Augmentation and Evaluation of Training Data for Deep Learning

Junhua Ding
Department of Computer Science
East Carolina University
Greenville, NC, USA
dingj@ecu.edu

XinChuan Li
School of Computer Science
China University of Geosciences
Wuhan, China
lihanyu2006@foxmail.com

Venkat N. Gudivada
Department of Computer Science
East Carolina University
Greenville, NC, USA
gudivadav15@ecu.edu

***Abstract*—Deep learning is an important technique for extracting value from big data. However, the effectiveness of deep learning requires large volumes of high quality training data. In many cases, the size of training data is not large enough for effectively training a deep learning classifier. Data augmentation is a widely adopted approach for increasing the amount of training data. But the quality of the augmented data may be questionable. Therefore, a systematic evaluation of training data is critical. Furthermore, if the training data is noisy, it is necessary to separate out the noise data automatically. In this paper, we propose a deep learning classifier for automatically separating good training data from noisy data. To effectively train the deep learning classifier, the original training data need to be transformed to suit the input format of the classifier. Moreover, we investigate different data augmentation approaches to generate sufficient volume of training data from limited size original training data. We evaluated the quality of the training data through cross validation of the classification accuracy with different classification algorithms. We also check the pattern of each data item and compare the distributions of datasets. We demonstrate the effectiveness of the proposed approach through an experimental investigation of automated classification of massive biomedical images. Our approach is generic and is easily adaptable to other big data domains.**

***Keywords*-big data; machine learning; neural network; deep learning; convolutional neural network; support vector machine; diffraction image**

I. Introduction

Scalable and high performance data processing infrastructure and analytics tools are needed to extract value from big data. For example, deep learning algorithms and GPUs have been widely adopted for analyzing big data [1]. A challenging factor for effectively extracting value from big data is its size and quality. Furthermore, the data available for training algorithms such as a deep learning classifier is often not large enough. Augmenting data through methods such as transforming existing data items into new ones is a widely used practice. However, it is difficult to determine whether or not the augmented data is valid. It is necessary to systematically evaluate the quality of the generated data through transformations. Also, the generated training data may include noise in the form of, for example, mislabeled data.

Published research has shown that anomalies and noise in training datasets could significantly decrease the performance and accuracy of data analysis [2] [3]. To address these problems, we have two choices: devise robust machine learning algorithm which can deal with noisy training data, or improve the quality of the data through filtering [4].

Both general purpose and domain-specific techniques and tools have been developed for quality assurance of big data. Gao *et al.* provides an overview of issues, challenges and tools for validation and quality assurance of big data [5]. 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. Web is one primary source of big data, and work on the evaluation of the veracity of web sources exists in the literature.

Machine learning algorithms have been used for detecting duplicates in data that originated from multiple sources [6]. *Data filtering* is an approach for improving data quality through noise removal. Data publishers and subscribers can filter noisy data using domain models and rules [7]. Due to the massive scale of big data, automated filtering of data is essential. However, investigations in this direction are just beginning to appear.

In this paper, we introduce a systematic approach for separating noisy data from biomedical image datasets to enable extraction of knowledge from big data. More specifically, we develop a machine learning approach for separating both invalid and noisy data from a dataset. Our approach includes an iterative process for separating noisy data from regular data using a deep learning classifier. We also discuss an approach to generating large volume of high quality data for improving classification accuracy. To ensure the quality of the augmented data, we evaluate data quality through cross validation of classification accuracy with different classification algorithms. We also check the pattern of each data item and compare their distributions.

We describe the proposed approach and demonstrate its effectiveness through separating the diffraction images of biology cells into several categories including noisy class. Diffraction images of cells are acquired using a polarization diffraction image flow cytometer (p-DIFC), which is used for quantifying and profiling 3D morphology of single cells [8]. The 3D morphological features of a cell that are captured in the diffraction image are used for accurately classifying cell

types. p-DIFC can take the diffraction images of nearly 100 cells per second. Using p-DIFC, we have collected over a million diffraction images for different types of cells.

II. Cell Diffraction Images

Work on classification of cell diffraction images using machine learning has been reported in the literature. However, p-DIFC imaging may include lots of *non-cell particles* such as *ghost cell body*, *aggregated spherical particles* (aka *fractured cells*), and cell debris and small particles (collectively referred to as *debris*). We refer to the viable cells with intact structures as cells. The diffraction images taken from the non-cell particles are also collected into the diffraction image dataset. The diffraction images of the non-cell particles comprise the noise data.

To accurately classify cells, it is necessary to separate the non-cell diffraction images (i.e., the noise) from the cell diffraction images. Manually separating the noise images from cell images is not feasible from a practical standpoint. To address this issue, we developed a deep learning [1] approach for automated classification of diffraction images. We classify the diffraction images into three categories: cells, fractured cells, and debris. We developed the classifier using a deep learning architecture based on AlexNet [9] and TensorFlow frameworks. We trained the classifier using diffraction images of cells, fractured cells, and debris.

The size of a raw 8-bit gray scale p-DIFC diffraction image is 640×480 pixels. Since AlexNet works with images of size 227×227 pixels, we resized the original diffraction images to 227×227. AlexNet classifier for diffraction images requires a large volume of training images. We developed an approach for generating several small diffraction images (aka augmented diffraction images) from the original images. The classification accuracy is cross checked with n-fold cross validation (NFCV) and a confusion matrix.

