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 which is captured in diffraction images. CMA includes a collection of scientific software tools, machine learning algorithms, and a large-scale cell image repository. In order to ensure the quality of big data system CMA, we developed a framework for rigorously validating the massive scale image data as well as adequately verifying both the software tools and machine learning algorithms. The validation of big data is conducted by iteratively selecting the data using a machine learning approach. An experimental approach guided by a feature selection algorithm is introduced in the framework to select an optimal feature set for improving the machine learning performance. The verification of software and algorithms is developed on the iterative metamorphic testing approach due to the non-testable property of the software and algorithms. 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 algorithm is evaluated with a stratified N-fold cross validation and confusion matrix. We describe the design of the proposed big data verification and validation framework with CMA as the case study, and demonstrate its effectiveness through verifying and validating the dataset, the software and the algorithms in CMA.

Index Terms—Big data, diffraction image, machine learning, deep learning, metamorphic testing

1 Introduction

Volume, velocity, variety, and value are the four characteristics that differentiate Big Data from regular data [1]. Volume and velocity refer to the unprecedented amount of data and the speed of its generation. Big Data is complex and heterogeneous. 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 applications [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 characteristics of big data create new challenges for the validation and verification tasks [5]. For example, data selection and validation are critical to the effectiveness and performance of big data analysis, but large volume and variety create a grand challenge. Existing work has shown that abnormal data existing in datasets could substantially impact the value extraction and decrease the accuracy of data analysis [6].

Many data analytics tools are complex and are difficult to test due to the absence of test oracles. Other approaches for verifying complex software 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, little 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 verification of data analytics software and algorithms as well as big data. To achieve the best validation and verification performance, feature representation, feature extraction and feature selection for machine learning used in the framework 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 provides techniques for big data validation through automated selection and classification of big data. The middle layer features an approach for verification and validation of machine learning algorithms including feature representation, extraction and optimization. Lastly, the top layer provides an approach for testing domain modeling systems, data analytics tools and applications. The framework covers the essential verification and validation tasks that are needed for any big data application,

- *J. Ding and V. Gudivada are with the Department of Computer Science, East Carolina University, Greenville, NC 27858 USA. E-mail: {dingj, gudivadav15}@ecu.edu.*
- *X. Hu is with the Department of Physics, East Carolina University, Greenville, NC 27858 USA. E-mail: hux@ecu.edu.*

Manuscript received 18 Apr. 2016; revised 14 Dec. 2016; accepted 24 Feb. 2017. Date of publication 9 Mar. 2017; date of current version 28 May 2021. Recommended for acceptance by V. Gudivada, V. Raghavan, and L. Berti-Equille. Digital Object Identifier no. 10.1109/TBDATA.2017.2680460

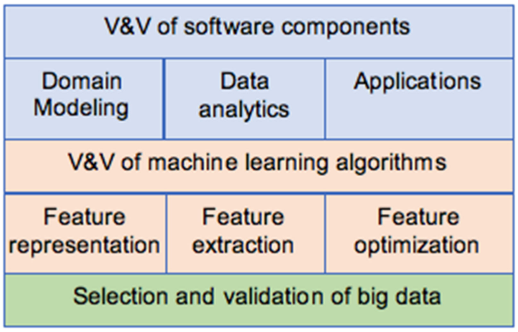


Fig. 1. A schema for V&V of big data systems.

and the techniques and tools proposed in this paper can be easily applied to the quality assurance of other big data applications.

We explain the proposed approach and demonstrate its effectiveness with a case study—a big data system called Cell Morphology Assay (CMA) for automated classification of diffraction images of biological cells. 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 for the development of morphology based cell assay tools. It includes a collection of scientific software tools for processing and analyzing image data, machine learning algorithms for feature selection and cell classifications, and a database for managing the big data.

The verification and validation framework of CMA supports image data validation, verification and validation of the machine learning algorithms and 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 approach for automatically selecting and classifying images. In order to find a better machine learning algorithm for the classification, different feature representations were investigated and reported. The validation of machine learning algorithms consists of two steps. First, optimized features are selected to achieve best performance and effectiveness of the machine learning algorithms. Different feature selection approaches are used for cross checking the selected features. N-Fold Cross Validation (NFCV) of the machine learning results is done in the second step.

The verification and validation of the scientific software in CMA is conducted with an *iterative metamorphic testing* [7], which is a metamorphic testing extended with iterative development of test oracles [8]. One major component of CMA is a collection of scientific software for supporting scientific investigation and decision making [9]. For example, 3D structure reconstruction software and light scattering modeling software are two such pieces of software in CMA. Many scientific software systems are non-testable because of the absence of test oracles [7], [9]. Metamorphic testing [7], [10] is a novel software testing technique and a promising approach for solving oracle problems. It creates tests according to metamorphic relation (MR) and verifies the predictable relation among the outputs of the related tests. However, the application of metamorphic testing to large-scale scientific software is rare because the identification of MRs for adequately testing complex scientific software is infeasible [9]. This paper introduces an iterative approach for developing MRs, where MRs are iteratively refined with reference to the analysis of test execution and evaluation results.

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. The case study 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. Section 5 explains the testing of scientific software in CMA. Section 6 describes the related work, and Section 7 concludes the paper.

2 Massive Scale Image Data System CMA

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

2.1 The Architecture of CMA

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 [11].

3D cell morphology provides rich information about cells that is essential for cell analysis and classification. Diffraction images of single cells are acquired using a polarization Diffraction Imaging Flow Cytometer (p-DIFC), which was invented and developed by co-author Hu [12]. Co-authors Ding and Hu have been studying cell morphology assay and classification for over a decade and developed big data system CMA. This system is used for modeling and analyzing 3D cell morphology and to identify and extract

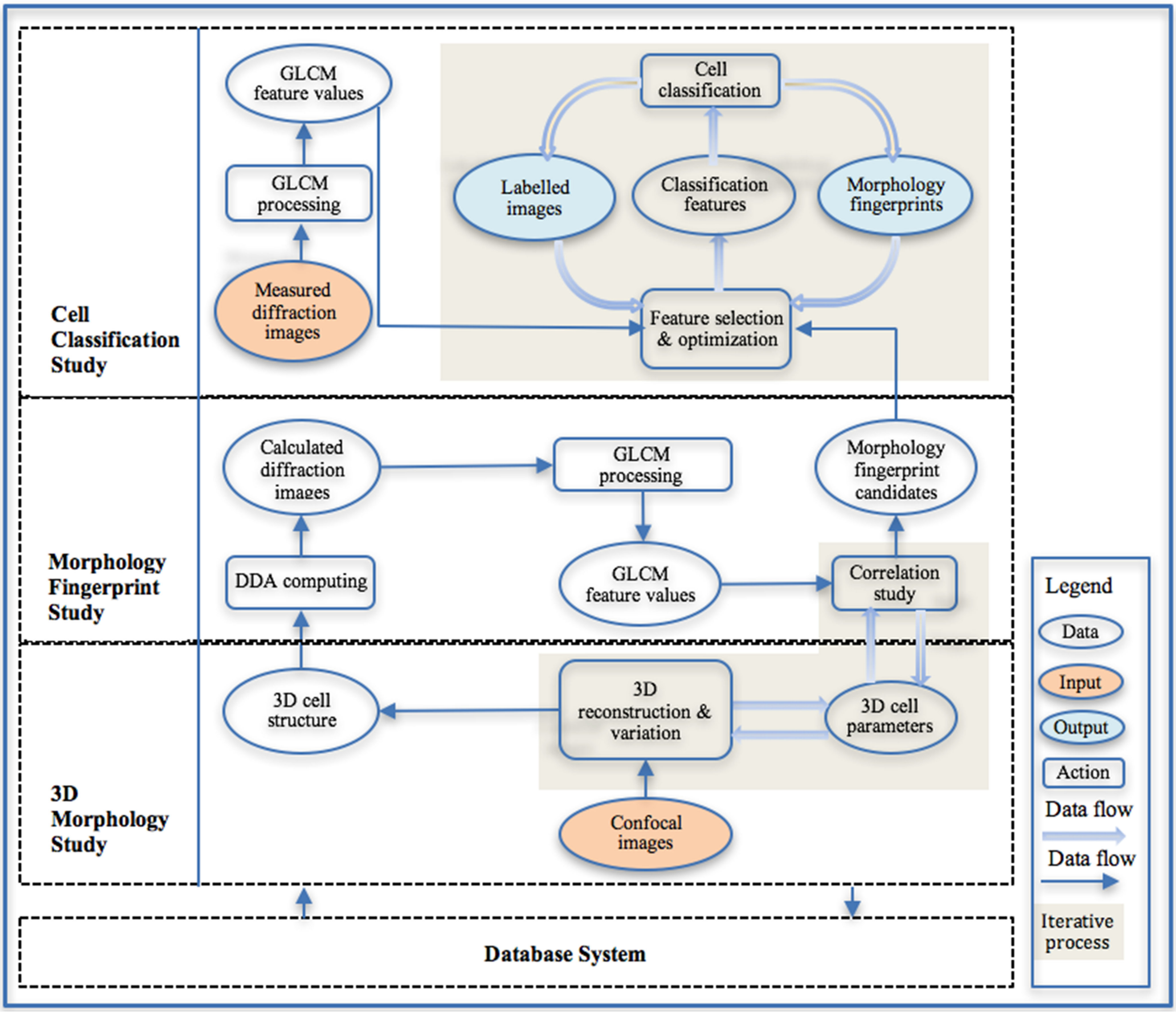


Fig. 2. An overall structure of CMA.

morphology patterns from the diffraction images of biological cells to support automated cell classification. The morphology 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.

