific software. A large number of diffraction images comprise the image database. The validation of the image data is implemented with a machine learning based image processing method for automatically selecting and classifying images. The evaluation of machine learning algorithms consists of two steps. The first step involves selecting optimized features to achieve best performance and effectiveness with machine learning algorithms. Cross validation of the machine learning results is done in the second step. Different feature selection approaches are used for cross checking the selected features and stratified N-Fold Cross Validation (NFCV) is applied for validating the feature selection results. Testing of scientific software is conducted with an iterative metamorphic testing technique [9], which is a metamorphic testing extended with iterative development of test oracles [10]. One of the major components of CMA is a group of scientific software, which is the software with a large computational component for supporting scientific investigation and decision making [11]. For example, 3D structure reconstruction software and light scattering modeling software in CMA are two such pieces of software. Many scientific software systems are non-testable because of the absence of test oracles [9] [11]. Metamorphic testing [9] [12] as a novel software testing technique is a promising approach for solving oracle problems. It creates tests according to metamorphic relationship (MR) and verifies the predictable relations among the outputs of the related tests. However, the application of metamorphic testing to large scale of scientific software is rare because the identification of MRs for adequately testing complex scientific software is infeasible [11]. This paper introduces an iterative approach for developing MRs, where MRs are iteratively refined with reference to the results of analyzing test execution and evaluation.

Although big data has become an important area of research recently, systematic work on quality assurance of big data is rare in literature. The framework introduced in this paper offers a comprehensive solution for ensuring the quality of big data. The framework is illustrated through verification and validation of CMA components. This case study also demonstrates the effectiveness of the proposed framework. The framework is extensible and is easy to adapt to big data systems.

The rest of this paper is organized as follows: Section 2 describes big data system CMA. Section 3 introduces the feature selection and validation for machine learning algorithms. Section 4 discusses the selection and validation of image data in CMA. Section 5 explains the testing of scientific software

in CMA. Section 6 describes the related work and Section 7 concludes the paper.

II. MASSIVE SCALE IMAGE DATA SYSTEM CMA

Like many other big data systems, CMA includes a big data repository, a group of software components for processing and analyzing the big data, and a set of data analytics algorithms. In this section, we discuss the architecture of CMA, the software components and algorithms.

A. The Architecture of CMA

The architecture and working flow of CMA is shown in Fig. 2. CMA includes four components: a database, a software component for studying 3D morphology of cells, a software component for finding *morphology fingerprints* from diffraction images of cells, and a software system for cell classification study. The morphology fingerprints are defined as a set of optimal features extracted from diffraction images for the classification. The foundation of CMA is the database, the core of CMA is a set of data analytics algorithms and image processing algorithms, and the business of CMA is the software components that are used for finding the morphology fingerprints for quick and accurate classification of cell types based on their diffraction images. The textual pattern of a diffraction image is used for classifying diffraction images using machine learning techniques. A diffraction image captures the unique 3D morphology information of each type of cells. Therefore, textual patterns extracted from diffraction images are able to classify cell types. However, how to define the textual patterns, how these patterns are correlated to 3D morphology of a cell, and how to find the optimal textual pattern parameters for the classification are unknown. CMA is designed to answer these questions. First, we model the light scattering of a cell based on its 3D morphology parameters using a scientific software. The modeling result of the light scattering of a cell is able to be converted into a diffraction image, and the correlation between the 3D morphology parameters and the textual pattern of the diffraction image can be built through an experiment study, which systematically changes the values of the 3D parameters to see the corresponding changes of the textual pattern in the diffraction image. In order to produce the 3D morphology parameters of a cell, a stack of confocal image sections are taken for the cell using a confocal imaging instrument. Then the confocal image sections are reconstructed for the 3D structure of the cell, and each cell component in the reconstructed 3D structure is assigned with a refractive index value. The 3D parameters are a 3D structure with assigned refractive index for every cell component. The morphology fingerprint study will produce a set of optimal parameters that can be used for classifying cells based on diffraction images. In order to confirm and refine the selected morphology fingerprints, they are used for classifying measured diffraction images using machine learning algorithms. Depending on the machine learning algorithm selected for the classification, the morphology fingerprints are first selected and optimized for better accuracy and performance. Then the morphology fingerprints are used for classifying different types of cells

based on the diffraction images. In this paper, the diffraction image that is taken using an imaging instrument is called measured diffraction images, and the diffraction image that is calculated using the modeling software is called calculated diffraction image.

B. Database

The database system is developed on MongoDB [13] as database management system and MongoChef [14] as client application to support remote access via Internet. Considering the properties of each image to be stored in the database could be often changed depending on applications of the image, and the data are mainly serving for data analytics, NoSQL MongoDB is a prefer choice than regular SQL database. The image data storing in the database include three collections: measured diffraction images and their processing results, calculated diffraction images and their processing results, and 3D reconstructed structures and morphology parameters data and their corresponding confocal images. The measured diffraction images of cells are acquired using polarized diffraction flow cytometer (p-DIFC) [7], and the calculated diffraction images are obtained using a light scattering modeling tool called ADDA [15] [8]. ADDA is an implementation of Discrete Dipole Approximation (DDA) [15]. The confocal images of cells are taken using confocal microscopes and are used for rebuilding the 3D structure of the cell. The data processing results include 3D cell structure data, 3D cell parameters that are individual segmentation results of intracellular organelles in each confocal image section; calculated results from ADDA simulation; feature values of each diffraction image; experiment results of feature selection; training and test data sets for machine learning, labeled images for cell classifications; and other results. More than 600 thousand images and their related data processing results have been added to the database, and new data is added daily. The data in the database may include noise images. For example, if a blood sample contains non-cell particles or fractured cells, the diffraction images taken from the sample will include abnormal diffraction images. If the abnormal diffraction images are labeled as normal cells in the training set, the accuracy of the cell classification could be substantially decreased [5]. Therefore, identifying and filtering out the abnormal data is important for ensure the quality of machine learning. In this paper, two machine learning approaches are introduced for addressing the issue. One is Support Vector Machine (SVM) [16] based approach integrated with image processing algorithm for automatically identifying abnormal diffraction images and separating normal diffraction images from abnormal one, and the other one is a deep learning [17] based approach. Although different SVM kernel functions were tried in our experiments, only the linear kernel function produced the best results.

C. A High Speed GLCM Calculator

To enable quantitative characterization of textual patterns in the diffraction images, Grey Level Co-occurrence Matrix (GLCM) [18] [19] features are calculated for each image. Haralick proposed GLCM for describing computable textural

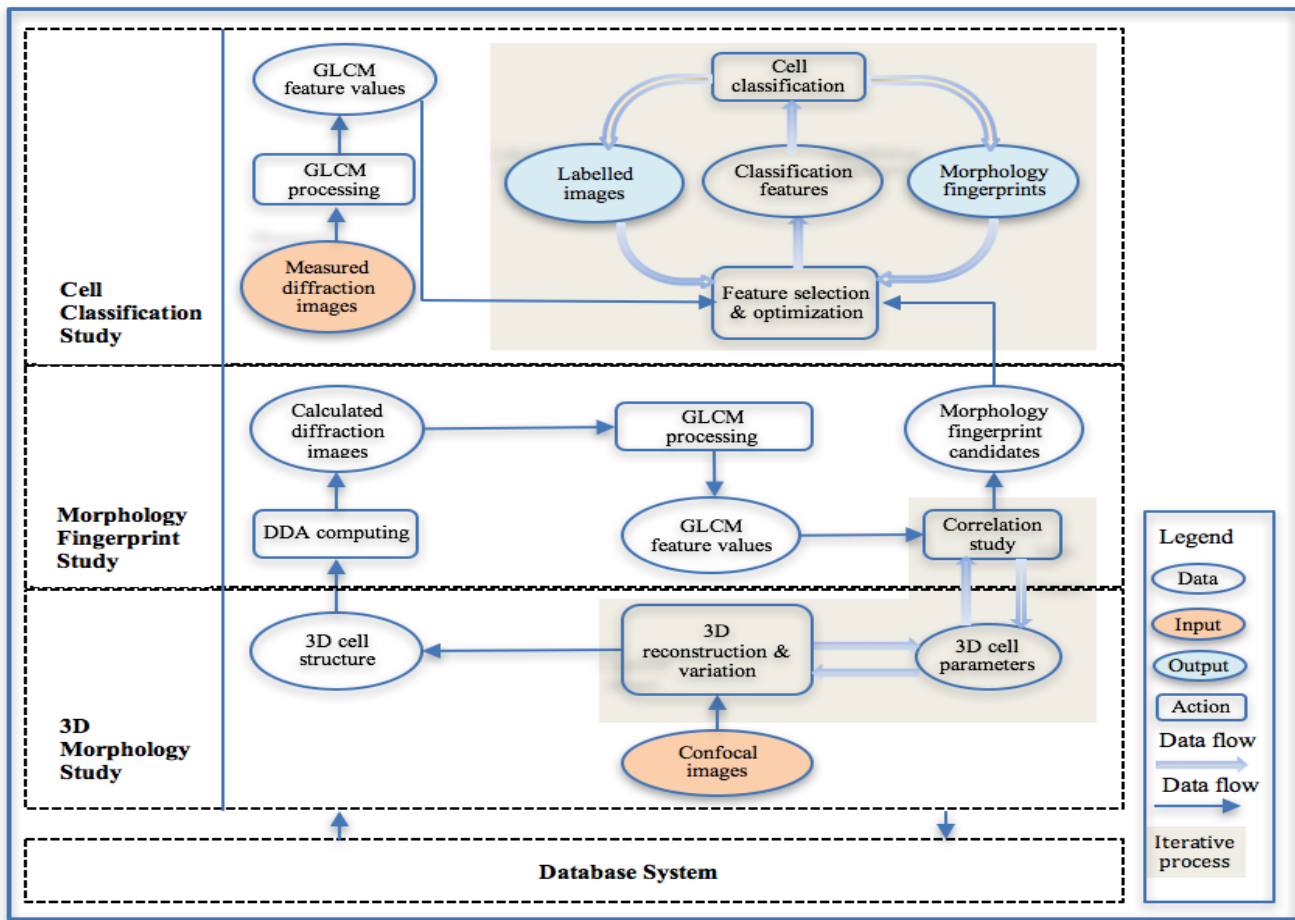


Fig. 2. An overall structure of CMA

features based on grey-tone spatial dependencies [20]. It defines how often different combinations of gray level pixels occur in an image for a given displacement/distance d in a particular angle θ . The distance d refers to the distance between the pixel under observation and its neighbor. The definitions of GLCM features of diffraction images include 14 original features and 3 extended features [21]. We developed a parallel program using NVIDIA's CUDA on GPUs for calculating GLCM and the 17 features to achieve computational speedup. The size of the co-occurrence matrix scales quadratically with the number of gray levels in the image. The diffraction image in our study is normalized to an 8-bit gray-level range from the originally captured 14-bit image. The GLCM implementation was created to support a wide range of gray-levels in case of the need for further normalization. The result of the optimized GPU implementation showed an average speedup of 7 for GLCM calculation of diffraction images, and 9.83 times for feature calculation [22]. The GLCM matrix and feature calculation results are also checked against a Matlab implementation [21], and a serial implementation in Java.