To check the quality of the training data, we developed a support vector machine (SVM) for classifying the three categories of diffraction images. First, we train the classifier on the original and the augmented diffraction image datasets separately. Next, we compare the classification accuracies. We also investigate whether the small images can capture enough morphology information as the original images. We require each small image to be different from its original image. Furthermore, we desire that all the small images which are generated from the same original to exhibit different textual patterns. Lastly, we check the distribution of selected feature values of the original and the augmented datasets to determine whether they are consistent.

The remainder of this paper is organized as follows. Section 2 provides the domain background in cell imaging and automated classification of diffraction images. Section 3 describes an approach for systematically evaluating the quality of augmented iamge datasets. Related work is discussed in Section 4 and Section 5 concludes the paper.

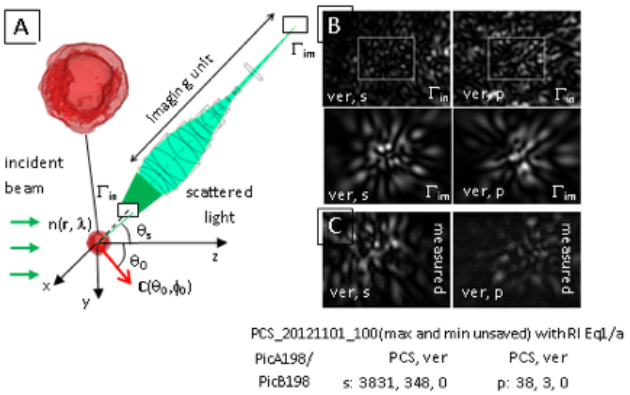


Fig. 1. (A). Light scattering schema of p-DIFC, (B).Software simulated diffraction images, and (C). p-DIFC acquired diffraction images.

III. Automated Classification of Diffraction Images

We first discuss morphology based cell classification. Next, we introduce automated classification of diffraction images using SVM and deep learning techniques.

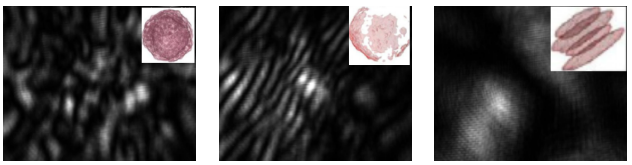
A. Morphology Based Cell Classification

Cells exhibit highly varied and convoluted 3-dimensional (3D) structures through intracellular organelles to sustain phenotypic variations and functions. Cell classification is important to biology and life science research. Morphology based approaches at the single cell-level is attracting intense research efforts for their direct relations to cellular functions. p-DIFC is used to rapidly acquire cross-polarized Diffraction Image (p-DI) pairs from single cells [8]. It adopts Stokes vectors and Mueller matrices to account for the polarization change in scattered light as a result of intracellular distribution of refractive index, $n(r, \lambda)$, or its 3D morphology. The incident light and its polarization state is represented by Stokes vector (I_0, Q_0, U_0, V_0), which propagates along the z-axis. Likewise, the scattered light and its polarization is represented state vector (I_s, Q_s, U_s, V_s) along (Θ_s, Φ_s) direction, as shown in Fig. 1. Different from images acquired by non-coherent light, the p-DI pairs present characteristic patterns due to the coherent light scatter emitted by the intracellular molecular dipoles induced by an incident laser beam [8]. The p-DI data thus provide a data source to probe the 3D morphology of the illuminated cells that requires machine learning techniques for extracting morphological and molecular information.

During the last decade, Ding *et al.* have developed different machine learning approaches, including Support Vector Machine (SVM) [10] and deep learning, for rapid and accurate cell morphology analysis of cell diffraction images [3] [11] [12].

B. Datasets

A collection of diffraction images acquired using p-DIFC may include images taken from non-regular cells especially fractured cells and debris in the samples. For some research



(a) (b) (c)

Fig. 2. A p-DIFC acquired diffraction image of (a) a viable cell of intact structure, (b) a ghost cell body or aggregated spherical particles, and (c) a cell debris or small particle. The top right corner shows the corresponding particle of each image.

projects, one needs only normal cell images. For some other research such as apoptosis study, we need only fractured cell images. Therefore, it is necessary to build a tool to automate the separation of the three types of diffraction images.

The three types of cell particles have different morphology structures that are precisely captured in p-DIFC diffraction images. Using these textual patterns, a biologist can separate the three types of images visually. Fig. 2 shows thee sample p-DIFC diffraction images and their corresponding particles. The textual pattern of the diffraction image of a viable cell of intact structure contains many bright normal speckles. On the other and, a ghost cell body or aggregated spherical particle includes bright strips. Lastly, a cell debris or small particle shows many large diffuse speckles.

The difference in textual patterns of the three categories of diffraction images is good enough to separate the three categories using machine learning algorithms. We acquired many diffraction images for the three categories of particles using p-DIFC and then selected several thousands of diffraction images as the initial dataset. For the experimental study, we selected a total of 7519 diffraction images. Each diffraction image was manually inspected and and its category was labeled. Normal cells are labeled as cells, fractured cells as strips, and debris as simply debris. The initial image dataset is comprised of 2232 normal cells, 1645 fractured cells, and 3642 debris. Each category of the diffraction images is stored in a separate directory. We note that some of the diffraction images could have been incorrectly labeled, whereas some others were difficult to label due to low visual quality.