The architecture and data flow of CMA is shown in Fig. 2. CMA includes four major components: a database, software tools for investigating 3D morphology of cells, another set of software tools for extracting morphology fingerprints from diffraction images of cells, and a framework for cell classification study. The foundation of CMA is the database, and the core of CMA is a set of data analytics and image processing algorithms. The principal function of CMA is realized by a collection of software components, which compute the morphology fingerprints from diffraction images. The fringe pattern of a diffraction image defines the unique 3D morphology information of the cell type. Therefore, fringe patterns extracted from diffraction images could be effective for classifying cell types. However, means for defining the fringe patterns, how these patterns are correlated to 3D morphology of a cell, and finding the optimal fringe pattern parameters for the classification are unknown. CMA is designed to answer these questions.

In order to investigate the correlation between the fringe pattern of a cell diffraction image and the 3D morphology of the cell, we model the light scattering properties of a cell based on its 3D morphology parameters using a scientific software tool. The modeling result of the light scattering of a cell is converted into a diffraction image, and the correlation between the 3D morphology parameters and the fringe pattern of the diffraction image is established through an experimental study, which systematically changes the values of the 3D parameters to see the corresponding changes of the fringe pattern in the diffraction image. To generate the 3D morphology parameters of a cell, a stack of confocal image sections are taken from the cell using a confocal microscope. Next, the confocal image sections are reconstructed for the 3D structure of the cell, and each cell organelle in the reconstructed 3D structure is assigned with a refractive index value. The 3D morphology parameters comprise a 3D structure with assigned refractive index value for every cell organelle. The study would establish the morphology fingerprints that can be used for classifying cells. Once the selected morphology fingerprints are refined and confirmed, they are used for classifying diffraction images using machine learning algorithms. Depending on the machine learning algorithm selected for the classification, suitable morphology fingerprints need to be selected and optimized for better classification accuracy and performance.

We call the diffraction images that are taken using a p-DIFC instrument as *measured diffraction images*, and the diffraction image that is calculated using the modeling software as *calculated diffraction image*.

2.2 The Database

The database system is developed using MongoDB [13] and MongoChef [14] as client application to support remote access via Internet. Given that the number and the type of features vary from one application to another, relational databases are not suitable for CMA implementation. Furthermore, the data is primarily used for analytics rather than query and update, NoSQL MongoDB is a preferred choice over relational databases. The image data stored in the database includes three collections: measured diffraction images and their processing results; calculated diffraction images and their processing results; and the 3D reconstructed structures and morphology parameters data and the corresponding confocal images. The measured diffraction images of cells are acquired using p-DIFC, and the calculated diffraction images are generated using a light scattering modeling tool called ADDA [15], [16]. ADDA is an implementation of Discrete Dipole Approximation (DDA) [15]. The confocal images of cells are taken using confocal microscopes and are used for reconstructing the 3D structure of cells. The data processing results include 3D cell structure data, 3D cell morphology 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 600,000 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 also contain noise images. For example, if a blood sample contains non viable cells or small particles, the diffraction images taken from such a sample will include abnormal diffraction images. If the latter are labeled as viable cells in the training set, the accuracy of the cell classification could be substantially decreased [6]. Therefore, classification and separation of the abnormal data is important for ensuring high classification accuracy. In this paper, we use two machine learning approaches to address the issue. The first is Support Vector Machine (SVM) [17] based approach, which is integrated with image processing algorithms for automatically identifying abnormal diffraction images and separating them from the normal ones. We tried different SVM kernel functions in our experiments, and only the linear kernel function produced the best results. The second is a deep learning [18] based approach. We experimented different deep learning architectures, and AlexNet [19] produced the best results.

2.3 A High Speed GLCM Calculator

To enable quantitative characterization of fringe patterns in the diffraction images, Gray-Level Co-occurrence Matrix (GLCM) [20], [21] features are computed. Haralick proposed GLCM for describing computable texture features based on gray-tone spatial dependencies [22]. It defines how often different combinations of gray level pixels occur in an image for a given displacement/distance d at 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 ones [23]. We developed a parallel program using NVIDIA's CUDA on GPUs for computing 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. However, the GLCM implementation supports a wide range of gray-levels. The results of the optimized GPU implementation show an average speedup of 7 times for GLCM calculation, and 9.83 times speedup for feature calculation [24]. The GLCM matrix and feature calculation results are also checked against a Matlab implementation [23], and a serial implementation in Java.

2.4 The Machine Learning Algorithms

SVM, k-means clustering, and deep learning are the machine learning algorithms used in this study. The goal of SVM is to build a classification model using training data where each instance has a target value (or class label) with a set of attributes (or features). Once the model is trained, it is used to predict target values for the test data with unknown target values [17]. 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 a SVM is to map the feature data on to a higher dimensional feature space and determine a maximum margin hyper plane or decision boundary to separate the two classes in the 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, and hand written character recognition. The k-means clustering algorithm 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 the center c_1 [25].

To improve 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. The feature set for SVM based classification of diffraction images is defined by the GLCM features. However, the feature set calculated from GLCM often contains highly correlated features and creates difficulties in computation, and model building [26]. An approach called Extensive Feature Correlation Study (EFCS) was used in this research to select an optimal feature set based on the features' formulation and numerical results on diffraction images. The results are validated using the Correlation based Feature Selection

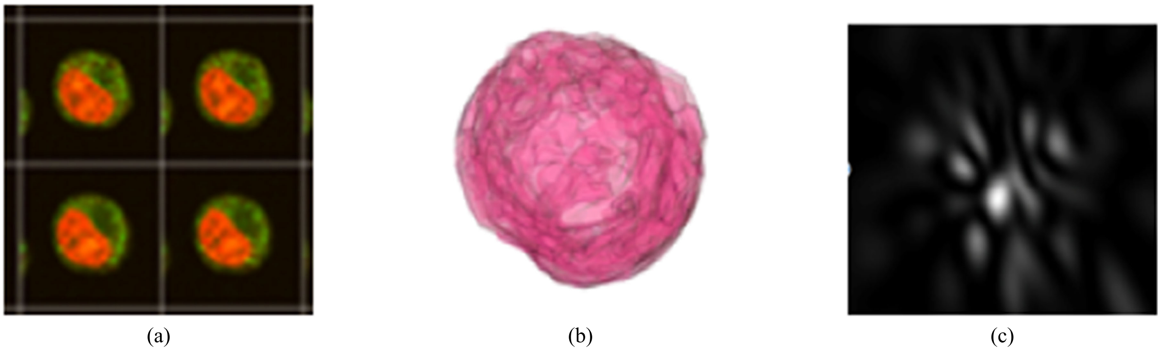


Fig. 3. An example of (a) confocal image sections of a cell, (b) a reconstructed 3D structure of a cell, and (c) an ADDA calculated diffraction image of a cell.

algorithm (CFS) [26] study and other research results. Based on EFCS result, we conducted an SVM based classification experiment with combinations of the selected features to find an optimal set of GLCM features. The empirical study also suggests the optimal GLCM displacement d and image gray level g for cell classification. Validation of the classification is conducted with 10FCV [23] and confusion matrix.

Over the past few years, deep learning approach has become very popular for classification problems [18]. Its breakthroughs ranging from halving the error rate for image based object recognition [19] to defeating professional Go game player in late 2015 [27]. 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-layer structure are called deep neural networks. Researchers in the 1980s and 1990s tried using stochastic gradient descent and back-propagation to train deep networks. Unfortunately, except for a few special architectures, deep neural network approaches did not succeed. The deep networks would learn, but very slowly to be of any practical use. Since 2006, a set of techniques has been developed that enable learning in deep neural networks. These techniques are based on stochastic gradient descent and back-propagation, but also introduce new ideas. These techniques have enabled much deeper (and larger) networks to be trained. It turns out that they perform far better on many problems than regular neural networks due to their ability to build a complex hierarchy of concepts. In this research, we conducted a preliminary investigation on automated selection and classification of diffraction images using a deep Convolutional Neural Network (CNN) called AlexNet [19]. We compare the accuracy and performance of the classification between SVM based and deep learning approaches.

2.5 The Software for Reconstructing the 3D Structure of a Cell

The special-purpose software was built for reconstructing 3D structure of a cell by processing its confocal image sections. The 3D structure of a cell is constructed using the recognized cell organelles in each confocal image section, which is 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 μm) 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. Next, the contours of segmented organelles between neighboring slices, and the 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 Fig. 3b and Fig. 3c shows a calculated diffraction image.

2.6 The Software for Modeling Light Scattering of a Cell

The ADDA software simulates light scattering using the realistic 3D morphology parameters reconstructed from the confocal images of cells [28]. DDA is a method to simulate light scattering from particles through calculating scattering and absorption of electromagnetic waves by particles of arbitrary geometry [15]. ADDA is a general implementation of DDA 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 voxel, orientation of the scatterer, definition of incident beam, and many others. ADDA can be configured for producing different outputs for different applications. In this study, we collect the Muller matrix from ADDA simulation to produce diffraction images using a ray-tracing technique [29]. Fig. 1c 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 enable the study of correlations between the 3D morphology parameters of a cell and the texture 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.

3 Feature Optimization and Validation

Different feature representations have been used for machine learning based classification of diffraction images of cells. For example, the frequency and the size of speckles of a diffraction images was used for classifying viable cell diffraction images and non-viable cell diffraction images

TABLE 1
SVM Classification with EFCS Selected GLCM 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

[21]. GLCM features of a diffraction image were also used for classifying cells [30]. In our recent work, multiple layers of image blocks of a diffraction image were used for deep learning of diffraction images [31].

