Validating a Deep Learning Framework by Metamorphic Testing

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

Xiaojun Kang
School of Computer Science
China University of Geosciences
Wuhan, China
xj_kang@126.com

Xin-Hua Hu
Department of Physics
East Carolina University
Greenville, NC, USA
hux@ecu.edu

***Abstract*—Deep learning has become an important tool for image classification and natural language processing. However, the effectiveness of deep learning is highly dependent on the quality of the training data as well as the net model for the learning. The training data set for deep learning normally is fairly large, and the net model is pretty complex. It is necessary to validate the deep learning framework including the net model, executing environment, and training data set before it is used for any applications. In this paper, we propose an approach for validating the classification accuracy of a deep learning framework that includes a convolutional neural network, a deep learning executing environment, and a massive image data set. The framework is first validated with a classifier built on support vector machine, and then it is tested using a metamorphic validation approach. The effectiveness of the approach is demonstrated by validating a deep learning classifier for automated classification of biology cell images. The proposed approach can be used for validating other deep learning framework for different applications.**

***Keywords*-deep learning; neural network; support vector machine; software validation; metamorphic testing;**

I. Introduction

Deep learning [1] is the most important breakthrough and the most promising technique in machine learning during the past decade. Its powerful ability to analyze after training with massive labelled data is evidenced by breakthroughs ranging from almost halving the error rate for image based object recognition [2] to defeating a low-level professional Go game players for the first time in late 2015 [3], which was improved further to defeat a top professional player in March of 2016.

A deep learning neural network normally has multiple hidden layers. An observation such as an image in deep learning can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc. One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction [4]. Various deep learning architectures such as convolutional deep neural networks, deep belief networks and recurrent neural networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks. However, a deep learning neural network could be very complex. For example, the relatively simple deep learning neural network AlexNet developed by Krizhevsky, Sutskever, and Hinton in 2012 [2] includes 5 convolutional layers and multiple max-polling layers in addition to 3 fully connected layers whose output of the last layer is fed to a 1000-way softmax to produce a distribution over the 1000 categories. Each convolutional layer filters each channel of the input image with multiple kernels such as the first layer has 96 and the fifth one has 256 kernels, and each fully-connected layers have 4096 neurons [2].

Due to the large number of features used for deep learning and the complexity of the deep learning architecture, the size of the training data set for deep learning is also large. The original AlexNet was trained with 1.2 million images. In order to get reasonable classification accuracy, the training data set for a deep learning normally include several hundred thousands of representative data items for each category, Therefore, it is necessary to evaluate the quality of the training data for deep learning. A deep learning execution environment such as Caffe [5], MXNet [6], Theano [7] and TensorFlow [8] may only support specific net architectures or platforms. Most of the execution environments were released during past 10 years and they are continuously updated for fixing bugs or adding new features. It is also important to evaluate the execution environment before it is applied for specific applications.

There are several widely used approaches like N-Folder Cross Validation (NFCV) and confusion matrix for evaluating the classification accuracy of machine learning algorithms. However, the focus of the approaches are on the performance of the machine learning algorithms or the net architecture. It is rare to mention how to validate the training data although they are as important as the net architecture to the accuracy of the machine learning. It is necessary to evaluate the training data, net architecture and the execution environment as a whole and provide analysis evidences for improving the learning performance. In this paper, we propose an approach for rigorously validating a deep learning framework that includes the net model, the training data, and execution environment. First, the deep learning performance is compared to the performance produced by an alternative machine learning approach with the same training and test data. The comparison

978-1-5386-0424-3/17 $31.00 © 2017 IEEE
DOI 10.1109/MET.2017.2

result serves as the baseline for further validation. Second, the deep learning performance is validated through metamorphic validation based on metamorphic processing of training data. The performance of deep learning is measured with NFCV and confusion matrix. The validation results would also serve as references for checking the validity of the deep learning execution environment. We introduce the validation approach and demonstrate its effectiveness through validating a deep learning framework for automated classifying biology cell images. The approach is a nature extension of the validation based on NFCV and confusion matrix and can be easily used for validating any deep learning framework. It provides a needed technique for ensuring the quality of deep learning especially the quality of massive training data used in deep learning and building concrete evidences for improving the performance of the deep learning.

The rest of this paper is organized as follows: Section 2 introduces the automated classification of diffraction images using deep learning and SVM. Section 3 describes the approach for validating the deep learning framework for the classification of diffraction images. Section 4 describes the related work and Section 5 concludes the paper.

II. Automated Classification of Diffraction Images

In this section, we first discuss the basic idea of morphology based cell imaging and classification, then introduce the automated classification of diffraction images using SVM and deep learning, and finally describe the metamorphic testing used for validating the deep learning framework.