C. SVM-based Image Classification

An SVM performs binary classification in general [10]. To implement a multiclass classification, several SVM classifiers are combined by comparing *'one against the rest'* or *'one against one'*. We have implemented the classifier for diffraction images using LIBSVM [13], which is an open-source toolkit for SVM.

The textual pattern of the diffraction images is defined using a group of Grey Layer Collaborative Matrix (GLCM) features [14]. We use a total of 20 features – 14 are features from the original image and 6 features from the extended images. The definition of each feature can be found in Ding's previous work [15]. The procedure of building an SVM classifier for diffraction images is given below:

TABLE I
CONFUSION MATRIX OF THE CLASSIFICATION OF DIFFRACTION IMAGES

	Cells	**Debris**	**Strips**
Cells	74.50%	16.00%	9.50%
Debris	6.50%	81.50%	12.00%
Strips	14.00%	24.00%	62.00%

1) Calculate GLCM features for each diffraction image in the training and testing datasets.
2) Label each diffraction image with its category such as its cell type, and build a feature vector consisting of its GLCM feature values and its label. The feature vectors of all diffraction images in a dataset form a feature matrix.
3) Train the SVM classifier using the select kernel and the feature matrix of the training dataset.
4) Test the classifier with diffraction images in test dataset, and validate the classification performance using criteria such as N-fold Cross Validation (NFCV) and confusion matrix.

We built an SVM classifier using the diffraction image dataset. We selected 1000 diffraction images for each of the three classes, and built the feature matrix with GLCM feature values and the corresponding types. Each feature vector includes 16 GLCM feature values since the value of one feature is all 0s and another three features are defined on image format, which was not accounted for in this study. The average classification accuracy of 10 fold cross validation (10FCV) for *cells, debris* and *strips* is 74.50%, 81.50% and 62.00%, respectively. The simplified confusion matrix is shown in Table I [16].

To improve the classification accuracy of the SVM classifier, we have experimented with many different techniques such as pre-selecting the images using image processing and cluster analysis techniques [3], and feature selection [17]. Our recent experiments have shown that deep learning approaches greatly improve the classification accuracy [18].

D. Deep Learning Based Image Classification

Diffraction images are relatively simple due to their low resolution and absence of background noise. Therefore, we selected AlexNet model [9] which is implemented in TensorFlow framework to build the deep learning classifier. As deep learning requires a large number of features, the size of the training dataset is also large.

AlexNet is trained using about 1.2 million images. We did not use the pre-trained AlexNet, but used only its net architecture. We initially made some minor changes of the architecture of AlexNet such as changing the output of the last fully connected layer from 1000 categories to 3, and removed some convolutional layers. These changes neither improved the training performance or the classification accuracy. To avoid the potential of introducing bugs, we decided to keep the original AlexNet architecture in tact. We have collected only 7519 raw diffraction images, which are not large enough

for training AlexNet. Therefore, we used data augmentation approaches for producing a larger volume training dataset.

E. Data Augmentation

The size of a raw diffraction image of a cell is 640×490 pixels. It is large enough to be divided into several small images of size 227×227 pixels, which is the size of AlexNet input image. Each small image should contain sufficient information to represent the original image according to p-DIFC principles [8]. If we check a diffraction image of a cell, it is not difficult to find the textual pattern is repeated in the image as shown in Fig. 2. Since the lens of the camera taking cell diffraction image of cell has a different angle to each part of the cell, the textual pattern is not simply repeated in the image. A carefully chosen sub-image can have enough information to be a representative of the whole image. A diffraction image may also include large black background which is useless for classification. Therefore, a rigorous approach for producing the small images is necessary. This property can be further confirmed by the diffraction images shown in Fig. 7, which are produced by simulating the light scattering of scatterers using aDDA (a light scattering simulation program) [19].

F. Cropping images

As noted earlier, AlexNet accepts input images of size 227×227 pixels. Also, the size of original diffraction images is 640×480 pixels. Therefore, a small image is about 1/5 of the size of the original image. Furthermore, since a diffraction image may contain significant black area, the center of the textual pattern such as bright speckles or strips may not be the center of the image. We need find the center of the textual pattern area to perform cropping, which is normally the brightest area in the image.

Given a 5×5 pixel window, cropping program calculates the average intensity of the window. Then it slides the window by several pixels in steps to cover the whole image, to determine a window that has the largest average intensity. For example, the intensity range of a 8-bit resolution image is from 0 to 255. The window with the largest average intensity is set as the center for cropping small images. If multiple windows have the largest average intensity, then the one furthest to the boundary is selected as the center. A small image is cropped from the original image around the center first, and then more small images are cropped through sliding the window from the center some pixels in any direction as shown in Fig. 3.

In our study, the 227×227-pixel window is moved from the center in 8 different directions and the angle θ between two adjacent directions is 45^o. The sliding window moves d pixels from last window in a direction and crops a new small image. The *moving distance* d ranges from 7 to 17 pixels. However, other d values may also work well. The sliding window can be moved in one direction several steps to crop multiple small images in that direction. When a sliding window is moved to a new position, it is necessary to ensure that the entire sliding window is still contained in the original image boundary. If not, the window is discarded and no further sliding in that direction takes place. Using the approach, we generated 56 small images from a normal cell image with a step distance d of 11 pixels (which means sliding 11 pixels 7 times in each direction), 40 images from a debris image with d as 14, and 72 images from a strips image with d as 9.