The focus of this section is on feature optimization and validation of GLCM features for SVM learning. Feature optimization is the process of selecting a subset of optimal features from a set of features [23], [26]. An automated cell classification based on GLCM features was developed in Hu's previous work [30], [32]. However, the GLCM feature set often contains highly correlated features and creates difficulties such as computational intensiveness, slow learning, and in some cases decreased classification accuracy [23]. Empirical approaches are used to find a set of optimized GLCM features for the cell classification and 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 the confusion matrix.

3.1 Feature Optimization for Cell Classification

In the first experiment, the dataset contains 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 [26], which is one of the most commonly used filter-type feature selection algorithm. All the feature vectors computed in this experiment are labeled with cell types to enable supervised learning. Also, we need to find an optimal displacement d for GLCM and investigate the gray level of the diffraction images that would result in high cell classification accuracy [23].

EFCS selects uncorrelated features by analyzing the trends 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 are calculated using orientation at 0, 45, 90, and 135 degree for each image, respectively. 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 at 1, 2, 4, 8, 16 and 32 and gray levels at 8, 16, 32, 64, 128 and 256. This resulted in 36 combinations for each image. Features are normalized to values between 0 and 1. The GLCM features are categorized into three groups—Contrast, Uniformity, and Correlation [20]. Features from each group are plotted on a single graph for all the 600 images with the same displacement and gray level. Finally, uncorrelated features are obtained from each group for all 108 (i.e., 3 groups × 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 retained into the optimized feature set, which are $CON, IDM, VAR, ENT, DENT, COR, SA, IMC1$. The definition of each feature is described in [23] and [20]. The details of this experiment are discussed in Ding's previous work [23].

The CFS algorithm is 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 finally selected. These are the features that were selected by the highest number of times in all the combinations. The accuracy of cell classification of the 600 images using SVM based on EFCS feature set is slightly better than the one with the CFS selected features.

We used LIBSVM [33], an open source library for SVM, to conduct the classification of diffraction images based on GLCM features. The type of each cell is known in advance. In the training phase, feature vectors and their corresponding cell type labels are given to SVM. Next, the 10FCV method is used to check the classification accuracy. This validation 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). The 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. Using the 8 EFCS selected features, the SVM classification accuracy achieved for the classification of the 600 diffraction images is 91.16 percent, which is slightly better than what is achieved by using all the 17 features [23].

Table 1 shows the cell classification results with different configuration of gray level g and displacement d values. Based on these results, we conclude that the 8 features selected by EFCS are effective for SVM based cell classification. Also, when the gray level g is 64 and displacement d is 2, the accuracy of cell classification is the highest. Therefore, selecting appropriate gray 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 gray level 64 and displacement 2 are the optimal feature set for classifying diffraction images using SVM. However, this approach entails enormous computational costs. We processed a total 21,600 diffraction images and extracted 4,320,000 feature values for the feature selection experiment.

3.2 Feature Optimization for Image Selection

In the previous section, we noted that the selected 8 GLCM features can be used for effectively classifying cell types based on diffraction images using SVM. Guided by the feature optimization result generated in the previous section, we investigated how the feature selection would affect the accuracy of a different SVM classification. In this experiment, SVM is applied for classifying diffraction images of viable cells from those of ghost cell bodies and debris. We

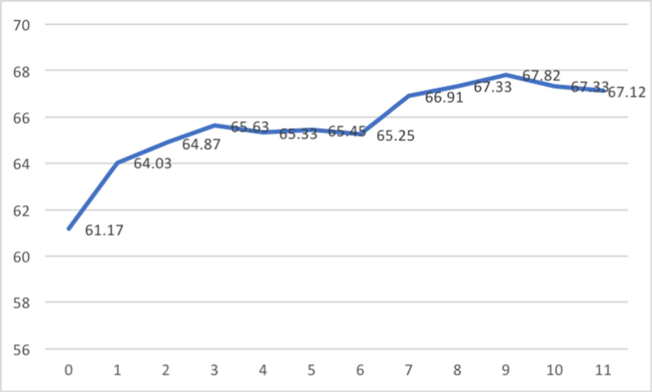


Fig. 4. An experiment result of feature selection.

selected 1,800 images for each of the three cell types, and then calculated the 17 GLCM features for each image with distance 2 and gray 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 percent. Then each time we removed one feature from the feature matrix but keep all the 8 features (e.g., $CON, IDM, VAR, ENT, DENT, COR, SA, IMC1$) and retrained the SVM classifier. We found that the accuracy of classification for all three classifiers slightly increased when some of the features were removed. Fig. 4 shows one of the experimental results, where the x axis represents the number of features that 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 dataset, the highest classification accuracy for classifying the viable cells using SVM was increased to 84.6 percent. The result further demonstrates that feature optimization is necessary for improving the classification accuracy.

4 Selection and Validation of Image Data

The diffraction images of cells taken using a p-DIFC may include abnormal images due to cell debris or small particles, and ghost cell bodies or aggregated spherical particles contained in the sample. The abnormal images decrease the accuracy of the cell classification [6]. If the sample size is small, it is feasible to manually remove 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 the whole data set that includes many abnormal images produced from non viable cells or debris. Different algorithms including SVM with GLCM features, SVM with image preprocessing, and deep learning with CNN are compared for their effectiveness in the image data selection.

4.1 The Data Set

Based on our previous experiment results, we know majority of abnormal diffraction images are generated from cell debris or small particles, and ghost cell bodies or aggregated spherical particles. A ghost cell body or aggregated spherical particle (simply called a fractured cell) normally produces strip patterns in its diffraction image, whereas a viable cell with intact structure (simply called a normal cell) usually generates speckle patterns, and the cell debris or a small particle (simply called debris) produces large diffuse speckle pattern [6]. Fig. 5 shows 3 diffraction images with different fringe patterns: Fig. 5a is a viable cell with the normal speckle pattern, Fig. 5b is a fractured cell with the strip pattern, and Fig. 5c is the debris with the large diffuse speckle pattern. The difference of the fringe patterns can be easily observed from their borderlines as shown in Fig. 6, extracted from the diffraction images using image processing algorithms. The data set includes 2,000 diffraction images for each of the three categories.

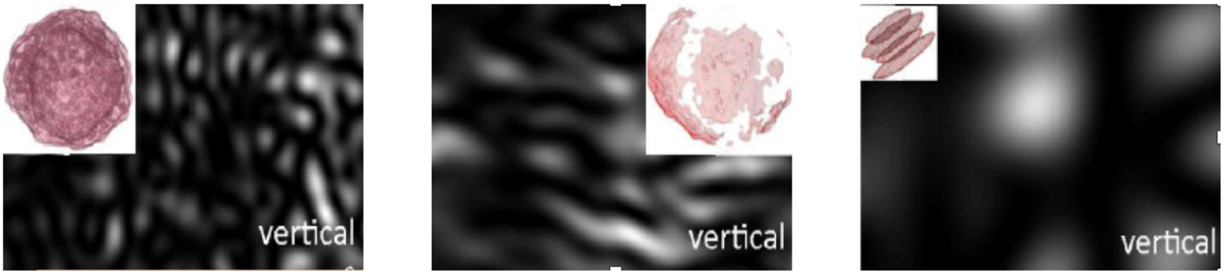
Based on the above observation, we developed a procedure that uses different algorithms to classify diffraction images into three categories based on their fringe patterns: normal cells, fractured cells, and debris.

4.2 An 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 the GLCM features of the images. We selected 2,000 diffraction images for each category, and then calculated the 17 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 percent, Therefore, a simple SVM based diffraction image selection is not good enough, an advanced technique is needed for improving the accuracy of the classification.

4.3 An Image Processing Based Data Selection

To improve the accuracy of the classification of diffraction images of normal cells, fractured cells and debris, images



(a)

(b)

(c)

Fig. 5. A diffraction image and its scatterer of (a) a viable cell, (b) a ghost cell body, and (c) the debris [6].

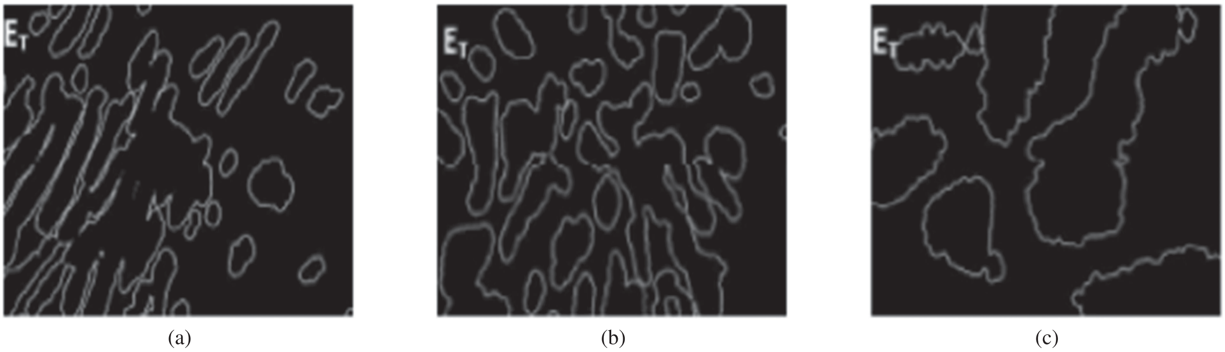


Fig. 6. The borderlines of (a) strip patterns, (b) normal speckles, and (c) large diffuse speckles in a diffraction image [6].

are preprocessed using advanced image processing algorithms. The image data selection procedure is comprised of four steps [6]. The first step is to find a set of borderline length parameters as shown in Fig. 6 for differentiating the strip pattern from the speckle pattern. Frequency histogram of each diffraction image for measuring the speckle size is calculated next. In the third step, the k-means clustering algorithm is applied to calibrate image data and separate images as strip and speckle patterns. The fourth step is to precisely classify the calibrated diffraction images into large diffuse speckles (i.e., debris) and normal speckles (i.e., normal cells) using SVM with GLCM features [6]. Compared to the simple SVM based approach, each of the three classifications enjoys over 90 percent accuracy.