D. Machine Learning Algorithms

Machine learning algorithms, SVM, k-means clustering and deep learning, are used in this research. The goal of SVM is to build a model based on training data where each instance

has a target value (or class labels) with a set of attributes (or features) and predict the target values for the test data given only their attributes [16]. SVM performs binary classification in general; however, several SVM classifiers can be combined to do multiclass classification by comparing "one against the rest" or "one against one" approaches. The basic idea of SVM is to map the input data on to higher dimensional feature space and determine a maximum margin hyper plane or decision boundary to separate the two classes of feature space. Margin is the distance between the hyperplane and the closest data point. SVM has been widely used in many applications such as classifying cancers in biomedical analysis, text categorization, hand written character recognition. The k-means clustering allows separation of events into k classes according to their distances to k centers under appropriate conditions. If an event is closer to a center c_1 than the others, it is assigned to the cluster represented by c_1 [23].

To optimize the performance and accuracy of cell classification based on diffraction images, we conducted an empirical study to find an optimized feature set for the machine learning algorithm. The feature set for SVM based classification of diffraction images is defined by GLCM features. However, the feature set calculated from GLCM often contains highly correlated features and creates difficulties in computation, learning and classification [24]. We conducted an empirical study to

select optimal GLCM features for the SVM based classification of diffraction images. An approach called Extensive Feature Correlation Study (EFCS) was used in this research to select an optimal set based on the features formulation and numerical results on diffraction images. The results are validated using the Correlation based Feature Selection Algorithm (CFS) [24] study and other research results. Based on EFCS result, we conducted an SVM based classification experiment with combination of the selected features to find an optimal set of GLCM features for the classification. The empirical study also suggests the optimal GLCM displacement d and image gray level g for the cell classification. Validation of the classification is conducted with 10FCV [21] and confusion matrix. Calculating GLCM for large number of diffraction images is computationally intensive. Therefore, we developed a parallel GLCM calculation program for batch processing.

Over the past few years, deep learning becomes the fastest-growing and most exciting method in machine learning for its powerful learning ability after training with massive labeled data [17], as evidenced by breakthroughs ranging from halving the error rate for image based object recognition [25] to defeating professional Go game player in late 2015 [26]. A neural network performs image analysis through many layers, with early layers answering very simple and specific questions about the input image, and later layers building up a hierarchy of ever more complex and abstract concepts. Networks with this kind of many-layers structure are called deep neural networks. Researchers in the 1980s and 1990s tried using stochastic gradient descent and backpropagation to train deep networks. Unfortunately, except for a few special architectures, they didn't have much luck. The networks would learn, but very slowly, and in practice often too slowly to be useful. Since 2006, a set of techniques has been developed that enable learning in deep neural networks. These deep learning techniques are based on stochastic gradient descent and backpropagation, but also introduce new ideas. These techniques have enabled much deeper (and larger) networks to be trained. It turns out that these perform far better on many problems than regular neural networks. The reason is the ability of deep networks to build up a complex hierarchy of concepts. In this research, we conducted a preliminary research on automated selection and classification of diffraction images using a deep Convolutional Neural Network (CNN) called AlexNet [25]. We compared the accuracy and performance of the classification between SVM based approach and deep learning approach and propose future direction for selection and validation of big data.

E. Software for Reconstructing the 3D Structure of a Cell

The software for reconstructing 3D structure of a cell is used for processing confocal image sections of a cell and building its 3D structure based on recognized cell organelles in each image section. The 3D cell structure with assignments of other parameters is converted as an input to ADDA program for simulating the light scattering of the cell. The 3D structure is built based on confocal image sections that are acquired with a stained cell translated to different z-positions using a

confocal microscope. Each image represents a section of the cell structure with very short focal depth (*i.e.* $0.5\mu m$) along the z-axis. Individual nucleus, cytoplasm, and mitochondria stained with different fluorescent dyes are segmented from the image background outside the cell using multiple pattern recognition and image segmentation algorithms based on the pixel histogram and morphological analysis. Then the contours of segmented organelles between neighboring slices, after interpolation of additional slices along the z-axis to create cubic voxels, are connected for 3D reconstruction and voxel based calculations of morphology parameters such as size, shape and volume. Four confocal image sections of a cell are shown in Fig. 3a, a 3D structure of the cell is shown in 3b, and 3c shows a calculated diffraction image of the cell.

F. Software for Modeling Light Scattering of a Cell

The software for calculating diffraction images is obtained with ADDA, which simulates light scattering using the realistic 3D structures reconstructed from confocal images of cells [27]. DDA is a method to simulate light scattering from particles through calculating scattering and absorption of electromagnetic waves by particles of arbitrary geometry [15]. As a general implementation of DDA, ADDA can be used for studying light scattering of many different particles from interstellar dusts to biological cells. The general input parameters of ADDA define the optical and geometry properties of a scatterer/particle including the shape, size, refractive index of each voxels, orientation of the scatterer, definition of incident beam, and many others. ADDA can be configured for producing different outputs for different applications. In our study, we collect the Muller matrix from ADDA simulation to produce diffraction images using ray-tracing technique [28]. Fig. 1(c) shows a calculated diffraction image generated from an ADDA simulation result. With this and the 3D structure reconstruction software, one can vary the structures of different intracellular organelles in a cell and investigate the related changes in texture parameters of the calculated diffraction images. These results allow the study of correlations between the 3D morphology parameters of a cell and the texture feature parameters of the diffraction image. The correlation results build a foundation to obtain candidates of morphology fingerprints from the texture parameters for cell classification based on diffraction images.

III. FEATURE REPRESENTATION, OPTIMIZATION AND VALIDATION

Different feature representations for machine learning of diffraction images of cells have been tried. For example, the frequency and the size of speckles of a diffraction images are learned from k-mean clustering for classifying cell diffraction images and non-cell diffraction images [19]. GLCM features of a diffraction image are used for classifying cell types [29] [29], and multiple layers of image blocks of a diffraction image are used for deep learning of diffraction images in our recent work [30]. The focus of this section is on feature optimization and validation of GLCM features for SVM learning. Automated cell classification based on GLCM features has been developed in Hu's previous work [31] [29]. However, the GLCM

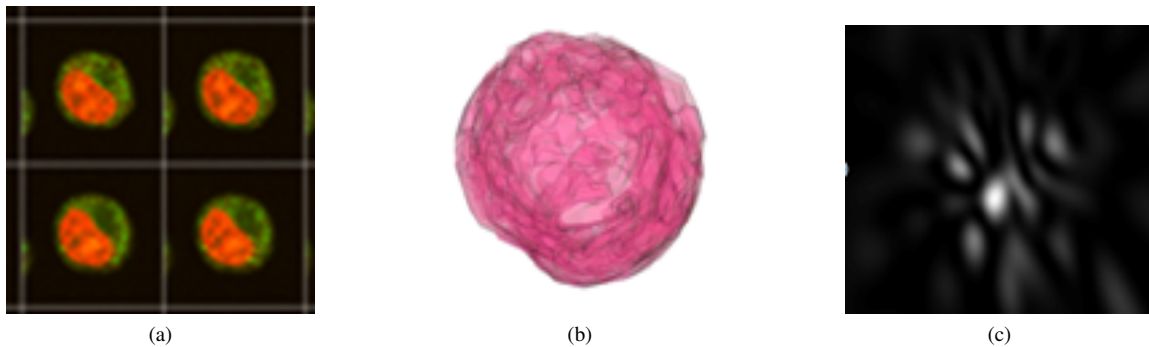


Fig. 3. An example of (a) confocal image sections of a cell, (b) a 3D structure of a cell, and (c) a calculated diffraction image of a cell.

feature set often contains highly correlated features and creates difficulties such as computational intensiveness, slow learning, and in some cases decreased classification accuracy [21], it is important to remove redundant features from the feature set. Feature selection is the process of selecting a subset of total features that are highly correlated with the subject or class and yet uncorrelated with each other [21] [24]. Empirical studies were used to find a set of optimized GLCM features for the cell classification or automated selection of diffraction images. The selected features are validated by classifying diffraction images using SVM, and the classification accuracy is validated using the 10FCV and confusion matrix.

A. Feature Optimization for Cell Classification

In the first experiment, the data set includes 600 diffraction images of 6 types of cells (100 images per cell type). EFCS is used to select an optimal set based on the features formulation and numerical results on diffraction images. Furthermore, to compare and validate the accuracy of these features, a second set of features are selected using CFS [24], one of the most commonly used filter-type feature selection algorithm. All the feature vectors computed in this experiment are labeled with a cell type and hence a supervised learning process is adopted. Also, we need to find an optimal displacement d of GLCM and investigate the gray level of the diffraction images that would result the best performance of cell classification [21].

EFCS selects uncorrelated features by analyzing the trend of all features. First, it lists the feature vectors that consists of all feature values and labeled cell type for each diffraction image. Next, each feature value is normalized. Third, a polynomial regression is used to plot the data trend for every feature of all images. Finally, all features are plotted on a single graph to analyze the correlation between features. In the experiment, four GLCMs were calculated using orientation at 0° , 45° , 90° , and 135° for each image, respectively. Then the average of all 4 orientations is calculated for every single feature. We computed the 17 feature values for each of the 600 images using different displacement d (i.e., 1, 2, 4, 8, 16 and 32) and gray levels (i.e., 8, 16, 32, 64, 128 and 256). This resulted in 36 combinations for each image. Each feature is normalized to values between 0 and 1. GLCM features are categorized into three groups – Contrast, Uniformity, and Correlation [18]. Every feature is categorized into one of three groups. All

features from each group are plotted on a single graph for all 600 images with the same displacement and gray level. Finally, uncorrelated features are obtained from each group for all 108 (i.e., 3 groups x 36 combinations) graphs by visual inspection. The nature of correlation between the features remains similar in all combinations. Eight of the 17 features from three groups are selected into the optimized feature set, which are *CON*, *IDM*, *VAR*, *ENT*, *DENT*, *COR*, *SA*, *IMC1*. The definition of each feature can be found in [21] and [18]. The details of the experiment can be found in Ding's previous work [21].

The CFS algorithm was executed in combination with exhaustive search for the combination of gray level g and displacement d for a total of 36 times for each of the 600 images. Although it yielded slightly different set of features for each combination, a set of eight features is selected. These are the features that are selected the highest number of times in all combinations. The accuracy of cell classification of the 600 images using SVM based on EFCS feature set is slight better than the one with the CFS selected features. Therefore, multiple optimized feature sets may exist.