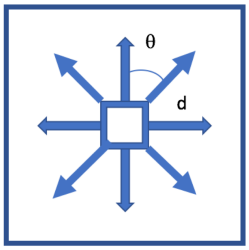


Fig. 3. An Illustration of Image Cropping

The small image generation via cropping is fully automated with a python program. Though the multiple small images cropped from the same image are different from each other, but they all represent the same original image and are labeled as the same category as that of the original image. Finally, 105291 small diffraction images of cells, 127733 small diffraction images of debris, and 92767 small diffraction images of strips are selected for training the deep learning classifier.

G. Pooling Images

Cropping technique does not work for the case where the whole image is critical for classification. In such cases, local features that are extracted from a local area are not enough to represent the global features extracted from the whole image. A different technique is needed for producing the training data from the limited number of original images. We experimented a *pooling technique* for producing large volume of training data. A raw diffraction image is downsampled into a small image using *pooling*. Multiple small images can be produced from a raw image with different pooling configurations. Also, small images can be produced with different pooling functions such max pooling or average pooling [20].

To produce multiple small diffraction images from one original image, we apply different pooling window sizes and different sliding stride to the same image. Since the size of the small image is 227×227 pixels, and the size of an original image is 640×480 pixels, we resize the original image into a square one. We cut three different size squares from an image, which are 455×455, 456×456, and 457×457 pixels. Next, 3×3, 4×4, and 5×5 pooling windows are applied to these three squares. The stride distance is set to 2 pixels. The size of the output image from the pooling is $s * s$ pixels, $s = (x - m)/c + 1$, where $x \times x$ pixels are the size of the input image of the pooling, $m \times m$ pixels are the size of the pooling window, and c is the stride distance. For example, if the input image is 455×455, pooling window is 3×3 pixels, and stride distance is 2, the size of the output image will be 227×227 pixels. Fig. 4 shows a comparison of the original images and their pooling images (the ratio of the images were changed due to formatting issues). It clearly shows that the textual patterns of the original image are well preserved in the pooling image. The pooling steps are:

1) For each diffraction image, select position (10, 10) of the image as position (0, 0) of the new cropped images

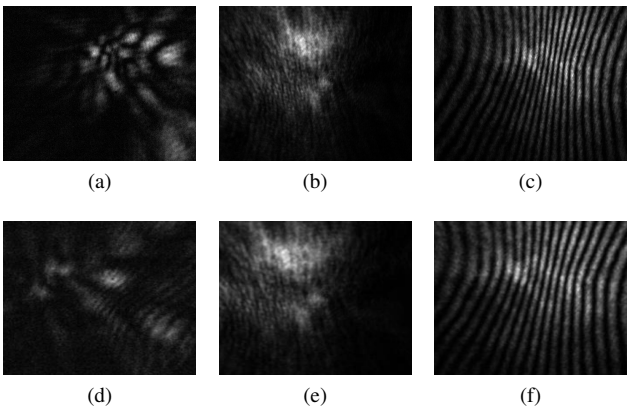


Fig. 4. Diffraction images and their pooling images (a) a cell, (b) an debris, (c) a fractured cell, (d) a cell after pooling, (e) a debris after pooling, and (f) a fractured cell after pooling.

to cropped three square images: 455×455, 456×456, and 457×457 pixels, respectively.

2) Move the cropping position from (10, 10) to (10+h, 10) to crop another three square images with size at 455×455, 456×456, and 457×457 pixels; where h is 10 for normal cell images, 20 for debris, and 5 for strips. Continue the step 16 times for normal cell images, 8 times for debris, and 32 times for strips. Therefore, each original cell image produces 48 different square images, each debris image produces 24 square images, an each strips image produces 96 square images. About 100,000 images can be produced for each category from the original diffraction images.
3) Pooling window 3×3 pixels is applied to 455×455 pixels images, 4×4 pixels window is applied to 456×456 pixels images, and 5×5 pixels window is applied to 457×457 pixels images. Each square image is downsampled into a 227×227 pixels small image after the pooling.
4) Each small image is labeled as the same category as the original image where the small image is produced from.

We have experimented with three pooling functions including average-pooling, max-pooling and min-pooling. However, a dataset uses only the same pooling function. The experimental results of the three different datasets will be discussed in the next section.

H. Experiment Results

All experiments were conducted on the same original cell diffraction images, fractured cells and debris. The three categories of images are stored in three different folders, and then cropping or pooling are applied to each image to produce and label around 100,000 small images for each category. The small images were stored in three folders according to their labels/categories. 8FCV and confusion matrix are used to validate the classification results. Many experiments have been conducted for checking and validating the classification accuracy, but we will describe only the important results in this section.

	cell	debris	strip
cell	99.02	0.62	0.37
debris	1.89	97.19	0.92
strip	0.06	0.18	99.76
cell	99.63	0.37	0
debris	1.99	97.19	0.82
strip	0.47	0.24	99.29
cell	99.26	0.62	0.12
debris	0.92	98.57	0.51
strip	0.06	0.12	99.82
cell	99.38	0.55	0.06
debris	0.92	98.82	0.26
strip	0	0.18	99.82

Fig. 5. Confusion matrix of a classification experiment.