4.4 A Deep Learning Based Image Data Selection

The classification based on SVM with preprocessed image data requires complex preprocessing including image processing and K-means clustering. That 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 viable cells and other particles. In this section, we introduce a deep learning approach for the automated image selection. The diffraction image dataset we used is still same as that used in the previous section. We used a deep learning framework called Caffe [34], and a deep learning model called AlexNet [19] to build the classifier. 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 the AlexNet classifier. In addition, AlexNet needs a much larger training dataset than an SVM does. The deep learning procedure for the image selection is summarized as follows:

First, generate a training dataset. We produce many small images at size 227×227 pixels through cropping image sections from a raw diffraction image. First, find the brightest 10-pixel diameter spot in a raw diffraction image, and choose the spot as the center and crop a 227×227 pixels image from the raw image. It is important to make sure that each cropped image is located within the original image. Next, choose a new center which is a shift 5 pixels in a direction from the center of the brightest 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. Different spots can be identified from an image to produce even more training data. Based on light scattering theory, a large portion of a diffraction image could contain enough information to represent the whole image, the appropriately cropped images should be good enough for training and testing the deep learning based classification. When we test the classification, any valid cropped image from a raw image is used for representing the whole image. However, we will experiment different approaches such as pooling 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 to address the size issue.

Second, each cropped diffraction image is labeled same as its raw image—$cells$ for normal cells, $debris$ for debris and small particles and $strip$ for fractured cells, and is placed into one of the three folders based on its label. In this experiment, the images in each folder are divided into 8 equivalent group. The first 6 groups are used as the training data, another group as the validation data, and the last group as testing data. The data folders of $cell$, $debris$, and $strip$ include 105,072, 121,344, and 99,216 cropped diffraction images, respectively.

Third, the training and testing is run with Caffe on NVidia GPU K40c, and the number of iteration of the training is set to 10,000. 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 percent, respectively. The confusion matrix of the classification is shown in Fig. 7. Compared to the SVM based classification, deep learning based data selection gives higher classification accuracy. However, deep learning needs large amount of training data, and it does not work on raw images directly. This could be a serious problem for other domain-specific images. For example, a cytopathology image is much larger and complex than a diffraction image, and it is extremely challenging to obtain large number of cytopathology images for deep learning. In that case, SVM based technique is still an alternative for automated data selection.

5 Metamorphic Testing of Scientific Software

It is difficult to know whether a reconstructed structure generated by the 3D reconstruction program represents the real 3D structure of a cell. Also, given an arbitrary input to ADDA program, 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

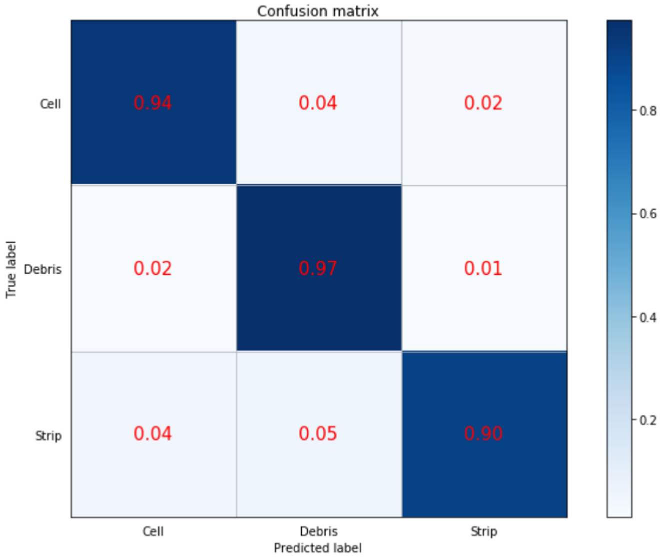


Fig. 7. The confusion matrix for a deep learning algorithm for image classification.

verify these products and use an iterative approach for developing MRs.

5.1 Metamorphic Testing

Metamorphic testing is a promising technique for solving oracle problems in non-testable programs. It has been applied to different domains such as bioinformatics, machine learning, 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 MR related tests, rather than checking the correctness of each individual output [7], [10]. If a violation of an MR is found, then the system under test (SUT) must have defects [7]. Specifically, metamorphic testing creates tests according to an MR and verifies the predictable relation among the actual outputs of the related tests.

Let $f(x)$ be an output of test x in program f and t be a transformation function for an MR. Given test x (called a source test), one can create a new metamorphic test $t(x, f(x))$ (called a follow-up test) by applying function t to test x. The transformation allows testers to predict the relation between the outputs of test x and its transformed test $t(x, f(x))$ according to the MR [7]. 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. It is important to evaluate the quality of MRs and their tests. 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.

5.2 Iterative Metamorphic Testing

The iterative metamorphic testing consists of three major steps: develop initial MRs and tests, test execution and evaluation, and refine MRs.

Develop Initial MRs and Tests: Based on the domain knowledge of the SUT and general framework of metamorphic testing [35], one can develop a set of initial MRs. The source tests are produced using general test generation approaches such as combinatorial testing, random testing and category-choice framework, and then each source test is transformed into a set of follow-up tests according to an MR. A source test together with its follow-up test form a test of the MR. The newly added test can be used for producing additional tests based on MRs.

Test Execution and Evaluation: The SUT is executed with every test, but outputs of the source test and its paired follow-up test are verified by their related MR. As soon as the SUT passes all tests, the testing is evaluated for test adequacy. We evaluate the testing with program coverage criteria, mutation testing, and mutated tests. A mutated test is a paired source and follow-up test whose outputs would violate their related MR. Mutated tests are used to check each MR can differentiate a positive test from a 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 a test from the mutated program is different from the original program.

Refine 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, 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 [36]. The ultimate goal of an MR refinement is to develop oracles that can verify individual tests.

5.3 Testing the 3D Structure Reconstruction Software

The most difficult part in the 3D structure reconstruction software is to correctly build the 3D structures of mitochondria in a cell. 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 wrong connection will result in a wrong 3D structure. However, 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 while its reconstructed structure is produced. In this case, iterative metamorphic testing is an ideal way for rigorously verifying the 3D structure reconstruction function. We test the function following the three general steps.

Develop Initial MRs and Tests: Fig. 8 shows a sample input to the program and its corresponding output, where (a) is a sample confocal image section of a cell, and (b) is a sectional view of the 3D reconstructed cell. We created 5 initial MRs as listed below. The details of the MRs were reported in previous work [37], but we use them here 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, defines the relation between the reconstructed 3D structure and the size of selected mitochondria in the image sections.

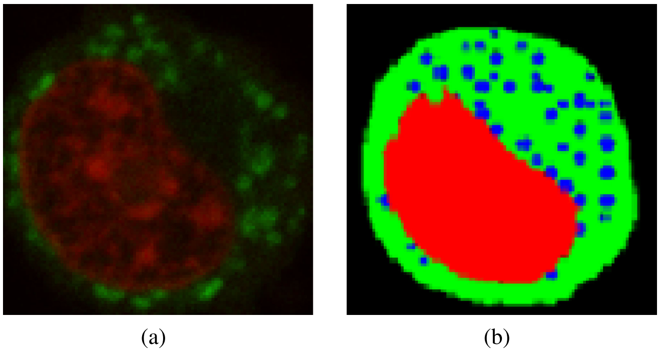


Fig. 8. (a) An example of a confocal image and (b) its processed image.

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

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

MR5: Locations, 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 according to each MR. For example, according to MR1, an artificial mitochondrion is added to one or more image sections in a stack of original image sections such as S using Matlab to produce MR1 related image sections T as the follow-up test. MR related source test and follow-up test are executed one by one and their output 3D reconstructured structures can be compared to determine whether the new added mitochondrion is appropriately built.

Evaluation of MRs and Tests: Test adequacy coverage criteria were evaluated for function coverage, statement coverage, definition-use pair coverage. The coverage difference between a source test and its follow-up test help us detected a defect in the original program [37]. Mutation test was also conducted for evaluating the quality of the MRs and their tests.

Refine MRs: MRs *Inclusive/Exclusive* and *Multiplicative* can be further refined to determine the exact change in mitochondria's volume. For example, if an artificial mitochondrion was added to the confocal image sections of the source test, the volume of the new added mitochondrion can be calculated based on its 3D model using Matlab. Then the volume difference between the reconstructed 3D structures of the source test and the follow-up one should be only the new added mitochondrion volume. 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 is the one as expected, and volume of mitochondria in the reconstructed 3D is increased as expected. But the increment of the volume in the follow-up test is not same as the volume of the new added mitochondrion, which would flag an error in the reconstruction function. Therefore, MR2 can be refined as MR6 defined as follows.

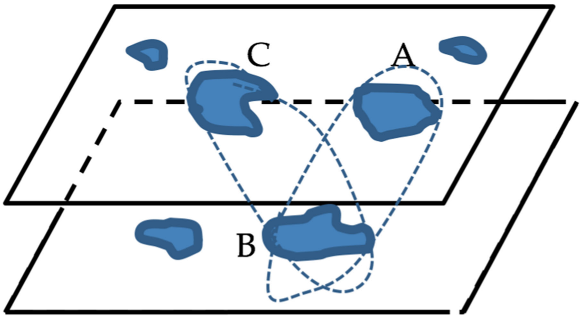


Fig. 9. An illustration of a possible reconstruction error.