We used LIBSVM [32], an open source library for SVM, to perform classification of diffraction images using the GLCM features. The type of each cell is known in advance. In the training phase, feature vectors and their corresponding cell type labels are fed to SVM. Next, the 10FCV method is used to check the classification accuracy. This method splits the data into 10 groups of same size. Each group is held out in turn and the classifier is trained on the remaining nine-tenths; then its error rate is calculated on the holdout set (one-tenth used for testing). This learning procedure is repeated 10 times so that in the end, every instance has been used exactly once for testing. Finally, the ten error estimates are averaged to yield the overall error estimation. More specifically, 90 images per each cell type are used for training and the remaining 10 images are used for the testing. Using the 8 EFCS selected features, the SVM classification accuracy achieved for the classification of the 600 diffraction images is 91.16%, which is slightly better than what is achieved by using all the 17 features [21]. Table I shows the cell classification results with different configuration of grey level g and displacement d values. Based on these results, we conclude that the EFCS selected 8 features are effective for SVM based cell classification. Also, when the

TABLE I
SVM BASED CLASSIFICATION WITH THE EFCS SELECTED FEATURES

	d=1	d=2	d=4	d=8	d=16	d=32
g=8	69	71.83	75.16	76.33	73.33	64
g=16	79.66	82	83	82.33	77.833	70
g=32	86.16	89.33	89.33	83.83	80.16	70
g=64	88.16	91.16	89.5	84	79.16	69.16
g=128	89.5	91	89.16	84	79.33	71.5
g=256	89.83	90.83	89.5	86.16	83.33	74.5

grey level g is 64 and displacement d is 2, the accuracy of cell classification is the highest. Therefore, selecting appropriate grey level of the diffraction images and displacement of GLCM could be important to the accuracy of SVM classification. The experiment result indicates that 8 GLCM features in addition to the grey level 64 and displacement 2 are the optimal values for classifying diffraction images using SVM. However, this approach entails enormous computation costs. We processed a total 21,600 diffraction images and extracted 4,320,000 feature values for the feature selection experiment. Therefore, a parallel program based on CUDA framework on GPUs for calculating GLCM matrix and features is essential. Furthermore, an optimal set of eight features performs better than using all the 17 GLCM features. The conclusion is further substantiated by the 10FCV with SVM.

B. Feature Optimization for Image Selection

In last section, we already saw the selected 8 GLCM features can be used for effectively classifying the cell types based on diffraction images using SVM. Guided by the feature optimization result generated from last section, we conducted an empirical study to show how the feature selection would affect the quality of a different SVM classification. In this experiment, SVM is applied for classifying diffraction images of normal cells from those of fractured cells and aggregated particles. We selected 1800 images for each of the three types of cells, and then calculated the 17 GLCM feature values for each image with distance 2 and grey level 64. We first trained the SVM classifiers with all 17 features, and the accuracy of 10FCV for the classification of all three types of cells is between 56% to 61%. Then we removed one feature from the feature matrix each time but keep all 8 features (e.g. *CON*, *IDM*, *VAR*, *ENT*, *DENT*, *COR*, *SA*, *IMC1*) selected in last section to retrain the SVM classifiers, We found the accuracy of classification for all three classifiers was slightly increased when some of the features were removed until some of the eight features were removed. Fig. 4 shows one of the experiment result, where the x axis represents the number of features were removed from the feature matrix, and y axis represents the 10FCV classification accuracy. After we removed the images that are difficult to be classified manually from the training data set, the highest classification accuracy for classifying the normal cells using SVM was 84.6%. The experiment result further shows feature selection is necessary for improving the classification accuracy.

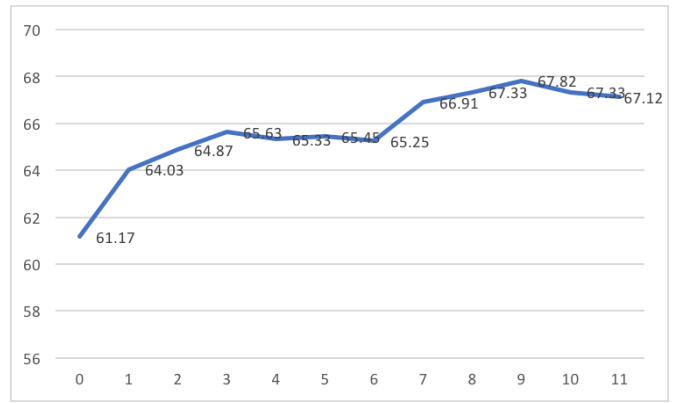


Fig. 4. A feature selection experiment result.

IV. IMAGE DATA SELECTION AND VALIDATION

The diffraction images of cells taken using a p-DIFC may include abnormal images due to debris or fractured cells contained in the samples. The abnormal images decrease the accuracy of the cell classification [5]. If the sample size is small, it is feasible to manually removing the abnormal images. However, when thousands of diffraction images are needed in the machine learning process, an automated approach for separating normal diffraction images from abnormal one is important to the performance and accuracy of the machine learning. In this section, we introduce a machine learning approach for automatically selecting normal diffraction images from whole data set that includes many abnormal images that were produced from aggregated small particles or fractured cells. Different algorithms including SVM with GLCM features, SVM with image preprocessing, and deep learning with CNN are compared for their effectiveness in data section.

A. Data Set

Based on our previous experiment results, we know majority of abnormal diffraction images are generated from fractured cells and aggregated small particles. A fractured cell normally produces stripe patterns in its diffraction image, whereas a normal cell usually generates speckle patterns. The aggregated small particles produce larger speckle patterns than normal cells in their diffraction images [5]. Fig. 5 shows 3 diffraction images with different textual patterns: Fig. 5a is a normal cell with the speckle pattern, Fig. 5b is a fractured cell with the stripe pattern, and Fig. 5c is aggregated small particles with the large speckle pattern.

Based on above observations, we developed a procedure with different algorithms to classify diffraction images into three categories based on their textual patterns in the diffraction images: normal cells, fractured cells, and debris.

B. SVM based Image Data Selection

One of the straightforward approaches for automated selection of diffraction images of cells is to design an SVM classifier based on GLCM features of the images. We selected 2000 diffraction images for each category, and then calculated the 17

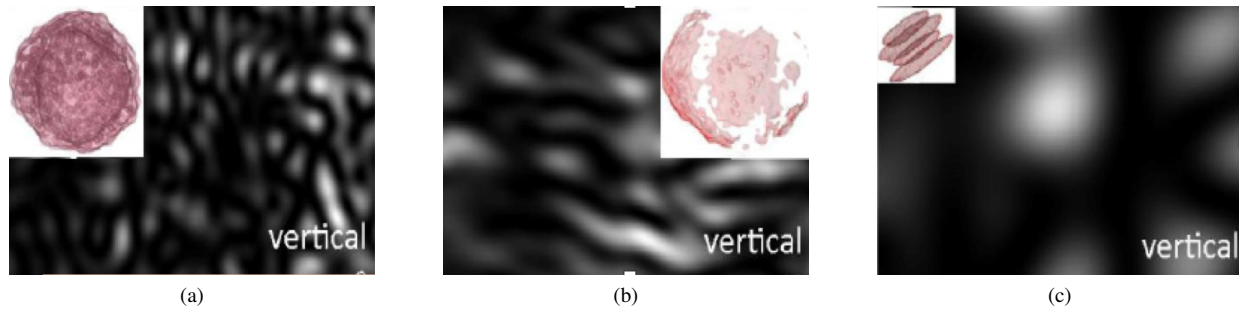


Fig. 5. A diffraction image and its scatterer of (a) a cell, (b) a fractured cell, and (c) aggregated small particles [5].

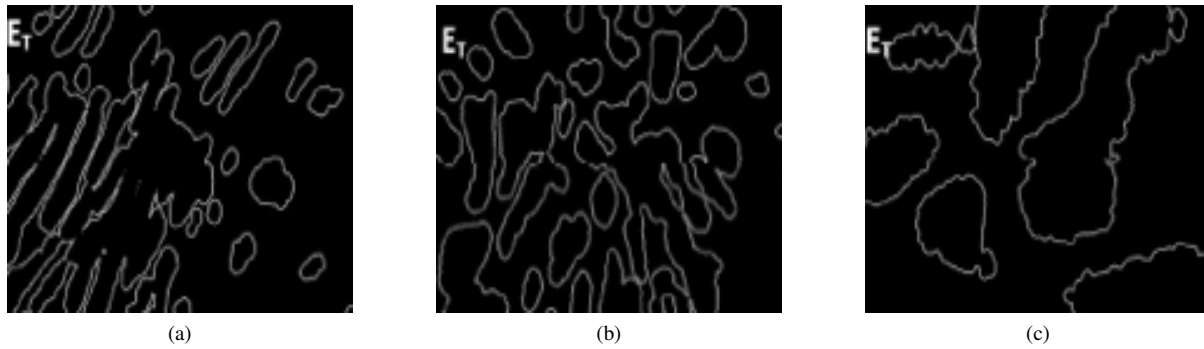


Fig. 6. Diffraction images of (a) borderlines with stripes, (b) normal speckles, and (c) large speckles [5].

GLCM features for each image. The feature matrix consisting of training image feature vectors are input to SVM for training the classifier. An image feature vector includes the image type and its GLCM feature values. 10FCV of the classification was conducted for each SVM classifier and the highest accuracy for the classification of the diffraction images for normal cells, fractured cells and debris was only 61%. Therefore, the experiment result of the simple SVM based diffraction image selection was not ideal, and advanced technique is needed for improving the accuracy of the classification.

C. Image Processing based Data Selection

In order to improving the accuracy of the classification of diffraction images of normal cells, fractured cells and aggregated small particles, advanced image processing algorithms are applied for preprocessing the images. The image data selection procedure includes four steps: The first step is to find a set of borderline length parameters for differentiating the stripe textual pattern from the speckle pattern. Frequency histogram of each diffraction image for measuring the speckle size is calculated next. In the third step, k-means clustering algorithm is applied to calibrate image data and separating images as stripe and speckle patterns. The k-means clustering algorithm also separates diffraction images into two groups: normal speckles and large speckles. The fourth step is to precisely classify the calibrated diffraction images into large speckles (*i.e.*, debris) and normal speckles (*i.e.*, normal cells) using SVM [5].

Step 1: Build borderline length parameters. First, a diffraction image is normalized for the intensity of each pixel,

and then it is converted into a binary $B(z, y)$, using the average pixel intensity as the threshold. By convoluting four Sobel operators with $B(z, y)$, one can derive a set of edge images, which are then used for calculating 5 borderline length parameters as $[C_v, C_h, C_l, C_r]$ and C_T , representing borderlines in four directional edges and one complete edge. The parameters can be used for separating images of the stripe patterns from the speckle patterns [5]. Fig. 6 shows three diffraction images that are processed for borderlines, where 6a shows a stripe pattern, 6b shows a normal speckle pattern, and 6c shows a large speckle pattern. It is easy to differentiate stripes from speckles. However, it is not easy to do so for normal and large speckles since the threshold for separating normal and large speckles has to be built on experiments.

Step 2: Measure speckle size. First, a normalized diffraction image is mapped into the frequency space using a 2D fast Fourier transform (FFT) to obtain a power spectrum image. Next, a frequency threshold is calculated based on an empirical formula. One can derive a histogram $N(f)$ of high frequency pixels in the power spectrum image. N_p as the sum of $N(f)$ is the number of pixels having high power and frequency in the power spectrum image. The diffraction images with normal speckle patterns usually have larger C_T and N_p values than the images with large speckle patterns. However, significant fluctuations in the values among diffraction images of cells exist due to spurious light and great morphological variations of imaged samples. Therefore, a calibration is necessary to minimize the effect of the fluctuations [5].