TABLE II
A CONFUSION MATRIX OF AN AVERAGE-POOLING DATA SET

	Cells	Debris	Strips
Cells	0.857	0.098	0.045
Debris	0.006	0.987	0.006
Strips	0.005	0.052	0.943

1) Experiment results with cropped images: 8FCV shows the average classification accuracy of normal cells at 99.36%, debris at 97.74% and fractured cells at 99.81%. Fig. 5 shows the confusion matrix of 4 groups. From 8FCV results, we note that the classifier built on AlexNet is effective for classifying the thee categories of diffraction images. Also, the dataset produced from the original images is sufficient for training the classifier.

2) Experiment results with pooled images: The 8FCV result of the classification based on the dataset generated using average-pooling shows that the average classification accuracy for debris and strips is a little bit higher than the dataset built by cropping. However, the average classification accuracy for cells is much lower at 85.7% *v.s.* 94.22%. As shown in Table II, nearly 10% cells are incorrectly classified as debris, and only 4.5% are incorrectly classified as strips.

We note that the difference between the textual patterns in diffraction images of normal cells and debris is the size of speckles – a debris has larger speckles. The average-pooling would decrease the difference between the normal cells and debris, which could be a reason that more normal cells are classified as debris. The 8FCV result and confusion matrix further confirm that a deep learning classifier is more effective. However, the dataset created using average-pooling could be improved. Therefore, we have also experimented with max-pooling and min-pooling functions for generating training data.

The 8FCV result of the classification based on the dataset created using max-pooling is almost identical to the result shown in Fig. 5. Average classification accuracy of cells is 87.9%, of debris is 98.5%, and of strips is 94.6%. The confusion matrix is also the same. However, the 8FCV result of the classification based on the dataset created using min-pooling is much better. The min-pooling function chooses the

TABLE III
A CONFUSION MATRIX OF A MIN-POOLING DATA SET

	Cells	Debris	Strips
Cells	0.935	0.036	0.030
Debris	0.024	0.961	0.015
Strips	0.023	0.044	0.933

TABLE IV
CONFUSION MATRIX OF AN SVM BASED CLASSIFICATION OF AUGMENTED DATA

	Cells	Debris	Strips
Cells	77.94%	10.97%	11.09%
Debris	7.33%	84.39%	8.28%
Strips	20.25%	18.69%	61.06%

minimal value of the sliding window to represent the whole window in the new image. The average classification accuracy of cells is improved to 93.5%, of debris to 96.1%, and of fractured cells to 93.3%. The confusion matrix is shown in Table III. Fig. 6 shows a comparison of the average-pooling images and their corresponding min-pooling images. We found that the textual patterns in min-pooling images are clearer, which might explain the improvement in the classification accuracy.

I. Discussion

In this section, we discuss how machine learning classifiers for separating noise data from training data are built. We also demonstrate its process and effectiveness through classifying three categories of diffraction images. Through separating fractured cell images and debris images from the training data, one can get noise-free cell images that are important for training a classifier. However, the classification accuracy of an SVM based classifier is not high enough. Therefore, we built the deep learning classifier. Training the deep learning classifier needs a large amount of high quality training data. Different data augmentation approaches including cropping and pooling are experimented for producing the needed data.

Our experimental results show that a deep learning classifier is highly effective, which can be used for automated selection of data from a large dataset and filtering noise data. The quality of dataset can be iteratively improved through multiple rounds of selection, in which the incorrectly classified data items from previous round of classification were inspected and re-labeled or removed for next round of training and classification. Since the quality of the training data is key to the quality the deep learning classifier, it is important to evaluate the quality of the augmented dataset in term of representativity, fidelity and variety. A high quality augmented dataset should have high a representativity, fidelity, and variety.

IV. EVALUATION OF THE QUALITY OF AUGMENTED DATA

In this section, we discuss how to systematically evaluate the quality of datasets that are produced from original diffraction images using cropping or pooling techniques. We evaluate the dataset using representativity, fidelity, and variety. Representativity means that the dataset includes all information in the original dataset and it can represent the original dataset to train a machine learning classifier. Fidelity refers to the fact that a generated data item cannot be distinguished from the original source. Variety means the augmented dataset should be normally distributed for non-trivial features. For the diffraction image case study, we first checked whether the small size diffraction images can be used for classifying the diffraction images based on SVM algorithm to achieve the similar accuracy as the original images. Then we checked the textual pattern of the small images to ensure the small image can capture enough morphology information as its original image. Finally, we compared the distribution of feature values of the augmented data set and the original image data set.

A. Checking the classification accuracy of the SVM classifier

Table I shows a confusion matrix of the classification of diffraction images using an SVM classifier which is trained using the original diffraction image datasets. We check the classification accuracy of the classifier based on the generated diffraction images. We trained the SVM classifier using the dataset consisting of the small diffraction images produced by cropping or pooling from the original diffraction images. These are the images that are also used for training the classifier shown in Table I. We select 3000 small images for each category and then conduct an 8FCV. The confusion matrix of the average result of 8FCV is shown in Table IV. Comparing the results in Tables I and IV, it is easy to find the classification accuracy of the two training datasets are almost identical. We conclude that the augmented dataset accurately represents the original dataset for training the SVM classifier. This provides evidence to use the augmented dataset to represent the original dataset for training deep learning classifiers.