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 on removing or resizing a mitochondrion.

5.4 Testing ADDA

ADDA has been extensively tested with special cases and other modeling approaches such as Mie theory [15], [28]. Fig. 10 is a comparison of the simulation results of Mie theory and ADDA [28], which shows ADDA and Mie theory produce nearly identical S_{11} results for a sphere scatterer. However, Mie theory can only calculate a regular scatterer, but ADDA can simulate a scatterer in any shape. Therefore, it is necessary to test ADDA for simulating any shape of scatterers using a different approach. A different implementation of DDA for testing ADDA is not available. Therefore, iterative metamorphic testing was used for testing ADDA. The purpose of testing ADDA is not to verify the correctness of its implementation. 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. Preliminary results

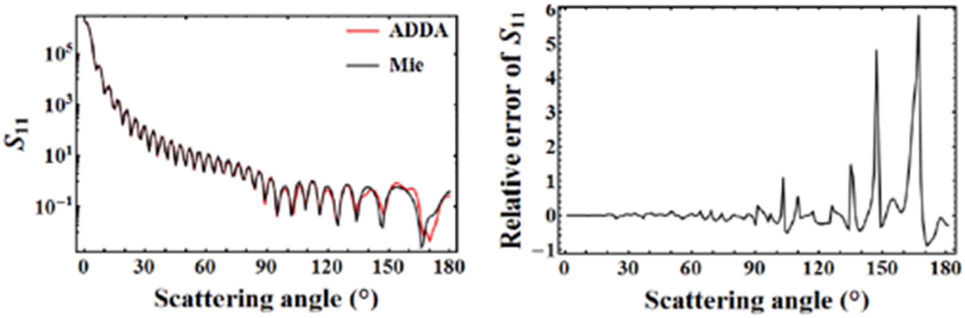


Fig. 10. A comparison between Mie theory and ADDA [28].

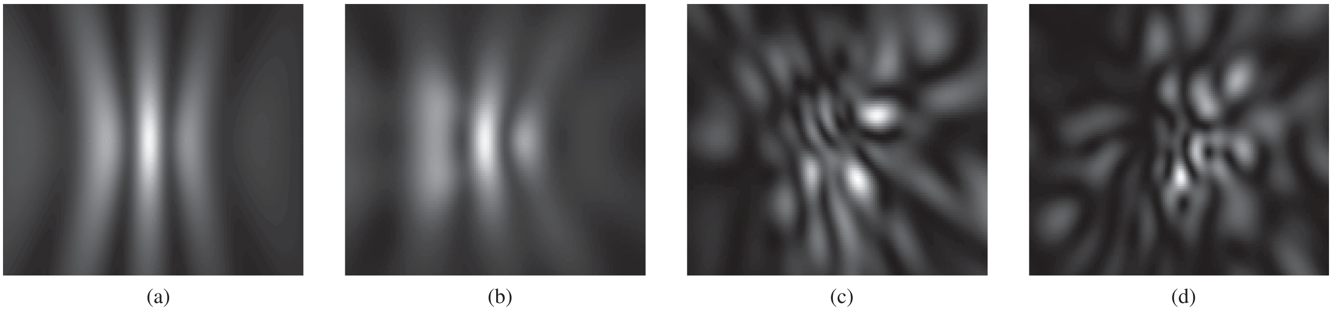


Fig. 11. The change of fringe patterns of diffraction images calculated with different configurations, (a) a sphere, (b) an ellipsoid, (c) and (d) are cells.

on testing ADDA using metamorphic testing were reported in Ding and Hu's previous work [8], [38], but we conducted a comprehensive and more rigorous testing in this research.

Develop Initial MRs and Tests. It is infeasible to find an oracle for checking the correctness of the output of an ADDA input since one input could produce a different output each time. The output could include thousands of parameters, so that the number of possible correct outputs of an input could be enormous. Therefore, we define MRs on the relation between an ADDA input and the fringe pattern of the diffraction image that is produced from an ADDA simulation output. Although an ADDA input would produce different output each time, the fringe pattern of the diffraction images generated from any output are supposed to be the same. Each input parameter such as the shape, size, orientation, and refractive index of a scatterer is a candidate for defining MRs, which define the relation between the change of one parameter and the change of the fringe pattern in the output diffraction image.

MR7: When the size, shape, orientation, refractive index or internal structure of a scatterer is changed, the fringe 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 fringe 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 fringe pattern can be compared manually by eyes or is compared by 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 fringe pattern of the ellipsoid is irregular compared to the fringe pattern of the corresponding sphere. This is shown in Figs. 11a and 11b, where (a) is a sphere, whose x, y and z axes are 5 $\mu\mathrm{m}$, and (b) is an ellipsoid, whose x and z axes are 5 $\mu\mathrm{m}$, and y is 7 $\mu\mathrm{m}$. However, due to the complexity of ADDA, it is infeasible to build an exact relation between the change of a parameter and the fringe pattern of the diffraction image in general. For example, Figs. 11c and 11d are two ADDA calculated diffraction images from the same reconstructed 3D morphology parameters of a cell but with different orientations. It impossible to know the precise relation between the orientation and the fringe pattern of the calculated diffraction images except the "difference". The relation "difference" is too broad to test ADDA adequately. Additional MRs are needed. 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 a sphere becomes larger, the texture bright lines in the diffraction image become slimmer.

Fig. 12 shows the ADDA calculated diffraction images of sphere scatterers with diameters in 5, 7, 9, and 12 μm, respectively. The output examples satisfy MR8. Furthermore, we define an MR based on the relation between the fringe pattern and a sphere scatterer with some portions removed. For example, we can check how the fringe pattern is changed when a sphere scatterer is cut into half.

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

We first tested a sphere scatterer with diameter 3 μm, 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 set as (0,0,0) [15]. We also conducted ADDA simulations with the same sphere that had its top quarter that faces the incident light beam removed, and the top outside 1/8 part

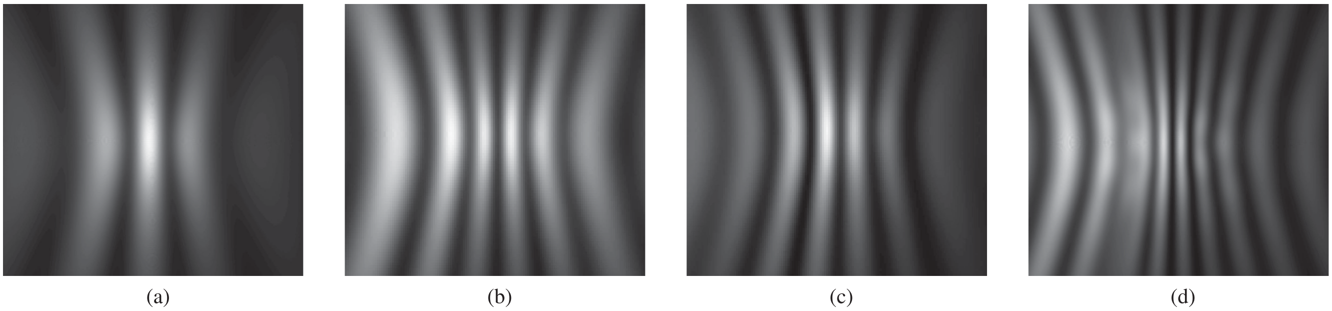


Fig. 12. Fringe patterns of ADDA calculated diffraction images of sphere scatterers with different diameters: (a) 5 μm, (b) 7 μm, (c) 9 μm, and (d) 12 μm.

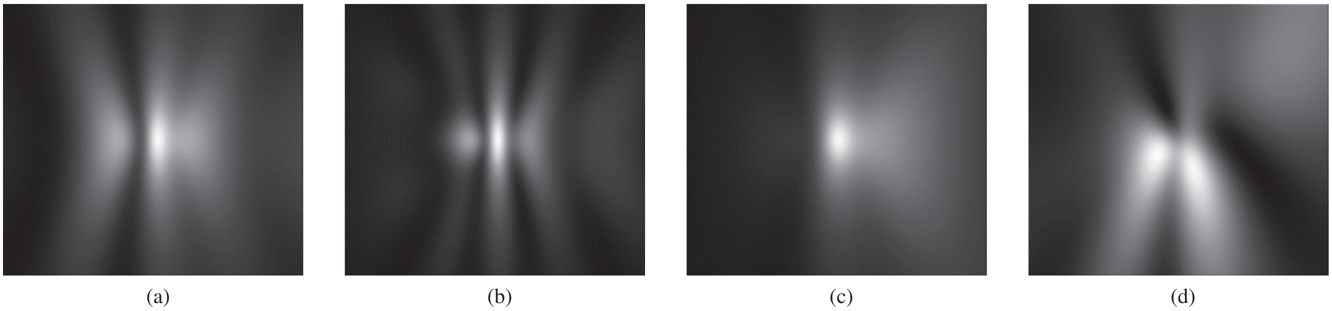


Fig. 13. The change of fringe patterns of ADDA calculated diffraction images of a sphere scatterer having partial cut: (a) no cut, (b) 1/2 cut, (c) 1/4 cut, and (d) 1/8 cut.

of the sphere was removed. The simulation results are shown in Fig. 13. It is easy to find that the symmetry property of the fringe pattern in the diffraction image of the sphere is lost as expected when some part of it is removed. Based on the same idea, we can check the change of the fringe pattern when a sphere scatterer is added to the other.