Step 3: Calibrate diffraction images. The k-means clustering algorithm is used for selecting tighter clustering diffraction

image data. The input to the algorithm is a pair values, N_p and C_T , of each diffraction image. Two clusters are used in this study, which means that $k = 2$, and the images are classified according two centers. The two center values are repeatedly updated with new N_p and C_T until they converge to the final values. To mitigate the fluctuation issues, only N_p and C_T values within some range are counted [5]. The higher ranked image data from the k-mean clustering is sent to an SVM classifier to separate the stripe patterns from speckle patterns. In the SVM, $\langle N_p, C_T \rangle$ pairs on different polarizations form the feature vector for each diffraction image. Although k-means based SVM classification can remove some diffraction images that have large speckles, it is not good enough to separate large speckles from normal speckles. Therefore, SVM based classification based on GLCM features is used for further classifying diffraction images with speckle patterns.

Step 4: Classifying images with speckle patterns. Once the diffraction images that have stripe patterns are separated using the k-means based classification, the images will be classified as normal speckle or large speckle patterns using an SVM classifier. The feature vector consists of all 17 GLCM feature values and labeled image pattern (*i.e.*, large speckle or normal speckle pattern) of each diffraction image. The accuracy is checked using 10FCV against visual inspection. The experiment study shows that the SVM approach achieves over 97% accuracy for separating the diffraction images with large speckle patterns from the normal speckle patterns [5]. Comparing to the simple SVM based approach, the SVM integrated with preprocessing enjoys much higher accuracy of the classification.

D. Deep Learning based Image Data Selection

In previous section, we described how to automate the data selection using SVM. However, the approach requires complex preprocessing including image processing and K-mean clustering. The approach is not scalable since the classification is based on the frequency and the size of the speckles in the diffraction image, which are specifically defined for the classification of normal cells and abnormal cells. In this section, we introduce a deep learning approach for the automated image selection. The diffraction image data set we used is still same as those used in last section. The deep learning framework used is Caffe [33], and the deep learning model is AlexNet [25]. The size of the raw diffraction image is $640 * 480$ pixels, but the input image to AlexNet is $227 * 227$ pixels. The raw images have to be processed before they can be used for training or testing AlexNet. In addition, AlexNet needs a larger training data set than SVM. The deep learning procedure for the image selection can be summarized as follows:

- 1) Produce a training data set. First, find the brightest 10-pixel diameter spot in a raw diffraction image, and then choose the spot as center and crop a $227 * 227$ pixels image from the raw image (make sure the cropped image is located within the original image, same for all other produced images). Then choose a new center that is shift 5 pixels from the center of the spot to crop another $227 * 227$ pixels image. Many different images can be

produced through shifting the center in a direction such as left or right with different distances such as 5 pixels or 8 pixels. Different spots can be identified from an image to produce even more training data for AlexNet. Since partial of a diffraction image normally is enough to represent all information in the diffraction image, the cropped images are good enough for training and testing the deep learning based classification. When we test the classification, any valid cropped image from an original image is used for representing the original image. However, we will experiment different approaches such as sampling and polling technique to find an optimal approach for producing image data from the original one in the future. In addition, multiple instance learning could be a promising direction for addressing the size issue.

- 2) The cropped images are grouped into three folders according to their labels (*i.e.*, cells, strips, and debris). Each cropped diffraction image is labeled as *cell*, *debris* or *strip* same as the label of its original image. In this experiment, the images in each folder are divided into 8 equivalent groups, which includes 6 groups of training data, 1 group of validation data, and 1 group of testing data. The data folders of *cell*, *debris*, and *strip* include 105,072, 121,344, and 99,216 images, respectively.
- 3) The training and testing is run with Caffe on NVidia GPU K40c, and the number of iteration of the training is set as 10000. We conducted a 8FCV for all three types of images, and average classification accuracy for *cell*, *debris*, and *strip* is 94.22%, 97.52%, and 90.34%, respectively. The confusion matrix of the classification is shown in Fig. 7. Comparing to the SVM based classification, deep learning based data selection is much easier with acceptable accuracy. However, deep learning needs large amount of training data, and it doesn't work on the raw images directly, which could be a serious problems for other domain specific images. For example, cytopathology images are much larger and complex than the diffraction images, and it is extremely challenging to obtain enough amount of cytopathology images for deep learning. In that case, SVM based technique is still an alternative for automated data selection.

V. METAMORPHIC TESTING OF SCIENTIFIC SOFTWARE

It is difficult to know whether a reconstructed structure generated by the 3D reconstruction program represents the real morphology of a cell. Also, given an arbitrary input to ADDA, it is difficult to know the correctness of the output. Both these scientific software products are typical of non-testable systems due to unavailability of test oracles. Therefore, we chose metamorphic testing to validate and verify these products and use an iterative approach for developing MRs.

A. Metamorphic Testing

Metamorphic testing is a promising technique for solving oracle problems in non-testable programs. It has been applied to several domains such as bioinformatics systems,

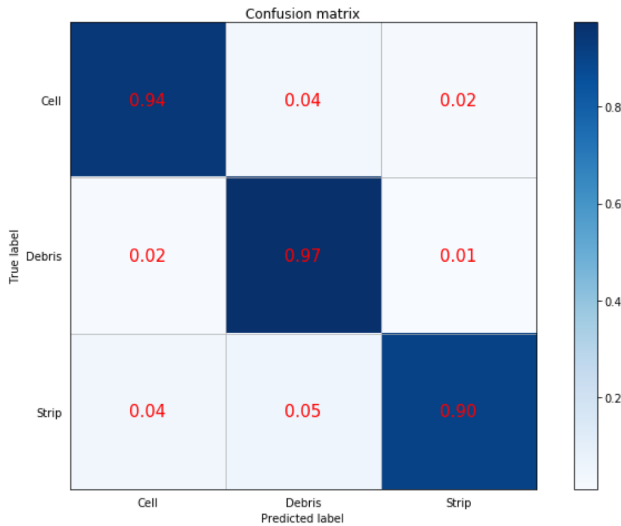


Fig. 7. The confusion matrix of an image classification

machine learning systems, compilers, partial differential equations solvers, large-scale databases, and online service systems. Metamorphic testing will become even more important for testing big data systems since many of them suffer the test oracle problem. Metamorphic testing aims at verifying the satisfiability of an MR among the outputs of the MR related tests, rather than checking the correctness of each individual output [9] [12]. If a violation of an MR is found, the conclusion is that the system under test (SUT) must have defects [9]. Specifically, metamorphic testing creates tests according to an MR and verifies the predictable relations among the actual outputs of the related tests. Let $f(x)$ be the output of test x in program f and t be a transformation function for an MR. Given test x , one can create a new metamorphic test $t(x, f(x))$ by applying function t to test x . The transformation allows testers to predict the relationship between the outputs of test input x and its transformed test input $t(x, f(x))$ according to the MR [9]. However, the effectiveness of metamorphic testing depends on the quality of the identified MRs and tests generated from the MRs. Given a metamorphic test suite with respect to an MR, violation of the MR implies defects in the SUT, but satisfiability of an MR does not guarantee the absence of defects. Therefore, it is important to evaluate the quality of MRs and the tests generated from the MRs. It is even more important to find a way for refining MRs and tests based on testing and test evaluation results. In this research, an iterative metamorphic testing is used for validating the two scientific software systems in CMA.

B. Iterative Metamorphic Testing

The iterative metamorphic testing consists of three major steps: identifying initial MRs and generating initial tests, test execution and evaluation, and refining MRs.

1) *Developing initial MRs and tests*: Based on the domain knowledge of the SUT and general framework of metamorphic testing [34], one can develop a set of initial MRs. The initial test inputs are produced using general test generation

approaches such as combinatorial testing, random testing and category-choice framework, and then each initial test input is converted into another set of test inputs according to an MR. The initial test inputs together with the converted one form a test of an MR. The oracle of the test is the MR that is used to convert the initial test input. The new added test inputs can be used for producing addition tests based on an MR.

2) *Test execution and evaluation*: The SUT is executed with every test, but test outputs are verified by MRs. As soon as the SUT passes all tests, the testing is evaluated for test adequacy criteria. We evaluate the testing with program coverage criteria, mutation testing, and mutated tests. Mutated tests are the tests whose outputs violate an MR. They are used to ensure each MR can differentiate positive tests from the negative one. Mutation testing requires every mutant be killed by at least an MR or weakly killed by a test. A mutant is weakly killed when the output of the mutated program is different from the original one.

3) *Refining MRs*: If a selected program coverage criterion cannot be adequately covered, mutants cannot be killed or weakly killed by existing tests or by simply adding new tests, then new MRs should be developed or existing MRs should be refined. Analyzing existing software engineering data like test results using advanced techniques such as machine learning is a promising approach for developing high quality test oracles and MRs [35]. The ultimate goal of MR refinement is to develop oracles that can verify individual tests.

C. Testing the 3D Structure Reconstruction Software

The most difficult part in our project is correctly building the 3D structures of mitochondria in cells. Each confocal image section may include many mitochondria that are so close to each other that two mitochondria in two adjacent sections could be incorrectly connected. The incorrect connection would result in a wrong 3D structure. But it is infeasible to check the reconstructed 3D structure by comparing it to the original cell that the confocal image was taken from since the cell is either dead or its 3D structure has been greatly changed during its reconstructed structure is produced. Iterative metamorphic testing was used for testing the software.

1) *Developing initial MRs and tests*: Fig. 8 shows a sample input to the program and its corresponding output, where (a) is partial of the input consisting of a stack of confocal image sections of a cell, and (b) is a sectional view of the 3D reconstructed cell. Based on domain knowledge and general guidelines for developing MRs, 5 MRs were created and are list as follows. The details of the MRs have been reported in previous work [36], but we used them to explain the iterative process for developing MRs.

MR1: Inclusive/Exclusive, which defines the correlation between the reconstructed 3D structure and the adding or removing of mitochondria.

MR2: Multiplicative, which defines the relation between the reconstructed 3D structure and the size of selected mitochondria in the image sections.

MR3: Lengths, which defines the relation between the reconstructed 3D structure and the length of selected mitochondria in the image sections.

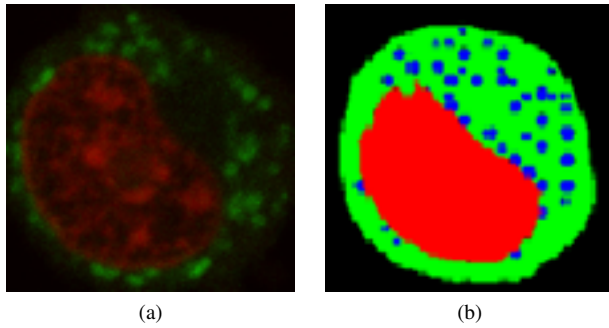


Fig. 8. (a) An example of a confocal image and (b) its processed image.

MR4: Shapes. which defines the relation between the reconstructed 3D structure and the shape of selected mitochondria in the image sections.

MR5: Locations, which defines the relation between the reconstructed 3D structure and the location of selected mitochondria in the image sections.

Tests are generated through transforming existing tests (such as an original image section) according to the MRs. For example, for MR1, new artificial mitochondria could be removed from one or more image sections in a stack of original image sections such as S using Matlab to produce MR1 related image sections T , and then S and T are paired as a test input for MR1. MR related test inputs are executed one by one so that their outputs can be compared to determine whether the test passes the MR. Based on the results, additional tests may be needed.

2) *Evaluation of MRs and tests:* Test adequacy coverage criteria are chosen for evaluating the quality of the test and the evaluation results could be used for refining the MRs and tests. For example, the function, statement, definition-use pair, and mutation coverages were used in this section.