B. Checking the textual pattern in diffraction images

Program aDDA [19] is an open-source tool for simulating light scattering of particles using discrete dipole approximation (DDA) approach. aDDA can be used to calculate a diffraction image of a scatterer. In theory, the diffraction image of a scatterer calculated from aDDA is identical to the p-DIFC acquired diffraction image of the same scatterer. We use aDDA simulated diffraction images to check replication property of textual patterns in the image to investigate whether partial of the image can represent the whole image. At the same time, we expect each of the small image cuts from the original image is unique. Fig. 7 shows 6 diffraction images calculated from 6 different scatterers. Fig. 7a is a diffraction image of a sphere scatterer, where the regular strips repeated in the image. If we cut a small window of image such as 300×300 pixels from the center of the original image (the size is 640×480 pixels), the small image is sufficient to represent the textual pattern of the whole image.

If we shift the window from the center with small distance, we can cut more small images to represent the whole image, but each of the images is different. The same property can be

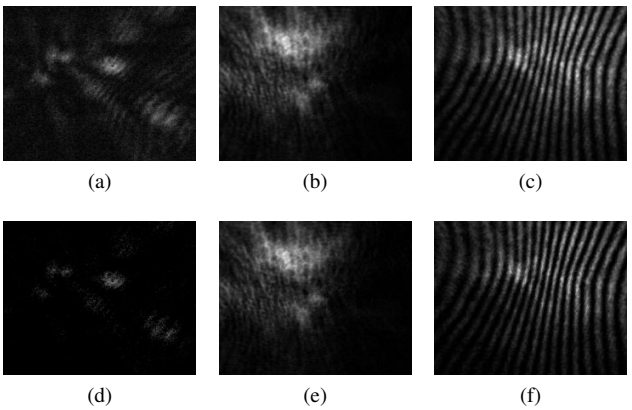


Fig. 6. Diffraction images produced with different pooling functions (a) an average-pooling image of a cell, and (d) the corresponding min-pooling image of (a); (b) an average-pooling image of a debris, and (e) the corresponding min-pooling image of (b); (c) an average-pooling image of a strips, and (f) the corresponding min-pooling image of (c).

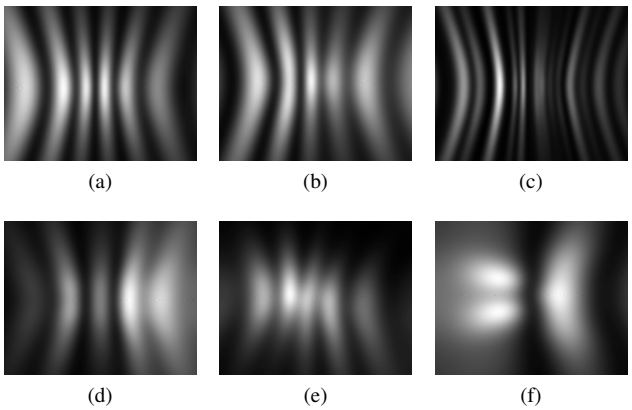


Fig. 7. aDDA calculated diffraction images of (a) a sphere, (b) an ellipsoid, (c) a bi-sphere, (d) a bi-coated with one sphere in the center, (e) a bi-coated with one sphere shifting from the center, and (f) an RBC.

observed from other images in 7, and small images can be cut from each of them, where (b) is a diffraction image of an ellipsoid scatterer, (c) is a diffraction image of a bisphere scatterer with two spheres stay side by side. The two spheres are clearly shown by the repeated strips patterns in the image. It is easy to see that a small window of the image also can capture the repeated strips pattern in the images. (d) is a bi-coated scatterer with one ellipsoid containing a sphere in its center, (e) is a bi-coated scatterer with one ellipsoid containing a sphere shifting from its center, and (f) is a scatterer for modeling a red blood cell with shape like a peanut. It is not difficult to see, if we properly cut a small image from each one in Fig. 7, each small image can be easily map to its original one as shown in Fig. 8.

The ratio of the images shown here is not consistent with the original images due to formatting issues, but they correctly explain the idea of how a small part of an image can represent a whole image. From this observation, we believe that if we cut small images around the center of the textual pattern in the image, the small image should have enough information to represent the original large image.

C. Checking the feature pattern in the data set

Our experiments have shown that any small diffraction image can accurately represent its original diffraction image for classification. The deep learning classifier always categorizes a small image into the same category as its source diffraction image. This is a good thing, but at the same time, it is necessary to check the contribution of the small images towards training effectiveness. If the small images from an original image have identical feature values, then these images are redundant to the training. Therefore, it is necessary to check how close the feature values of these small images area. Given an image to a trained deep learning classifier, we collect the output at the last fully-connected layer, which includes 4096 features in AlexNet. Then we compare the feature values between two input diffraction images. Although it is not

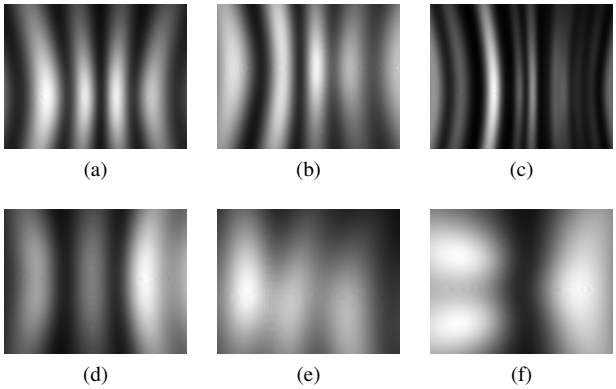


Fig. 8. small diffraction images cropped from images in Fig. 7, (a) a sphere, (b) an ellipsoid, (c) a bi-sphere, (d) a bi-coated with one sphere in the center, (e) a bi-coated with one sphere shifting from the center, and (f) an RBC.

difficult to find the difference between two feature vectors, it is fairly challenging to calculate the difference between two feature vectors since each feature is not a simple scalar parameter. Therefore, we use a different way to evaluate the small diffraction images. Since the textual pattern is essential to the classification of diffraction images, we can check the difference of GLCM feature values between two images. If the GLCM feature values of the small images are different from its original images, we also need to check the distribution of the dataset of small images and the distribution of the dataset of the original images. If the GLCM feature values of the small diffraction images that produced from the same original image are different, and the distribution of the GLCM features of the dataset of the small image is consistent with the distribution of the GLCM features of the dataset of the original images, we believe the dataset of the small images represent the original images well and contributes to the generalization of training.