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

We first calculated a diffraction image for a 5 μm diameter sphere using ADDA, and then we added one identical sphere to form a bisphere scatterer. The two spheres are separated by 1.5 μm and they are aligned along x axis. The orientation of the bishphere is set as (0, 0, 0), which is same as the single sphere. Figs. 14a and 14b show the ADDA calculated diffraction images for a sphere scatterer and a bisphere scatterer, respectively. The fringe pattern in the diffraction image of the bisphere scatterer clearly shows the two spheres in the scatterer. If we changed the orientation of the bisphere from (0,0,0) to (0, 270, 0) and (90, 90, 0), then the fringe patterns of their diffraction images are changed as shown in Figs. 14c and 14d. The results satisfy MR10 and MR7.

In order to create tests that can cover as many cases as possible, the combinatorial testing method is used in this research. 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 *{3 μm, 5 μm, . . . , 16 μm}*, shapes are *{sphere, ellipsoid, bisphere, prism, egg, cylinder, capsule, box, coated, cell1, cell2, . . .}*, orientations are {(0, 0, 0), (10, 90, 0), (270, 0, 0), . . .}, and refractive index values are *{1.0, . . . 1.5}*. Using pairwise testing, one can create many tests, and then select the valid tests as the source tests. Next, the source tests are used to create follow-up tests for each MR. Using this method, many execution scenarios of ADDA can be tested and their results can be systematically verified using the MRs.

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 percent statements of ADDA program. Mutation testing was conducted to check the effectiveness of the tests but it was applied only to one critical module in ADDA. Instead of testing the software 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. This is because that these two operators achieve an 80 percent reduction in the number of mutants and only 5 percent reduction in the mutation score as shown in the empirical results reported by Wong and Mathur [39], [40]. 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 that a slight change in ADDA may cause a catastrophic error in the calculation. Therefore, creating powerful mutants—ones that do not crash the software or produce trivial errors—for testing ADDA is difficult.

Refine MRs. Since scatterers in regular shapes such as sphere have been extensively tested [15], [28], we are more interested in scatterers in irregular shapes such as cells. ADDA software is used to investigate the correlation between the 3D morphology of a cell and the fringe pattern of its diffraction image. This is essential to understand how

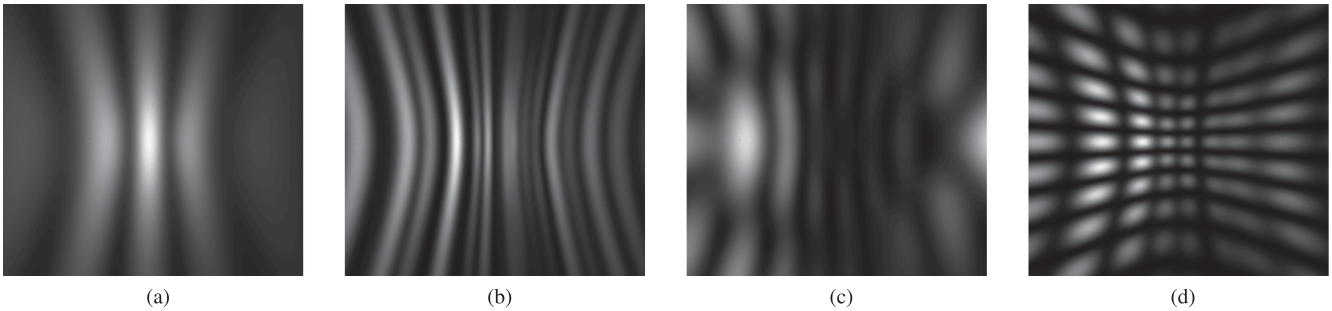
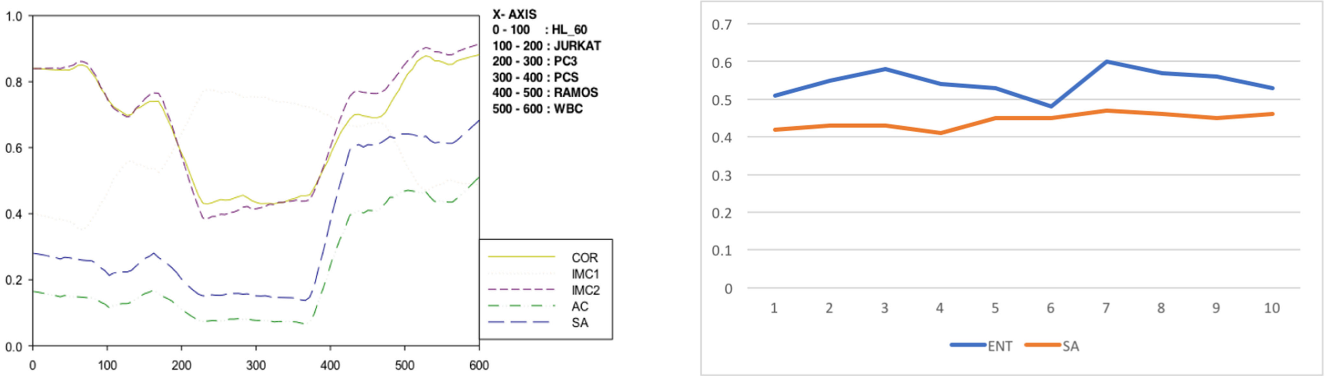


Fig. 14. The change of fringe patterns of ADDA calculated diffraction images of a sphere and bisphere scatterers at different orientations: (a) Single sphere, (b) bisphere at (0, 0, 0), (c) bisphere at (0, 270, 0), and (d) bisphere at (90, 90, 0).



(a) (b)

Fig. 15. (a) Values of selected GLCM features of measured diffraction images, it has 600 images for 6 types of cells [23]. (b) Values of two GLCM features of 10 ADDA calculated diffraction images from the same type of cells.

the diffraction images can be used for the cell classification. Therefore, we can define MRs based on the classification of diffraction images that are produced by ADDA. The first two MRs are defined on the correlation of the GLCM features among a group of related diffraction images, and the last two MRs are defined through refining the previous MRs on the classifications of ADDA calculated diffraction images.

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

The experiment process is summarized as follows: (1) Select 100 p-DIFC measured diffraction images from one cell type. (2) Calculate the 17 GLCM features for each image, and plot the feature values and their corresponding image IDs in a 2D diagram. (3) Check the relation of each feature among the images. (4) Select another 100 cells whose cell type are same to the calculated one. Then take the confocal image sections for each cell using a confocal microscope. (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 diffraction images using ADDA with the 3D morphology parameters. (7) Calculate the GLCM features for each calculated diffraction image and plot the feature values in a two-dimensional diagram. (8) Compare the feature relation between the measured images and the calculated ones. If a 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 required. Although the ADDA calculation and p-DIFC measurement were conducted on the same type of cells, their GLCM feature values of the diffraction images could be substantially different. This is because the unknown of the exact value of the refractive index of the nucleus in a cell. However, it is not a problem since we only care about the consistency between the relation of a GLCM feature among ADDA calculated diffraction images and the relation of the GLCM feature among the corresponding p-DIFC measured diffraction images. Preliminary experimental results shown in Fig. 15 support MR11. However, the precise relation among the same type of cell images are not easily detected based on just one GLCM feature. The comparison of the relation of the measured and calculated images is vaguely defined. More advanced MRs are needed for adequately testing ADDA.

Based on above discussion, we check how the fringe pattern of an ADDA calculated diffraction image is correlated to the change of cell morphology parameters. We know that when the refractive index or the size of an intracellular organelle, or the orientation of the scatterer is changed, the fringe pattern of the ADDA calculated diffraction image will change. We conducted an experiment to validate ADDA via checking the relation between cell morphology parameters and the fringe pattern of ADDA calculated diffraction images. First, the 3D structure of a cell is constructed based on its confocal image sections using the 3D reconstruction software. Next, a series of morphology parameters are built by changing the value of one parameter each time, such as resizing the nucleus of the cell or changing the refractive index values of a nucleus [37]. The series of morphology parameters and the orientation are input to ADDA for producing a series of diffraction images. The GLCM feature values of each ADDA calculated diffraction image are calculated, and finally the values of GLCM features of the calculated diffraction images and the values of the corresponding morphology parameter are plotted in a 2D diagram to check their correlation. For example, a viable cell with intact structure would produce a diffraction image with normal speckle patterns, but a ghost cell body would produce a diffraction image with strip patterns due to its high degree of symmetry in its structure. Therefore, one can create a series of 3D structures of a cell through resizing the nucleus in a viable cell. Then the change of the fringe pattern defined in GLCM features in the calculated diffraction images should be correlated to the change of the nucleus structure in the morphology parameter. Previous experiments have shown the correlation exists between the GLCM features of p-DIFC measured diffraction images of cells and their morphology parameters [6], [23]. This observation helps us to develop MR7′ as follows, which is a refined version of MR7.

MR7′: The fringe pattern of the diffraction image of a viable cell, a ghost cell body and a debris particle is different. Specifically, the fringe pattern of the diffraction image of a viable cell is a group of small speckles, the fringe pattern of the diffraction image of a ghost cell body is a group of stripes, and the fringe pattern of the

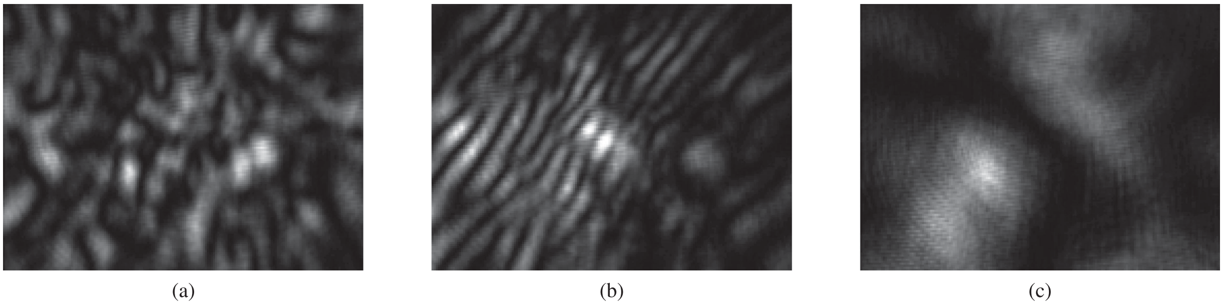

(a) (b) (c)

Fig. 16. A p-DIFC measured diffraction image of (a) a viable cell, (b) a ghost cell body, and (c) a debris particle.

diffraction image of a debris particle is a group of large diffuse speckles.