3) *Refining MRs:* MRs *Inclusive/Exclusive* and *Multiplicative* can be further refined to determine the exact change of mitochondria volume. For example, if an artificial mitochondrion is added to the original confocal image sections, the mitochondrion volume can be calculated based on its 3D model using Matlab. Then the volume difference between the original images and the updated ones should be the one calculated using the Matlab. If the result is different, something must be wrong. The refined MR is more effective to find subtle errors such as the one shown in Fig. 9, where A and B are supposed to be connected, but new added C causes C and B be connected. The number of mitochondria in the reconstructed 3D would be the one as expected, and volume of mitochondria in the reconstructed 3D would be increased as expected, but the volume increasing (although it is not necessary to know the exact volume of the mitochondria) between the reconstructed 3D and the calculated 3D model using Matlab will not be the same, which would flag an error in the reconstruction. MR6 is a refined version of MR2.

MR6: Volume. If an artificial mitochondrion whose volume is x is added to the confocal image sections, the volume of mitochondria in the reconstructed 3D structure should be increased by x . The MR is still valid for MRs that are defined

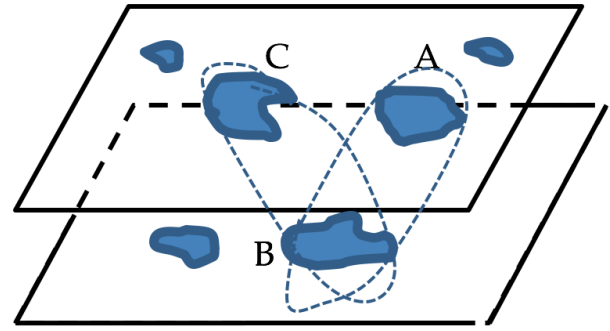


Fig. 9. An illustration of a possible reconstruction error.

on removing or resizing a mitochondrion.

D. Testing ADDA

ADDA has been extensively tested with special cases and other modeling approaches such as Mie theory [27] [15]. Fig. 10 is a comparison of the simulation results of Mie theory and ADDA [27], which shows ADDA and Mie theory produce near identical S_{11} of a sphere scatterer. However, Mie theory only can calculate a regular scatterer. ADDA can simulate a scatterer in any shape. Therefore, it is necessary to test ADDA program for simulating any shape of scatterers using a different approach. A different implementation of DDA for testing the ADDA is not available either. Therefore, iterative metamorphic testing is used for testing ADDA. As stated earlier, testing the software includes three phases: developing initial MRs and tests for the initial testing, evaluation of the initial testing, and refining MRs based on initial testing and test evaluation results. The purpose of the testing in CMA is not to verify the correctness of the implementation of ADDA, instead, it is used for validating whether the simulation results from ADDA can serve the investigation of the morphology fingerprint for classifying cell types based on diffraction images.

1) *Developing initial MRs and tests.* It is infeasible to find an MR that directly defines the relation between an input and its output of ADDA since an input may include dozens of parameters and an output may include thousands of closely related data items. Therefore, we define MRs on the relation between ADDA input parameters and the textual properties of the diffraction image that is generated from an ADDA simulation output. Each input parameter such as the shape, size, orientation, and refractive index of a scatterer could be a candidate for defining MRs, which define the relation between the change of one parameter and the change of the textual patterns in the output diffraction image.

MR7: When the size, shape, orientation, refractive index or internal structure of a scatterer is changed, the textual pattern of the diffraction image is changed.

The MR only considers the change of one parameter each time. When the value of one of these parameters of a scatterer is changed, the textual pattern of the output diffraction image should be different to the original one. Of course, the change of orientation should not affect a perfect sphere. The textual pattern can be compared manually by eyes or is compared by

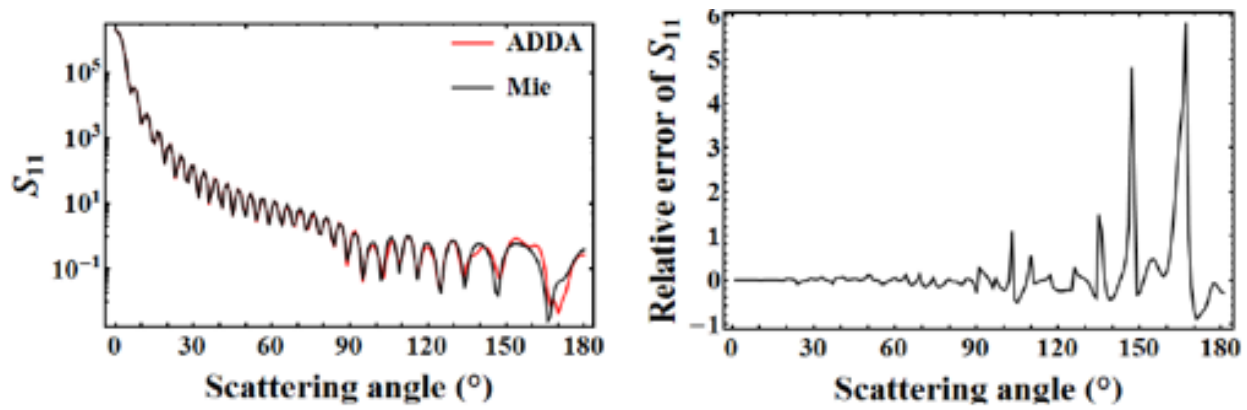


Fig. 10. A comparison between Mie theory and ADDA [27].

the GLCM features calculated from the images. For example, if a scatterer is changed from a sphere into an ellipsoid without changing the value of any other parameters, the textual pattern of the ellipsoid is irregular comparing to the textual pattern of the corresponding sphere as shown in Fig. 11 (a) and (b), where (a) is a sphere, whose x , y and z axes are $5\mu\text{m}$, and (b) is an ellipsoid, whose x and z axes are $5\mu\text{m}$, and y is $7\mu\text{m}$. However, due to the complexity of ADDA, it is infeasible to build an exact relation between the change of a parameter and the textual pattern of the diffraction image in general. For example, Fig. 11 (c) and (d) are two ADDA calculated diffraction images from the same reconstructed morphology structures of a cell but with different orientations, it impossible to know the precise relation between the orientation and the textual patterns of the calculated diffraction images except the "difference". The relation *difference* is too broad to enough test ADDA. Additional MRs are needed for adequately testing ADDA. The idea is to identify MRs that can better define the "difference" when one parameter is changed. The ADDA simulation results of scatterers in regular shapes such as spheres have been tested with Mie theory, which is the foundation for creating other MRs for refining relation "difference".

MR8: *When the size of the sphere becomes larger, the texture bright lines in the diffraction image become slimmer.*

The results can be compared to calculated results from Mie theory, but they should satisfy MR8 also. Fig. 12 shows the ADDA generated diffraction images of sphere scatterers with diameters in $5\mu\text{m}$, $7\mu\text{m}$, $9\mu\text{m}$, and $12\mu\text{m}$, respectively. The output examples satisfy MR8. Furthermore, we define an MR based on the relation between the textual pattern and a sphere scatterer with some portions that were removed. For example, we can check how the textual pattern is changed when a sphere scatterer is cut into half.

MR9: *When a portion of a sphere scatterer is removed, the textual pattern of the diffraction image is changed accordingly.*

We first tested a sphere scatterer with diameter $3\mu\text{m}$, and its ADDA result was checked against the result calculated from Mie theory. Then we used ADDA to simulate the same sphere that was removed by half with the cut part directly facing the incident light beam, and the orientation is (0,0,0) [15]. We also simulated the same sphere that was removed with the top one quarter that is facing the incident light beam, and the

same sphere that was removed with the top outside 1/8 using ADDA. The simulation results are shown in Fig. 13, where we can see that the symmetry property of the textual pattern in the diffraction image of the sphere scatterer is lost when some part of it is removed, and the result is reasonable. The test result satisfies MR9. Using the same idea, we can check the change of textual pattern of the ADDA calculated diffraction images when a sphere scatterer is added with another. For example, we can compare the textual pattern of a sphere and bispheres.

MR10: *When an identical sphere scatterer is added to a sphere scatterer, the textual pattern of the diffraction image is changed accordingly.*

We first calculated a diffraction image for a $5\mu\text{m}$ diameter sphere using ADDA, and then we added one identical sphere to form a bisphere scatterer, and the two spheres are separated by $1.5\mu\text{m}$ and they are aligned along x axis. The orientation of the bisphere is set as (0, 0, 0), which is same as the single sphere (*i.e.*, one sphere is in front of the other related to the incident light beam). Fig. 14 (a) and (b) show the ADDA calculated diffraction images for a sphere scatterer and a bisphere scatterer, respectively. The textual pattern in the diffraction image of the bisphere scatterer clearly shows the two spheres in the scatterer. If we changed the orientation for the sphere from (0,0,0) to (0, 270, 0) and (90, 90, 0), then the textual patterns of their diffraction images are changed as shown in 14 (c) and (d). The results satisfy MR10 and MR7.

We know that when the refractive index, the size of a cell organelle, or the orientation of the scatterer is changed, the textual pattern of the ADDA calculated diffraction image will change. Based on this observation, we can conduct an experiment study using the 3D reconstruction software and ADDA. First reconstruct the 3D structure of a cell based on its confocal image sections using the 3D reconstruction software, and then build a serial of 3D structures through changing the reconstructed 3D structure using Matlab such as resizing the nucleus of the cell, changing refractive index values of some voxels, or orientations [36]. The serial of 3D structures as well the refractive index values of the cell organelles, and the orientation of each scatterer are input to ADDA for producing the diffraction images. The GLCM feature values of each image is calculated, and finally the correction between the GLCM feature values and the morphology change of the

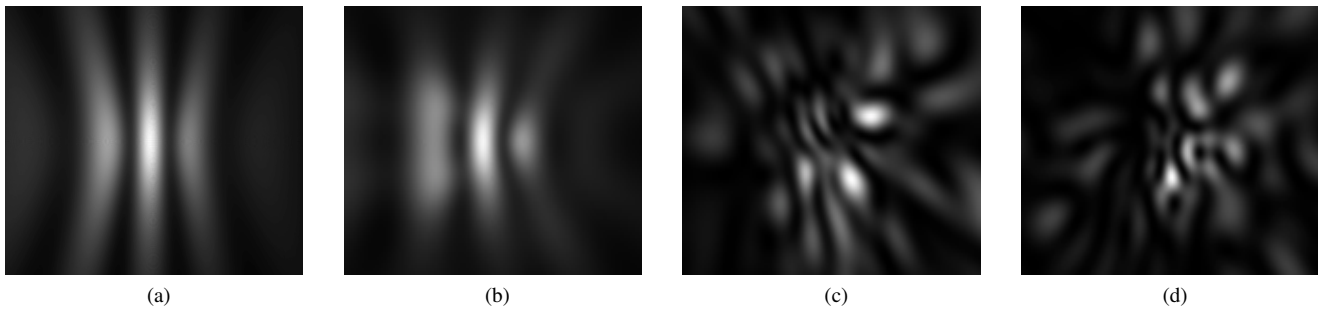


Fig. 11. The change of textual patterns of diffraction images calculated with different configurations, (a) is a sphere, (b) is an ellipsoid, (c) and (d) are cells.