1) Comparing GLCM feature values of diffraction images: We first calculated the GLCM feature values for each diffraction image, and the small images are grouped together with their original image. Then we compared every GLCM feature

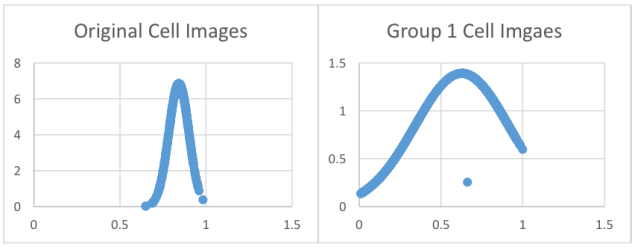


Fig. 9. Compare the distribution of a GLCM feature values of the data set of the original images and the data set of the small images pooled from the original images.

for all images in a group. If two images have different values of at least one GLCM feature, the two images are considered as different. Table V shows a partial comparison results of pooled small images and its original image in 6 GLCM features. $Img-1$ to $Img-5$ are pooled image from original image $Img-0$. We checked every group of images, and did not find two identical images in each group.

2) Comparing GLCM feature distributions of datasets: We created a distribution of a GLCM feature for all original images that belong to the same type. Then we created the same distribution for a group of small images that were produced from the original images. We compared the two distribution to see whether the distributions are consistent. Fig. 9 shows a comparison of the normal distribution of a GLCM feature of the original diffraction image data and the one of the small diffraction images pooled from the original ones.

We created a normal distribution with the normalized feature values (i.e. min-max normalization), mean of the values and standard deviation, and the curve was drawn based on the probability mass function. It is not difficult to see that the two distributions are not exactly the same. However, both of them are normally distributed. Different GLCM features and different groups images are checked using the same distribution. We found that the distribution pattens between the dataset of original images and the datasets that are pooled or cropped images from the original images are consistent. Therefore, we conclude that both the pooling and cropping are effective for data augmentation of diffraction images.

V. Related Work

Deep learning researchers are faced with the trade-off between using better deep learning architectures and better training data [21]. However, building a deep learning architecture for small training datasets is a grand challenge. Even a deep learning architecture is targeted for low-shot classification requires data augmentation [22]. Therefore, using large training data is a more feasible approach for building a high quality deep learning solution. For example, the original AlexNet was trained with 1.2 million images, and the classifier for categorizing the three categories of diffraction images required over 100,000 diffraction images for each category. However, many domain specific applications cannot produce enough data for the deep learning. Data augmentation through producing high quality artificial training data based on original data is a widely adopted practice for enhancing the training dataset in deep learning.

Each domain specific application can produce artificial data according to the domain models such as using aDDA for producing diffraction images of cells. Sampling from a large image is also an effective approach for producing image data [23] [24]. In this paper, we have used cropping and pooling from original images for producing large volume of training data. The augmented datasets are systematically evaluated in terms of representativity, fidelity and variety.

Generative models are proposed recently for producing artificial data using deep learning techniques [25]. Although large amount of initial data are required to produce artificial data using generative models, it is a promising technique for generating a large amount high quality artificial data. Through learning the transformation relation from similar datasets to produce data for low-shot learning is also a promising idea for data augmentation [22]. However, poor quality data could cause serious problems such as wrong prediction or low accuracy of the classification.

Quality attributes of big data such as availability, usability, and reliability have been well defined in some publications [26] [7]. Although general techniques and tools are developed for quality assurance of big data, much more work remains to be done on the quality assurance of domain specific big data such as health care management data, social media data, and finance data.

Machine learning algorithms such as Gradient Boosted Decision Tree (GBDT) are used for detecting data duplication [6]. Data filtering is an approach for quality assurance of big data through removing bad data from data sources. For example, Ekambaram *et. al* recently reported a machine learning approach for finding label noise in the training data [27].

VI. Summary and Future Work

Training a deep learning model normally requires a large volume of training data as well as high quality training data. Large volume of training data may include noise data. Therefore, it is necessary to separate the noise data from the training data. In this paper, we proposed a deep learning approach for the automated classification of training data into different categories of data, one of which is a noise category.

In many cases, the original training data needs to be transformed to fit the input size requirements of deep learning models. In other cases, new data is required through data augmentation as the original data is insufficient in size. We discussed different data augmentation approaches. We have also evaluated the quality of the training data through cross validation of the classification accuracy.

To demonstrate the proposed approaches to data augmentation and their effectiveness, we conducted a thorough experimental study on automated classification of massive diffraction images. The proposed approaches and experience collected from this experimental study can be adopted for data augmentation and evaluation of big data in other domains.