The diffraction images shown in Fig. 5 are ADDA calculated diffraction images. It is easy to find the fringe patterns of the three types of images are consistent to the fringe patterns of the corresponding p-DIFC measured images shown in Fig. 16.

Finally, 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 modeled 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 using ADDA calculated diffraction images.

MR12: The ADDA calculated diffraction images can be classified by the shapes of their 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—8 different sizes and 25 different orientations of the scatterer. Since the refractive index would substantially impact the fringe pattern of a diffraction image, all ADDA calculations are assigned with the same refractive index of 1.06. A total of 600 images were produced for three shapes: sphere, bisphere, and ellipsoid. Each image is processed for the GLCM feature values and labeled with the shape type of the scatterer. The values of the eight selected GLCM features of a diffraction image and its labeled shape type form a feature vector. The SVM classifier is trained and tested with ADDA calculated diffraction images using LIBSVM [33]. The classification accuracy from 10FCV for each shape of scatterers is 100 percent. The experiment results indicate that ADDA is 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. Metamorphic testing can be conducted next via checking the MR among the corresponding ADDA calculated diffraction images.

MR13: An ADDA calculated cell diffraction image can be classified according to its cell type.

Per the experiment results discussed in Section 4, we know diffraction images of cells can be used to accurately classify the viable cells, debris particles, and ghost cell bodies. 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. They have the same patterns as those images taken by p-DIFC shown in Fig. 16. Using the SVM based classification approach discussed in Section 4, it is not difficult to check whether the ADDA calculated diffraction images can be correctly classified.

Fringe patterns derived from the GLCM features in p-DIFC measured diffraction images have been successfully used for classifying cell types [21], [30], [32]. Combining the test results of MR12, ADDA calculated diffraction images of cells should be sufficient for classifying cells. We simulated 25 orientations for the 3D morphology parameters of each cell using ADDA. Each diffraction image is processed for the 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 an SVM classifier. The 10FCV is used for checking the classification accuracy. Our preliminary results show that ADDA calculated diffraction images can be used for classifying select cell types [6]. Recently, we tested the classification of 30 ADDA calculated diffraction images including 10 viable cells, 10 ghost cell bodies and 10 debris particles using the deep learning classifier discussed in Section 4.4. The classification accuracy for each category was 100 percent, which strongly suggested that the morphology properties of the scatterers were appropriately modeled in ADDA. However, there are many different types of cells, and some of them are only slightly different in 3D structures. Whether diffraction image-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 we still achieve high accuracy in classifying different types of cells that are only slightly different in morphology parameters, it would be safe to conclude that ADDA is well implemented for simulating the light scattering of cells.

5.5 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 examples of software that do not have test oracles, which are called non-testable programs. Although metamorphic testing can be used for testing the non-testable 3D structure reconstruction and ADDA software, the effectiveness of the testing is highly dependent on the quality of MRs. Therefore, MRs should be rigorously evaluated during the testing, and the initial MRs should be iteratively refined based on test results. We conducted two empirical studies on verification and validation of non-testable software using the iterative metamorphic testing approach. The same approach can be used for testing any other software components including regular software in a big data application. 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 ADDA testing. If more data, the one generated by more scatterers that have different morphological structures and more scatterers from different types of cells, can pass the MRs 7 through 13, it would be safe to conclude that ADDA has been sufficiently validated.

6 Related Work

Quality assurance of big data systems includes quality assurance of datasets, data analytics algorithms, and big data applications. In this paper, we proposed a framework for the verification and validation of big data systems and illustrated the process with a massive biomedical image data system called CMA. The framework includes scientific software testing, feature selection and validation of machine learning algorithms, and automated data selection and validation. In this section, we discuss related work on these 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 classification. The quality attributes of big data systems include availability, usability, reliability, and relevance. Furthermore, each attribute includes sub-attributes: availability encompasses accessibility, timeliness, and authorization; usability includes documentation, metadata, structure, readability and credibility; accuracy, integrity, completeness, consistency and auditability are associated with reliability [1], [41]. Gao, Xie and Tao have provided an overview of these issues, discussed challenges, and list tools for validation and quality assurance of big data systems [42]. They define 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, domain-specific techniques are typically needed. For example, data quality assurance in the healthcare domain is quite different from those in biomedical sciences or banking and financial services.

Web is one of the primary sources of big data, but the trustworthiness of the web sources has to be evaluated. There are many investigations on evaluating the veracity of web sources using hyperlinks, browsing history, or the factual information provided by the source [43]. Furthermore, some evaluations are based on the relationship between web sources and their information [44]. Finding duplicates in data gathered from different sources is also an important quality assurance task in big data. Machine learning algorithms such as Gradient Boosted Decision Tree (GBDT) have been used for detecting duplicates [45]. Data filtering is an approach for quality assurance through removing bad data from data sources. For example, Apache Samza [46], which is a distributed stream processing framework, has been adopted for detecting and removing bad data. Nobles et al. have conducted an evaluation of the completeness and availability of electronic health record data. They identify undesirable data in datasets using machine learning algorithms such as SVM or deep learning. The undesired data could be incorrectly labeled in training data, which is known as class label noise. This could reduce the performance of machine learning algorithms. To address these problems, one can improve the machine learning algorithm to handle poor data or improve the quality of the data through filtering [47]. Due to the massive scale of big data, automated filtering using machine learning is a preferred choice. In this paper, we proposed a data filtering technique based on automated data classification. The other quality assurance techniques can be integrated into our framework and vice versa.

Feature selection is a central issue in machine learning for identifying a set of features to build a classifier for a domain-specific task [26]. The process is to reduce irrelevant, redundant and noisy features to improve both learning performance and prediction accuracy. Hall reported a feature selection algorithm called CFS to select features based on the correlation of the features and the class they would predict [26]. CFS has been used for cross checking the feature selection for SVM based classification of diffraction images [23]. In this paper, we introduced an easy to use and more practical approach for feature selection. Our experimentally based feature selection would produce a slightly better feature set in term of the accuracy of the classification of diffraction images [23]. How the feature selection would impact the classification accuracy or machine learning cost has been investigated [48], [49]. More advanced feature selection approaches such as the one discussed in [50] 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. Our current work on resizing diffraction images through random sampling or down-sampling pooling is promising and will pave the way for using deep learning for classification of diffraction images.

Testing of scientific software adequately is a grand challenge problem. One of the greatest challenges occurs due to the oracle problem [9]. Many different approaches have been proposed to address the oracle problem including testing with special cases, experimental results, different implementations, and analysis of formal models of the software [9]. However, none of these techniques can adequately test the scientific software that is affected by the oracle problem. Metamorphic testing is a promising technique to address the problem though developing oracles based on MRs [7], [10]. Metamorphic testing was first proposed by Chen et al. [7] for testing non-testable systems. It has been applied to several domains such as bioinformatics, machine learning, and online service systems. An empirical study has been conducted to show the fault-detection capability of metamorphic relations [51]. A recent application of metamorphic testing to validate compliers has found several hundred bugs in widely used C/C++ compilers [52]. Metamorphic testing has been applied for testing a large NASA image database system [53]. Also, it has been successfully used for the assessment of the quality of search engines including Google, Bing, and Baidu [54]. As noted earlier, the quality of metamorphic testing is highly dependent on the quality of the MRs.

Knewala, Bieman and Ben-Hur recently reported a result on the development of MRs for scientific software using a machine learning approach which is integrated with data flow and control flow information [36]. In our research, we used test evaluation and test results for refining initially created MRs and iteratively developing new MRs. Generation of adequate tests in metamorphic testing is challenging due to the complexity of data types and a large number of input parameters in the SUT [9]. Combinatorial techniques [55] used for testing CMA are powerful tools for generating tests for metamorphic testing.

7 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 are introduced to automate the data selection process. Also, an experimentally based approach is proposed for feature optimization to improve the accuracy of machine learning based classification. An iterative metamorphic testing is used for validating the scientific software in CMA, and machine learning algorithms are used for developing and refining MRs. Machine learning algorithms are evaluated through cross validation and confusion matrix. The framework addresses the most important issues of verification and validation in big data. Furthermore, it can also be used for verification and validation of any big data system in a systematic and rigorous way.

Acknowledgments

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 their assistance with the experiments. This research is supported in part by grants #1262933 and #1560037 from the National Science Foundation. We gratefully acknowledge NVIDIA Corporation for their donation of the Tesla K40 GPU, which is used for conducting the research reported in this paper.

References

[1] V. Gudivada, R. Raeza-Yates, and V. Raghavan, “Big data: Promises and problems,” *IEEE Comput.*, vol. 48, no. 3, pp. 20–23, Mar. 2015.

[2] Y. Bengio, “Learning deep architectures for AI,” *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[3] Apache, “Hadoop,” 2016. [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 Comput.*, vol. 49, no. 4, pp. 31–42, Apr. 2016.