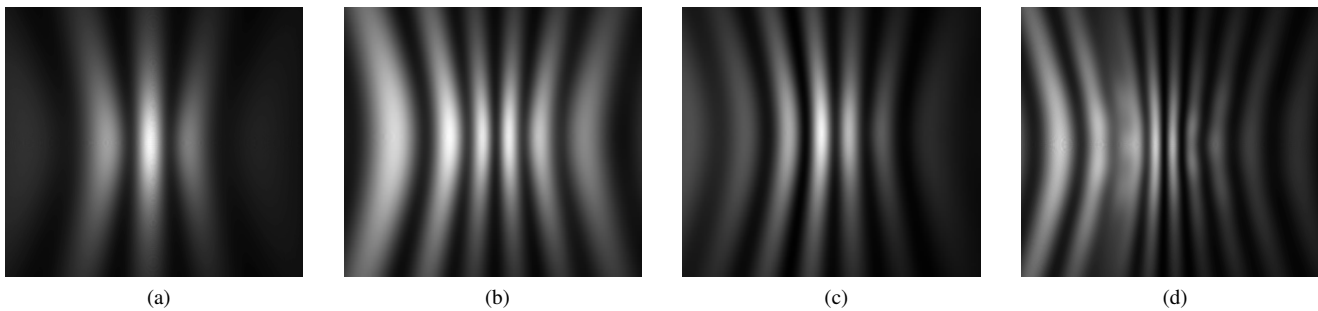


Fig. 12. Textual patterns of ADDA calculated diffraction images of sphere scatterers with different diameters: (a) $5\mu\text{m}$, (b) $7\mu\text{m}$, (c) $9\mu\text{m}$, and (d) $12\mu\text{m}$

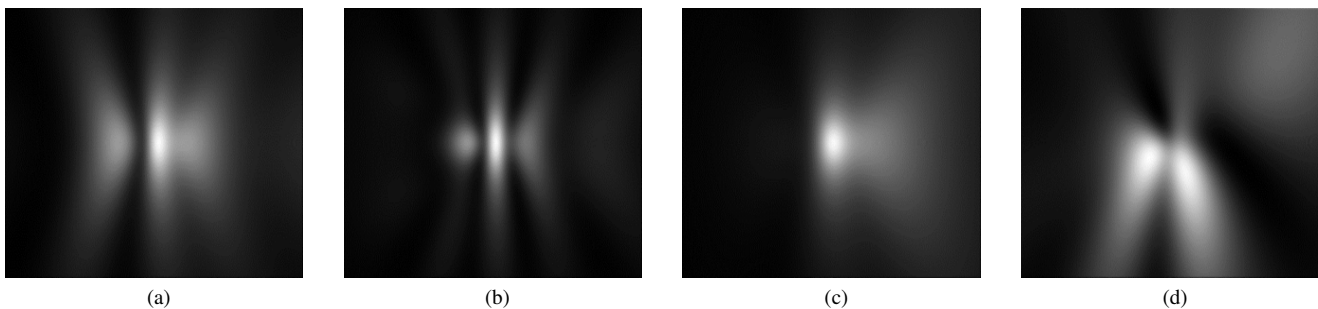


Fig. 13. The change of textual patterns of ADDA calculated diffraction images of a sphere scatterer with partial cut: (a) no cut, (b) $1/2$ cut, (c) $1/4$ cut, and (d) $1/8$ cut.

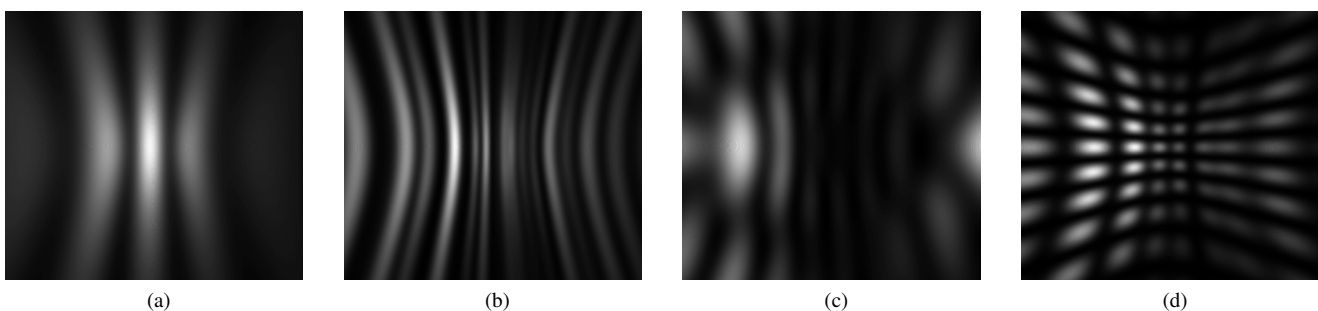


Fig. 14. The change of textual patterns of ADDA calculated diffraction images of a sphere and bisphere scatterers at different orientations: (a) single sphere, (b) bisphere at $(0,0,0)$, (c) bisphere at $(0,270,0)$, and (d) bisphere at $(90,90,0)$.

calculated diffraction images are checked. We already conducted preliminary study to show a correlation exists between GLCM features and the morphology of biological cells [21] [5]. According to experiment results discussed in Section IV, we know a normal cell would generate a diffraction image

with small speckle pattern, and an aggregated small particle would generate a diffraction image with much larger speckle pattern. A fractured cellular structure possesses high degree of symmetry in its structure and would produce a diffraction image with stripe pattern. This observation helps us to develop

MR7', which is a refined MR of MR 7. The diffraction images shown in Fig. 6 were produced by ADDA, and they support MR 7.

MR7': The textual patterns of the diffraction images of normal cells, fractured cells and aggregated small particles are different. Specifically, the textual pattern of the diffraction image of a normal cell is a group of small speckles, the textual pattern of the diffraction image of a fractured cell is a group of stripes, and the textual pattern of the diffraction image of aggregated small particles is group of large speckles.

In order to create tests that can cover as many cases as possible, the combinatorial technique is used to create tests. For example, the four input parameters used for testing ADDA are the *scatterer size, shape, refractive index, and orientation*. The possible values of size are $\{3m, 5m, \dots, 16m\}$, shapes are $\{sphere, ellipsoid, bi-sphere, prism, egg, cylinder, capsule, box, coated, cell1, cell2, \dots\}$, orientations are $\{(0, 0, 0), (10, 90, 0), (270, 0, 0), \dots\}$, and refractive index values are $\{1.0, \dots, 1.5\}$. Using pairwise testing, one can create many tests, and then select the valid tests as the initial tests to create tests for each MR. Using this method, many execution scenarios of ADDA can be tested by the tests and their results can be systematically verified by the MRs, which is not possible for other testing approaches.

2) *Evaluation of MRs and tests.* Several hundred tests were created based on domain knowledge, combinatorial technique and initial MRs. ADDA passed all tests for MR7 to MR10 and the tests covered 100% statements of ADDA program. Mutation testing was conducted to check the effectiveness of the testing but it was applied only to one critical module in ADDA. Instead of testing the program with full mutants created with mutation testing tools, only a few mutants were instrumented in the code manually. We checked the consistency between the outputs of the mutated program and the original one. In the case study, Absolute Value Insertion (ABS) and Relational Operator Replacement (ROR) are the two mutation operators that were used for creating mutants because the two operators achieve an 80% reduction in the number of mutants and only 5% reduction in the mutation score as shown in the empirical results reported by Wong and Mathur [37] [38]. A total of 20 mutants (10 ABS mutants, and 10 ROR mutants) were created and checked. Seventeen of them were killed by crashing or exception of the program execution. The other 3 mutants were killed by the MRs due to the absence of any diffraction pattern in the images. We found a slight change in ADDA may cause a catastrophic error in the calculation, therefore, creating powerful mutants (i.e. a mutant doesn't crash the program or produce trivial errors) for testing ADDA is difficult.

3) *Refining MRs.* Since scatterers in regular shapes such as sphere have been extensively tested [27] [15], we are more interested in scatterers in irregular shapes such as cells. ADDA in this project is used to investigate the correlation between the 3D morphology of a cell and the text pattern of its diffraction image so that to understand how the diffraction images can be used for the classification of cell types. Therefore, we can define MRs based on the classification of diffraction images that are produced by ADDA. The first MR is defined on the correlation of the GLCM features among the diffraction

images of a group of cells, and the second one is defined on the classifications of ADDA calculated diffraction images.

MR11: If the values of a GLCM feature among a group of measured diffraction images from the same type of scatterers are related, then the same relation among the values of the same GLCM feature of the corresponding calculated diffraction images exists.

The experiment process is summarized as follows: (1) Select 100 p-DIFC measured diffraction images taking from one type of cells. (2) Calculate GLCM features for each image, and plot the GLCM feature values and their corresponding image IDs in a two dimensional diagram. (3) Check the relation of each feature among the images. (4) Select 100 cells same as the measured one. Make sure the cells are selected from the same type of cell samples to ensure the cells to be used are as close as possible to the measured one. Then take confocal image sections for each cell using a confocal microscope instrument. (5) Reconstruct the 3D structures of the cells using the 3D reconstruction software, and assign refractive index values of organelles for each cell to produce 3D morphology parameters. (6) Calculate the diffraction image using ADDA with the 3D morphology parameters at the same orientation. (7) Calculate the GLCM features for each calculated diffraction image and plot the feature values in a 2 dimensional diagram. (8) Compare the feature relation between the measured images and calculated one. If the similar relation among the two groups of diffraction images exists, the test passes. Otherwise, further investigation such as producing more calculated images with different orientations is needed. Although the ADDA calculation and p-DIFC measurement were conducted in the same type of cells, their GLCM feature values (or textual patterns) of the diffraction images could be substantially different since the ADDA calculation cannot 100% precisely define the 3D morphology parameters of a real cell such as it is impossible to know the exact value of the refractive index of the nuclear in a cell. Preliminary experimental results as shown in Fig. 15 support MR11. The GLCM values are normalized in Fig. 15. However, the precise relation among the same type of cell images are not easily detected based on one GLCM feature. The comparison of the relations of the measured and calculated images is vaguely defined. Therefore, more advanced MRs are needed for enough testing ADDA.

The new MRs should be created based on the classification of ADDA calculated diffraction images since the classification would need multiple parameters together such as all GLCM features or intensity of all pixels in an image. Two MRs were developed based on the classification of diffraction images using machine learning techniques. The first one is on the classification of scatterers based on their shapes to understand whether the morphology features of the scatterers have been correctly modelled by ADDA so that their diffraction images can be used for the classification. If the shapes of the scatterers can be precisely classified based on the calculated diffraction images, more sophisticated MRs can be developed based on the classification of cell types. We developed the second MR based on the classification of different types of cells based on their ADDA calculated diffraction images.