TABLE V
A COMPARISON OF GLCM FEATURE VALUES AMONG DIFFRACTION IMAGES

	ASM	CON	COR	VAR	IDM	SAV
Img-1	0.069028397	0.300829789	0.978817148	0.164436833	0.597584362	0.231334924
Img-2	0.656232866	0.224157652	0.980417054	0.130393754	0.864684597	0.109085359
Img-3	0.967753732	0.017326016	0.688163634	0.000792628	0.989551474	0.026260393
Img-4	0.026913115	0.159871456	0.99144913	0.166720538	0.621992081	0.292738468
Img-5	0.587408623	0.235948351	0.951577511	0.063657455	0.836049419	0.093207132
Img-0	0.329891354	0.001331282	0.997723423	0.117280715	0.84966343	0.282349741

ACKNOWLEDGMENT

The authors would like to thank Dr. Xin-Hua Hu and Pruthvish Patel at East Carolina University for assistance with the experiments. This research is supported in part by grants #1560037 and #1730568 from the National Science Foundation. We gratefully acknowledge NVIDIA Corporation for Tesla K40 GPU gift, which is used for conducting this research.

REFERENCES

[1] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[2] E. Giannoulatou, S.-H. Park, D. Humphreys, and J. Ho, "Verification and validation of bioinformatics software without a gold standard: a case study of bwa and bowtie," *BMC Bioinformatics*, vol. 15(Suppl 16):S15, 2014.

[3] J. Zhang, Y. Feng, M. S. Moran, J. Lu, L. Yang *et al.*, "Analysis of cellular objects through diffraction images acquired by flow cytometry," *Opt. Express*, vol. 21, no. 21, pp. 24 819–24 828, 2013.

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

[5] J. Gao, C. Xie, and C. Tao, "Big data validation and quality assurance – issuses, challenges, and needs," in *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, March 2016, pp. 433–441.

[6] C. H. Wu and Y. Song, "Robust and distributed web-scale near-dup document conflation in microsoft academic service," in *2015 IEEE International Conference on Big Data (Big Data)*, Oct. 2015, pp. 2606–2611.

[7] V. Gudivada, R. Raeza-Yates, and V. Raghavan, "Big data: Promises and problems," *IEEE Computer*, vol. 48, no. 3, pp. 20–23, 2015.

[8] K. Jacobs, J. Lu, and X. Hu, "Development of a diffraction imaging flow cytometer," *Opt. Lett.*, vol. 34, no. 19, p. 29852987, 2009.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1097–1105.

[10] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, Jan. 1998.

[11] J. Ding, D. Zhang, and X. Hu, "An application of metamorphic testing for testing scientific software," in *1st Intl. workshop on metamorphic testing with ICSE*, Austin, TX, May 2016.

[12] K. Dong, Y. Feng, K. Jacobs, J. Lu, R. Brock *et al.*, "Label-free classification of cultured cells through diffraction imaging," *Biomed. Opt. Express*, vol. 2, no. 6, p. 17171726, 2011.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[14] R. Haralick, "On a texture-context feature extraction algorithm for remotely sensed imagery," in *Proceedings of the IEEE Computer Society Conference on Decision and Control*, Gainesville, FL, Dec. 1971, pp. 650–657.

[15] S. K. Thati, J. Ding, D. Zhang, and X. Hu, "Feature selection and analysis of diffraction images," in *4th IEEE Intl. Workshop on Information Assurance*, Vancouver, Canada, August 2015.

[16] J. Ding, J. Wang, X. Kang, and X. Hu, "Building an svm classifier for automated selection of big data," in *2017 IEEE International Congress on Big Data*, Honolulu, HI, 2017.

[17] S. Vilkomir, J. Wang, N. L. Thai, and J. Ding, "Combinatorial methods of feature selection for cell image classification," in *2017 IEEE Intl. Workshop. on Combinatorial Testing and Applications*, Prague, Czech, July 2017.

[18] J. Ding, X. Kang, X. H. Hu, and V. Gudivada, "Building a deep learning classifier for enhancing a biomedical big data service," in *2017 IEEE Intl. Conf. on Services Computing*, Honolulu, HI, June 2017.

[19] (2016, Sept.) Adda project. [Online]. Available: https://github.com/adda-team/adda

[20] (2017, Jan.) Deep learning tutorial. [Online]. Available: http://deeplearning.net/tutorial/lenet.html

[21] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[22] B. Hariharan and R. B. Girshick, "Low-shot visual object recognition," *CoRR*, vol. abs/1606.02819, 2016. [Online]. Available: http://arxiv.org/abs/1606.02819

[23] D. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *Intl. Conf. on Medical Image Computing and Computer-assisted Intervention*, 2013, pp. 411–418.

[24] B. Dong, L. Shao, M. D. Costa, O. Bandmann, and A. F. Frangi, "Deep learning for automatic cell detection in wide-field microscopy zebrafish images," in *2015 IEEE 12th Intl. Symposium on Biomedical Imaging (ISBI)*, April 2015, pp. 772–776.

[25] (2017, Jan.) Open ai: Generative models. [Online]. Available: https://openai.com/blog/generative-models/

[26] L. Cai and Y. Zhu, "The challenges of data quality and data quality assessment in the big data era," *Data Science Journal*, vol. 14:2, pp. 1–10, 2015.

[27] R. Ekambaram, D. Goldgof, and L. Hall, "Finding label noise examples in large scale datasets," in *2017 IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, Oct. 2017.