[5] D. Rao, V. N. Gudivada, and V. V. Raghavan, “Data quality issues in big data,” in *Proc. IEEE Int. Conf. Big Data Workshop Data Qual*, Oct. 2015, pp. 2654–2660.

[6] J. Zhang, 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.

[7] T. Y. Chen, S. C. Cheung, and S. Yiu, “Metamorphic testing: A new approach for generating next test cases,” Dept. Comput. Sci., Hong Kong Univ. Sci. Technol., Tech. Rep. HKUST-CS98–01, 1998.

[8] J. Ding, D. Zhang, and X. Hu, “An application of metamorphic testing for testing scientific software,” presented at the *1st Int. Workshop Metamorphic Testing ICSE*, Austin, TX, USA, May 2016.

[9] U. Kanewala and J. M. Bieman, “Testing scientific software: A systematic literature review,” *Inf. Softw. Technol.*, vol. 56, no. 10, pp. 1219–1232, 2014.

[10] S. Segura, G. Fraser, A. Sanchez, and A. Ruiz-Cortés, “A survey on metamorphic testing,” *IEEE Trans. Softw. Eng.*, vol. 42, no. 9, pp. 805–824, Sep. 2016.

[11] R. Maximilian and T. Poggio, “Models of object recognition,” *Nature Neuroscience*, vol. 3, pp. 1199–1204, 2000.

[12] K. Jacobs, J. Lu, and X. Hu, “Development of a diffraction imaging flow cytometer,” *Opt. Lett.*, vol. 34, no. 19, pp. 2985–2987, 2009.

[13] Mongodb, 2016. [Online]. Available: https://www.mongodb.com/

[14] Mongochef, 2016. [Online]. Available: http://3t.io/mongochef/

[15] M. Yurkin and A. Hoekstra, “User manual for the discrete dipole approximation code ADDA 1.3b4,” 2014. [Online]. Available: https://github.com/adda-team/adda/tree/master/doc

[16] ADDA project, 2016. [Online]. Available: https://github.com/adda-team/adda

[17] C. Hsu, C.-C. Chang, and C.-J. Lin, “A practical guide to support vector classification,” Nat. Taiwan Univ., Taipei, Taiwan, 2003, [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[20] R. Haralick, “On a texture-context feature extraction algorithm for remotely sensed imagery,” in *Proc. IEEE Comput. Soc. Conf. Decision Control*, 1971, pp. 650–657.

[21] K. Dong, et al., “Label-free classification of cultured cells through diffraction imaging,” *Biomed. Opt. Express*, vol. 2, no. 6, pp. 1717–1726, 2011.

[22] R. M. Haralick, K. Shanmugan, and I. H. Dinstein, “Textural features for image classification,” *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[23] S. K. Thati, J. Ding, D. Zhang, and X. Hu, “Feature selection and analysis of diffraction images,” presented at the *4th IEEE Int. Workshop Inf. Assurance*, Vancouver, Canada, 2015.

[24] J. Dixon and J. Ding, “An empirical study of parallel solution for GLCM calculation of diffraction images,” *38th Annu. Intl. Conf. IEEE Eng. Med. Biol. Soc.*, Orlando, FL, Aug. 2016.

[25] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2012.

[26] M. A. Hall, “Correlation-based feature selection for machine learning,” PhD dissertation, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 1999.

[27] E. Gibney, “Google AI algorithm masters ancient game of go,” *Nature*, vol. 529, pp. 445–446, Jan. 2016.

[28] M. Moran, “Correlating the morphological and light scattering properties of biological cells,” PhD dissertation, Dept. Dept. Phys., East Carolina Univ., Greenville, NC, USA, 2013.

[29] 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. 31568–31574, 2014.

[30] X. Yang, 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, pp. 2172–2183, 2014.

[31] M. Zhang, “A deep learning based classification of large scale biomedical images,” ProjectReport-MS-CS-17-0011, East Carolina University, Greenville, NC, Nov. 2016.

[32] Y. Feng, 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.

[33] C.-C. Chang and C.-J. Lin, "LIBSVM," 2016. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[34] Caffe project, 2016. [Online]. Available: http://caffe.berkeleyvision.org/

[35] J. Mayer and R. Guderlei, "An empirical study on the selection of good metamorphic relations," in *Proc. 30th Annu. Int. Comput. Softw. Appl. Conf.*, 2006, pp. 475–484.

[36] U. Kanewala, J. M. Bieman, and A. Ben-Hur, "Predicting metamorphic relations for testing scientific software: A machine learning approach using graph kernels," *J. Softw. Testing Verification Rel.*, vol. 26, no. 3, pp. 245–269, 2015.

[37] J. Ding, T. Wu, J. Q. Lu, and X. Hu, "Self-checked metamorphic testing of an image processing program," presented at the *4th IEEE Int. Conf. Secur. Softw. Integr. Rel. Improvement*, Singapore, 2010.

[38] J. Ding, D. Zhang, and X. Hu, "A framework for ensuring the quality of a big data service," presented at the *13th IEEE Int. Conf. Services Comput.*, San Francisco, CA, 2016.

[39] W. E. Wong and A. Mathur, "Reducing the cost of mutation testing: An experimental study," *J. Syst. Softw.*, vol. 31, no. 3, pp. 185–196, 1995.

[40] 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, Sep./Oct. 2011.

[41] L. Cai and Y. Zhu, "The challenges of data quality and data quality assessment in the big data era," *Data Sci. J.*, vol. 14, no. 2, pp. 1–10, 2015.

[42] J. Gao, C. Xie, and C. Tao, "Big data validation and quality assurance-issuses, challenges, and needs," in *Proc. IEEE Symp. Service-Oriented Syst. Eng.*, Mar. 2016, pp. 433–441.

[43] X. Dong et al., "Knowledge-based trust: Estimating the trustworthiness of web sources," *Proc. VLDB Endowment*, vol. 8, no. 9, pp. 938–949, May 2015.

[44] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 6, pp. 796–808, Jun. 2008.

[45] C. H. Wu and Y. Song, "Robust and distributed web-scale near-dup document conflation in Microsoft academic service," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2015, pp. 2606–2611.

[46] Apache samza, 2016. [Online]. Available: http://samza.apache.org/

[47] J. A. Saez, B. Krawczyk, and M. Wozniak, "On the influence of class noise in medical data classification: Treatment using noise filtering methods," *Appl. Artif. Intell.*, vol. 30, no. 6, pp. 590–609, Jul. 2016.

[48] M. Yousef, D. S. D. Müşerref, W. Khalifa, and J. Allmer, "Feature selection has a large impact on one-class classification accuracy for microRNAs in plants," *Advances Bioinf.*, vol. 2016, 2016, Art. no. 5670851.

[49] F. Min, Q. Hu, and W. Zhu, "Feature selection with test cost constraint," *Int. J. Approximate Reasoning*, vol. 55, no. 1, pp. 167–179, 2014.

[50] 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," *Mach. Learn.*, vol. 101, no. 1–3, pp. 163–186, 2015.

[51] H. Liu, F.-C. Kuo, D. Towey, and T. Chen, "How effectively does metamorphic testing alleviate the oracle problem?" *IEEE Trans. Softw. Eng.*, vol. 40, no. 1, pp. 4–22, Jan. 2014.

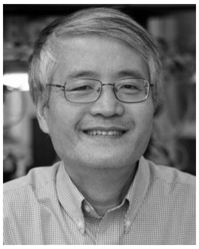
[52] V. Le, M. Afshari, and Z. Su, "Compiler validation via equivalence modulo inputs," in *Proc. 35th ACM SIGPLAN Conf. Program. Language Des. Implementation*, 2014, pp. 216–226.

[53] M. Lindvall, D. Ganesan, R. Árdal, and R. E. Wiegand, "Metamorphic model-based testing applied on NASA DAT: An experience report," in *Proc. 37th Int. Conf. Softw. Eng.*, May 2015, vol. 2, pp. 129–138.

[54] Z. Zhou, S. Xiang, and T. Chen, "Metamorphic testing for software quality assessment: A study of search engines," *IEEE Trans. Softw. Eng.*, vol. 42, no. 3, pp. 264–284, Mar. 2016.

[55] C. Nie and H. Leung, "A survey of combinatorial testing," *ACM Comput. Survey*, vol. 43, no. 2, pp. 1–29, Feb. 2011.

Junhua Ding received the BS degree in computer science from China University of Geosciences, in 1994, the MS degree in computer science from Nanjing University, in 1997, and the PhD degree in computer science from Florida International University, in 2004. He is an associate professor of computer science and university scholar with East Carolina University, where he has been since 2007. During 2000-2006, he was a software engineer with Beckman Coulter. From 2006 to 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 analytics. He is a member of the IEEE.



Xin-Hua Hu received the BS and MS degrees from Nankai University, Tianjin, China, in 1982 and 1985, respectively, the MS degree in physics from Indiana University, in 1986, and the PhD degree in physics from the University of California at Irvine, in 1991. He joined the physics faculty in 1995 and is currently a professor with East Carolina University. His main research interests relate to the investigations of light scattering and their applications in probing biological tissues and cells.

Venkat Gudivada received the BTech degree in civil engineering from JNT University, the MS degree in civil engineering from Texas Tech University, and the MS and PhD degrees in computer science from the University of Louisiana at Lafayette. He is a professor and chair of the Computer Science Department, East Carolina University. Prior to this, he was a professor and founding chair of the Weisberg Division of Computer Science, Marshall University. His industry tenure spans over six years in senior leadership roles at Wall Street companies. He received his PhD and MS degrees in Computer Science from the University of Louisiana at Lafayette. His current research interests include data management, high performance computing, cognitive computing, image and natural language understanding, and personalized learning. He is a member of the IEEE.