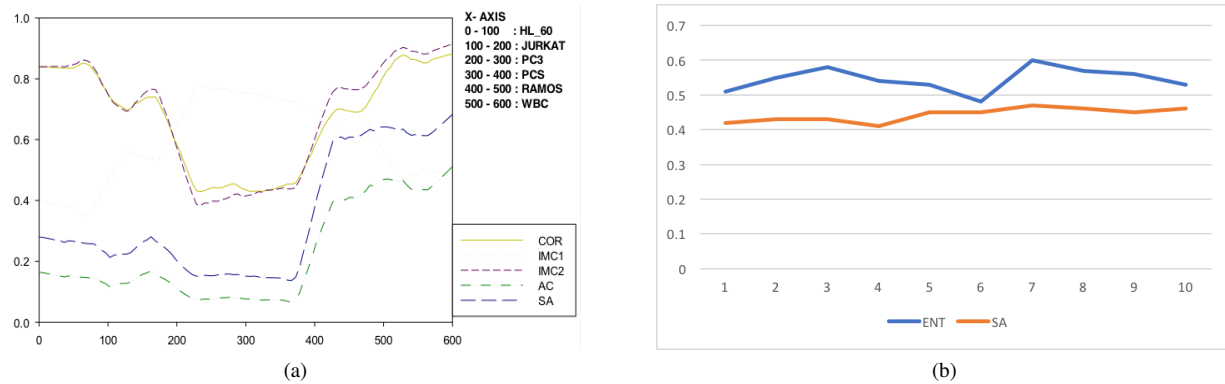


Fig. 15. (a) Values of selected GLCM features of measured diffraction images, it has 600 images for 6 types of cells. [21] (b) Values of two GLCM features of 10 ADDA calculated diffraction images from the same type cells.

MR12: The ADDA calculated diffraction images can be classified by the shapes of the scatterers.

We produced 200 diffraction images for each shape of scatterers using ADDA. The 200 images of the scatterers that are in the same shape were generated with different combinations of parameters size (8 different sizes) and orientation (25 different orientations) of the scatterer. Since refractive index would substantially impact the textual pattern of a diffraction image, all ADDA calculations were assigned with the same refractive index 1.06. A total of 600 images were produced for three shapes: sphere, bi-sphere and ellipsoid. Each image was processed for GLCM feature values and labeled with the shape type of the scatterer. 8 selected GLCM feature values of each image and its labeled shape type form a feature vector, and the SVM classifier is trained and tested with ADDA calculated diffraction images on LIBSVM [32]. The 10FCV result shows the accuracy of the classification of each shape of the scatterers is 100%. The experiment result showed that ADDA was well implemented for regular shape scatterers and the test passed MR12. In ADDA, we can model a scatterer in any shape through specifying the voxels that build the scatterer. Different scatterers can be modeled based on an MR and an initial scatterer and then a metamorphic testing can be conducted through checking the MR among the corresponding ADDA calculated diffraction images. Since great amount of computing resources needed to conduct the experiment, we haven't collected enough data to show its effectiveness. It will be our next step to test ADDA. Finally, we check whether ADDA calculated diffraction images can be used for cell classification.

MR13: The ADDA calculated diffraction images of cells can be classified according to the types of cells.

According to the experiment results discussed in Section IV, we know diffraction images of cells can be used to accurately classify the normal cells, aggregated small particles and fractured cells. Therefore, we can produce a number of diffraction images for the three types of scatterers using ADDA and then check whether the images can be correctly classified using the machine learning algorithms. Fig. 5 shows the ADDA calculated diffraction images of the three different types of scatterers, which have the same patterns as those

images taken by p-DIFC as shown in Fig. 16, where (a) is normal cell, (b) is fractured cell, and (c) is aggregated particles. Using the SVM based classification approach discussed in Section IV, it is not difficult to check whether the ADDA calculated diffraction images can be correctly classified.

Textual patterns in terms of GLCM features in p-DIFC measured diffraction images have been successfully used for classifying cell types [19] [31] [29]. Combining the testing results of MR13, we believe ADDA calculated diffraction images of cells should be sufficient for classifying cell types. We simulated 25 orientations for the 3D morphology parameters of each cell. Each diffraction image is processed for GLCM features and labeled as the type of the cell. The feature vector matrix consisting of the same type of cells is used for training a SVM classifier for classifying cell types. 10FCV is used for checking the classification accuracy. Our preliminary results show that ADDA calculated diffraction images can be used for classifying the types of selected cells. However, there are so many different types of cells, and some of them only are slightly different in 3D structures. Whether diffraction images based cell classification can be used for classifying cell types that are only slightly different in 3D structures is still an open question. The cell types we used in all experiments are significantly different in 3D morphology. If more experiments with many different types of cells still can produce high accuracy in cell classification, it would be safe to conclude that ADDA is well implemented for simulating the light scattering of cells. We haven't conducted a classification study of ADDA calculated diffraction images of cells using deep learning since deep learning requires a much large training data set that is not available to us, and it doesn't work directly on the raw diffraction images due to the large size of the image.

E. Discussion

Software components are one of the major parts in a big data system as shown in Fig. 1. Verification and validation of the software components are challenging due to the absence of test oracles in many cases. The 3D reconstruction software and ADDA software are two typical software that don't have test oracles, which are also called "non-testable" programs.

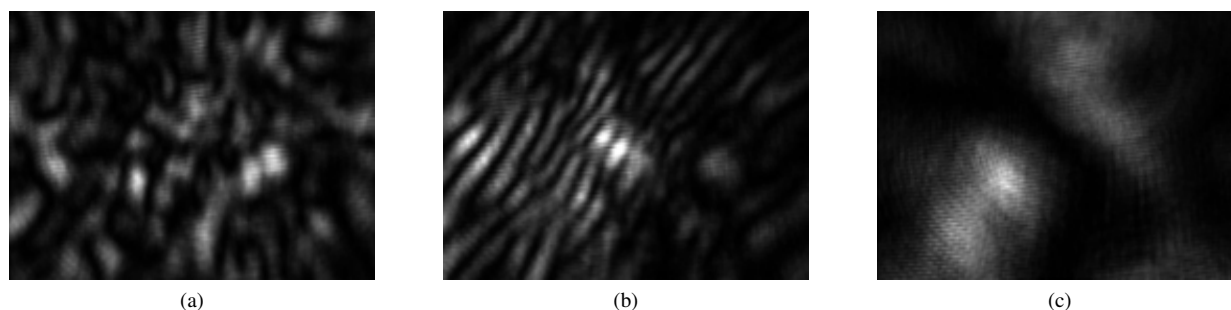


Fig. 16. A p-DIFC measured diffraction image of (a) a cell, (b) a fractured cell, and (c) aggregated small particles.

Although metamorphic testing can be used for testing the non-testable 3D structure reconstruction program and ADDA, the effectiveness of the testing is highly dependent on the quality of MRs. Therefore, the MRs should be rigorously evaluated during the testing, and the initial MRs should be iteratively refined based on testing results. We conducted two empirical studies on verification and validation of "non-testable" program using the iterative metamorphic testing approach. The same approach can be used for testing any software components including regular software in a big data system. The experiment results demonstrated that subtle defects can be detected by the iterative metamorphic testing, but not by the regular metamorphic testing. ADDA is difficult to test due to the difficulty involved in developing highly effective MRs. The empirical study has illustrated the iterative process for building MRs using machine learning approach and demonstrated its effectiveness for testing ADDA. If more data such as more scatterers that have different morphological structures and more scatterers from different types of cells can pass the MRs 7 to 13, it would be safe to conclude that ADDA is enough validated.

VI. RELATED WORK

Quality assurance of big data systems includes quality assurance of data sets, data analytics algorithms and big data applications. The focus of this paper is to propose a framework for the verification and validation of big data systems and illustrate the process with a massive biomedical image data system called CMA. The framework includes testing scientific software, feature selection and validation of machine learning algorithms, and automated data selection and validation. In this section, we discuss related work on the three topics.

Data quality is critical to a big data system since poor data could cause serious problems such as wrong prediction or low accuracy of the classification. The quality attributes of big data include the availability, usability, reliability, and relevance. Furthermore, each attribute includes some detail quality attributes: availability includes accessibility, timeliness, and authorization; usability includes documentation, metadata, structure, readability and credibility; reliability includes accuracy, integrity, completeness, consistency and auditability [39] [1]. Gao, Xie and Tao have given an overview of the issues, challenges and tools of validation and quality assurance of big data [40], where they defined big data quality assurance

as the study and application of quality assurance techniques and tools to ensure the quality attributes of big data. Although many general techniques and tools have been developed for quality assurance of big data, much more work are on the quality assurance of domain specific big data such as health care management data, biomedical data or finance data. Web sources are the main sources of big data, but the trustworthiness of the web sources has to be evaluated. There are many work on the evaluation of the veracity of web sources such as the evaluation based on hyperlinks and browsing history or the factual information provided by the source [41], and evaluation based on the relationships between web sources and their information [42]. Finding the duplicated information from different data sources is also an important task of quality assurance of big data. Machine learning algorithms such as Gradient Boosted Decision Tree (GBDT) have been used for detecting the duplication [43]. Data filtering is an approach for quality assurance of big data through removing bad data from data sources. For example, Apache Samza [44], which is a distributed stream processing framework, has been adopted for finding bad data. Nobles and *et al.* have conducted an evaluation of the completeness and availability of electronic health record data. The quality assurance of big data proposed in this paper is to separate undesired data in data sets using machine learning techniques such as SVM or deep learning. The undesired data could be incorrectly labeled in training data, which is known as class label noise. It could reduce the performance of machine learning. In order to address the problem, one can improve the machine learning algorithm to handle poor data or to improve the quality of the data through filtering the data to reduce the impact of the poor data [45]. Due to the massive scale of big data, automated filtering data using machine learning could be a prefer choice for the improvement of the performance of machine learning. The other quality assurance techniques can be easily integrated into our framework, and our technique could be also easily used for finding poor data in other domain specific big data.

Feature selection is a central issue in machine learning for identifying a set of features to build a classifier for a domain specific task [24]. The process is to reduce the irrelevant, redundant and noisy features to improve learning performance and even accuracy. Hall has reported a feature selection algorithm called CFS to select features based on the correlation of the features and the class they would predict [24]. CFS

has been used for cross checking the feature selection for SVM based classification of diffraction images [21]. In this paper, we introduced an easy to use and more practical experiment based feature selection. The experiment based feature selection would produce a slight better feature set in term of the accuracy of the classification of diffraction images [21]. How the feature selection would impact the classification accuracy or machine learning cost has been also reported [46] [47]. More advanced feature selection approaches such as the one discussed in [48] can be introduced into the framework proposed in this paper. The feature selection discussed above might not be effective for deep learning of biomedical images. Recently, resizing diffraction images through randomly sampling or pooling has been shown as promising techniques in our research of deep learning of diffraction images.

Adequately testing scientific software is a grand challenge problem. One of the greatest challenges that occurs due to the characteristics of scientific software is the oracle problem [11]. Many different approaches have been proposed to address the oracle problem for testing scientific software such as testing the software with special cases, experiment results, even different implementations, or formal analysis of the formal model of the software [11]. However, none of these techniques can adequately testing the scientific software that suffers the oracle problem. Metamorphic testing is a most promising technique to address the problem though developing oracles based on MRs [9] [12]. Metamorphic testing was first proposed by Chen *et al.* [9] for testing non-testable systems. It has been applied to several domains such as bioinformatics systems, machine learning systems, and online service systems. An empirical study has been conducted to show the fault-detection capability of metamorphic relations [49]. A recent application of metamorphic testing to validate compilers has found several hundreds bugs in widely used C/C++ compilers [50]. Metamorphic testing has been applied for testing a large image database system in NASA [51], and also successfully used for the assessment of the quality of search engines including Google, Bing and Baidu [52]. However, the quality of metamorphic testing is highly depended on the quality of the MRs. Knewala, Bieman and Ben-Hur recently reported a result on the development of MRs for scientific software using machine learning approach integrated with data flow and control flow information [35]. In this paper, metamorphic testing is used for the validation of the scientific software in CMA. In our research, the testing evaluation and testing results are used for refining initially created MRs and developing new MRs. MRs would be refined iteratively until all test criteria are adequately covered by the tests. Generation of adequate tests in metamorphic testing is challenging due to the complexity of data types and large number of input parameters in the SUT [11]. Combinatorial technique [53] used for testing software in CMA could be a powerful tool for producing tests for metamorphic testing of scientific software.

VII. CONCLUSION

In this paper, we introduced a framework for ensuring the quality of big data infrastructure CMA. Machine learning

based procedures including SVM and deep learning were introduced to automate the data selection process and an experiment based approach was proposed for feature optimization to improve the accuracy of machine learning based classification. An iterative metamorphic testing was used for validating the scientific software in CMA, and machine learning was used for developing and refining MRs. Cross validation and confusion matrix were conducted for evaluating the machine learning algorithms. The framework addressed the most important issues on the verification and validation of big data, and it can be used for verification and validation of any big data systems in a systematic and rigorous way. In the future, our focus is on the quality measurement and validation of big data using machine learning techniques.

ACKNOWLEDGMENT

The authors would thank Sai Thati, John Dixon, Eric King, Jiabin Wang and Min Zhang at East Carolina University and Jun Zhang at Tianjin University for assistances of the experiments. This research is supported in part by grants #1262933 and #1560037 from the National Science Foundation. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

REFERENCES

- [1] V. Gudivada, R. Raza-Yates, and V. Raghavan, "Big data: Promises and problems," *IEEE Computer*, vol. 48, no. 3, pp. 20–23, 2015.
- [2] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [3] Apache. (2016) Hadoop. [Online]. Available: <http://hadoop.apache.org/>
- [4] V. Gudivada, D. Rao, and V. Raghavan, "Renaissance in database management: Navigating the landscape of candidate systems," *IEEE Computer*, vol. 49, no. 4, pp. 31–42, 2016.
- [5] J. Zhang, Y. Feng, M. S. Moran, J. Lu, L. Yang *et al.*, "Analysis of cellular objects through diffraction images acquired by flow cytometry," *Opt. Express*, vol. 21, no. 21, pp. 24 819–24 828, 2013.
- [6] R. M. and T. Poggio, "Models of object recognition," *Nature Neuroscience*, vol. 3, pp. 1199–1204, 2000.
- [7] K. Jacobs, J. Lu, and X. Hu, "Development of a diffraction imaging flow cytometer," *Opt. Lett.*, vol. 34, no. 19, p. 29852987, 2009.
- [8] (2016) Adda project. [Online]. Available: <https://github.com/adda-team/adda>
- [9] T. Y. Chen, S. C. Cheung, and S. Yiu, "Metamorphic testing: a new approach for generating next test cases," Tech. Rep. HKUST-CS98-01, Dept. of Computer Science at Hong Kong Univ. of Science and Technology, 1998.
- [10] J. Ding, D. Zhang, and X. Hu, "An application of metamorphic testing for testing scientific software," in *1st Intl. workshop on metamorphic testing with ICSE*, Austin, TX, May 2016.
- [11] U. Kanewala and J. M. Bieman, "Testing scientific software: A systematic literature review," *Information and Software Technology*, vol. 56, no. 10, pp. 1219–1232, 2014.
- [12] S. Segura, G. Fraser, A. Sanchez, and A. Ruiz-Cortés, "A survey on metamorphic testing," *IEEE Trans. on Software Engineering*, vol. PP, no. 99, pp. 1–1, 2016.
- [13] (2016) MongoDB. [Online]. Available: <https://www.mongodb.com/>
- [14] (2016) Mongochef. [Online]. Available: <http://3t.io/mongochef/>
- [15] M. Yurkin and A. Hoekstra. (2014) User manual for the discrete dipole approximation code adda 1.3b4. [Online]. Available: <https://github.com/adda-team/adda/tree/master/doc>
- [16] C. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," National Taiwan University, 2003.
- [17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [18] R. Haralick, "On a texture-context feature extraction algorithm for remotely sensed imagery," in *Proceedings of the IEEE Computer Society Conference on Decision and Control*, Gainesville, FL, 1971, pp. 650–657.

- [19] K. Dong, Y. Feng, K. Jacobs, J. Lu, R. Brock *et al.*, "Label-free classification of cultured cells through diffraction imaging," *Biomed. Opt. Express*, vol. 2, no. 6, p. 17171726, 2011.
- [20] R. M. Haralick, K. Shanmugan, and I. H. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 610–621, 1973.
- [21] S. K. Thati, J. Ding, D. Zhang, and X. Hu, "Feature selection and analysis of diffraction images," in *4th IEEE Intl. Workshop on Information Assurance*, Vancouver, Canada, 2015.
- [22] J. Dixon and J. Ding, "An empirical study of parallel solution for glcm calculation of diffraction images," Orlando, FL, 2016.
- [23] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2012.
- [24] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, Hamilton, Newzealand, 1999.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1097–1105.
- [26] E. Gibney, "Google ai algorithm masters ancient game of go," *Nature*.
- [27] M. Moran, "Correlating the morphological and light scattering properties of biological cells," Ph.D. dissertation, East Carolina University, Greenville, NC, 2013.
- [28] R. Pan, Y. Feng, Y. Sa, J. Lu, K. Jacobs, and X. Hu, "Analysis of diffraction imaging in non-conjugate configurations," *Opt. Express*, vol. 22, no. 25, pp. 31 568–31 574, 2014.
- [29] X. Yang, Y. Feng, Y. Liu, N. Zhang, L. Wang *et al.*, "A quantitative method for measurement of hl-60 cell apoptosis based on diffraction imaging flow cytometry technique," *Biomed Opt Express*, vol. 5, no. 7, p. 21722183, 2014.
- [30] M. Zhang, "A deep learning based classification of large scale biomedical images," Department of Computer Science, Tech. Rep., 2016.
- [31] Y. Feng, N. Zhang, K. Jacobs, W. Jiang, L. Yang *et al.*, "Polarization imaging and classification of jurkat t and ramos b cells using a flow cytometer," *Cytometry A*, vol. 85, no. 11, pp. 817–826, 2014.
- [32] C.-C. Chang and C.-J. Lin. (2016) Libsvm. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [33] (2016) Caffe project. [Online]. Available: <http://caffe.berkeleyvision.org/>
- [34] J. Mayer and R. Guderlei, "An empirical study on the selection of good metamorphic relations," in *30th Annual International Computer Software and Applications Conference (COMPSAC06)*, 2006, pp. 475–484.
- [35] U. Kanewala, J. M. Bieman, and A. Ben-Hur, "Predicting metamorphic relations for testing scientific software: A machine learning approach using graph kernels," *Journal of Software Testing, Verification and Reliability*, vol. 26, no. 3, pp. 245–269, 2015.
- [36] J. Ding, T. Wu, J. Q. Lu, and X. Hu, "Self-checked metamorphic testing of an image processing program," in *4th IEEE Intl. Conference on Security Software Integration and Reliability Improvement*, Singapore, 2010.
- [37] W. E. Wong and A. Mathur, "Reducing the cost of mutation testing: an experimental study," *J. Systems and Software*, vol. 31, no. 3, pp. 185–196, 1995.
- [38] Y. Jia and M. Harman, "An analysis and survey of the development of mutation testing," *IEEE Trans. Softw. Eng.*, vol. 37, no. 5, pp. 649–678, 2011.
- [39] L. Cai and Y. Zhu, "The challenges of data quality and data quality assessment in the big data era," *Data Science Journal*, vol. 14:2, pp. 1 – 10, 2015.
- [40] J. Gao, C. Xie, and C. Tao, "Big data validation and quality assurance – issues, challenges, and needs," in *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, March 2016, pp. 433–441.
- [41] X. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang, "Knowledge-based trust: Estimating the trustworthiness of web sources," *Proc. VLDB Endow.*, vol. 8, no. 9, pp. 938–949, May 2015.
- [42] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, no. 6, pp. 796–808, Jun. 2008.
- [43] C. H. Wu and Y. Song, "Robust and distributed web-scale near-dup document conflation in microsoft academic service," in *2015 IEEE International Conference on Big Data (Big Data)*, Oct. 2015, pp. 2606–2611.
- [44] (2016) Apache samza. [Online]. Available: <http://samza.apache.org/>
- [45] J. A. Saez, B. Krawczyk, and M. Wozniak, "On the influence of class noise in medical data classification: Treatment using noise filtering methods," *Applied Artificial Intelligence*, vol. 30, no. 6, pp. 590–609, Jul. 2016.
- [46] M. Yousef, D. S. D. Muserref, W. Khalifa, and J. Allmer, "Feature selection has a large impact on one-class classification accuracy for micrnas in plants," *Advances in bioinformatics*, vol. 2016, 2016.
- [47] F. Min, Q. Hu, and W. Zhu, "Feature selection with test cost constraint," *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 167–179, 2014.
- [48] H. A. L. Thi, H. M. Le, and T. P. Dinh, "Feature selection in machine learning: an exact penalty approach using a difference of convex function algorithm," *Machine Learning*, vol. 101, no. 1-3, pp. 163–186, 2015.
- [49] H. Liu, F.-C. Kuo, D. Towey, and T. Chen, "How effectively does metamorphic testing alleviate the oracle problem?" *IEEE Trans. on Software Engineering*, vol. 40, no. 1, pp. 4–22, 2014.
- [50] V. Le, M. Afshari, and Z. Su, "Compiler validation via equivalence modulo inputs," in *35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, Edinburgh, United Kingdom, 2014, pp. 216–226.
- [51] M. Lindvall, D. Ganesan, R. rdal, and R. E. Wiegand, "Metamorphic model-based testing applied on nasa dat: an experience report," in *37th Intl. Conference on Software Engineering*, vol. 2, May 2015, pp. 129–138.
- [52] Z. Zhou, S. Xiang, and T. Chen, "Metamorphic testing for software quality assessment: A study of search engines," *IEEE Trans. on Software Engineering*, vol. 42, no. 3, pp. 264–284, 2016.
- [53] C. Nie and H. Leung, "A survey of combinatorial testing," *ACM Computer Survey*, vol. 43, no. 2, pp. 1–29, Feb. 2011.

PLACE
PHOTO
HERE

Junhua Ding is an Associate Professor of Computer Science at East Carolina University, where he has been since 2007. He received a B.S. in Computer Science from China University of Geosciences in 1994, an M.S. in Computer Science from Nanjing University in 1997, and Ph.D. in Computer Science from Florida International University in 2004. During 2000–2006 he was a Software Engineer at Beckman Coulter. From 2006–2007 he worked at Johnson and Johnson as a Senior Engineer. His research interests center on improving the understanding, design and quality of biomedical systems, mainly through the application of machine learning, formal methods and software testing. He has published over 60 peer-reviewed papers, and his research is supported by two NSF grants. Junhua Ding is a member of IEEE and ACM.

Xin-Hua Hu is a Professor of Physics at East Carolina University.

Venkat Gudivada is a Professor and Chair of Computer Science at East Carolina University.