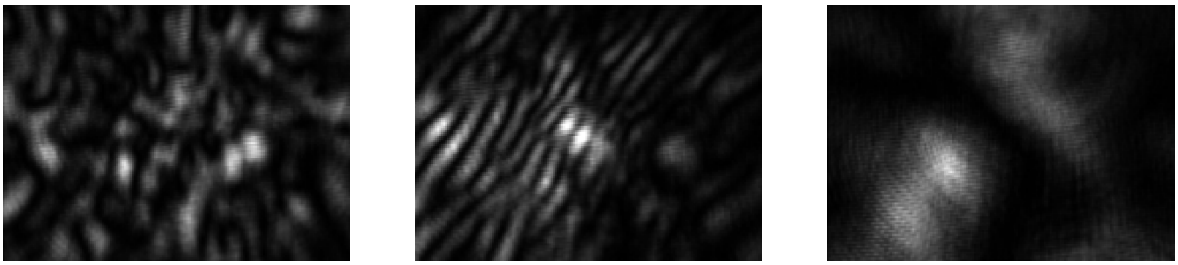
A. Morphology Based Cell Imaging and Classification

Cells are basic elements of life and possess highly varied and convoluted 3-Dimensional (3D) structures by intracellular organelles to sustain their phenotypic variations and functions. Morphology based cell classification at the single-cell level attracts intense research efforts for their direct relations to cellular functions. Besides the needs for cell staining, existing microscopy methods, however, are labor-intensive and time-consuming to acquire and analyze. A new method of polarization Diffraction Imaging Flow Cytometry (p-DIFC) has been developed by Hu and his team to acquire cross-polarized Diffraction Image (p-DI) pairs from single cells rapidly [9] [10]. 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. The p-DI data thus provide a big data source to probe the 3D morphology of the illuminated cells that requires powerful machine learning tools for extracting morphological and molecular information. To develop significant capability for rapid and accurate cell morphology assay, Ding and Hu have built a platform for enabling automated processing and analysis of massive image data for identification of the morphology fingerprints from massive diffraction cell images. The concept of morphology fingerprints arises from the correlations between the 3D morphology and diffraction patterns of light scattered by cells. Different machine learning algorithms including Support Vector Machine (SVM) [11] and deep learning were used in the platform for classifying cell types based on diffraction images [10] [12]. Fig. 1 shows three sample diffraction images acquired using p-DIFC.

In this paper, we are going to build a classifier for automated classification of diffraction images of viable cells of intact structures (simply called normal cells) from diffraction images of ghost cell bodies or aggregated spherical particles (simply called fractured cells) and cell debris or small particles (simply called debris). From Fig.1, we observe the images of normal cells include normal speckle patterns, the fractured cells include strip patterns, and the debris include large diffuse speckle patterns. Automatically removing debris and separating fractured cell images from normal cell images are important to ensure the quality of training data in machine learning, which is more important to deep learning since it needs large amount of training data. We selected 2500 diffraction images for each category, but each category may include images that are incorrectly labelled known as class label noise. The resolution of each is 8-bit at grayscale, and the size of the image is 680 * 480 pixels. We name the category of normal cell images as *cells*, the category of fractured cell images as *strips*, and the category of debris as *debris*.

B. An SVM based Classifier

We built an SVM classifier for the automated classification of the p-DIFC acquired diffraction images of cells into three categories: cells, debris and strips. SVM performs binary classification in general; however, several SVM classifiers can be combined to do multiclass classification by comparing *ones against the rest*. The SVM classification is implemented based on the textual pattern of the diffraction image, which is defined by a group of Grey Level Co-occurrence Matrix (GLCM) features [13]. The definitions of the GLCM features used for the SVM classification of diffraction images can be found in Ding and Hu's previous work [14]. We conducted an experiment study to select an optimal feature set that includes 8 GLCM features. The GLCM of each image was calculated using displacement $d = 2$ at orientations 0^o, 45^o, 90^o, and 135^o, respectively. Each GLCM feature of an image is computed on the average of the feature values of all four orientations [14]. The SVM classifier was built on an open source SVM library called LIBSVM, which has needed features such as allowing the user to set different parameters, use different kernels, and implement multiclass classification [15]. In the first round, the 7500 diffraction images were calculated for the selected GLCM features and labelled with its category, and then the feature matrix was used for training the SVM classifier. 10FCV and confusion matrix were used for checking the classification accuracy. The average accuracy of the 10FCV for the cells was 64.5%, the strips was 67.4%, and debris was 65.2%. The confusion matrix showed 27.8% cells were incorrectly classified as debris, and 6.7% were incorrectly classified as strips; 17.6% strips were incorrectly classified as cells, and 14.8% were incorrectly classified as debris; 28.9%



(a) (b) (c)

Fig. 1. A p-DIFC acquired diffraction image of (a) a viable cell of intact structures, (b) a ghost cell body or aggregated spherical particles, and (c) a cell debris or small particle.

debris were incorrectly classified as cells, and 5.9% were incorrectly classified as strips. We manually checked each of the diffraction images that were incorrectly classified during 10FCV. If an image was obviously incorrectly labelled, then the image was relabelled for the correct category. If an image was difficult to be manually classified, then the image was removed from the data set. The new data set included 2128 cell images, 2246 strip images and 2214 debris images. Then we trained the SVM classifier with the new data set and conducted 10FCV again, and the accuracy of the classification of cells, debris and strips is 84.1%, 85.3%, and 85.1%, respectively, and the confusion matrix showed incorrect rates for remained two categories were similar. Different SVM kernels have been tried and different size of the data set have been used, but around 85% was the highest classification accuracy we got for automated classification of the three categories of diffraction images using the SVM classifier. Although sophisticated image processing technique combining with other machine learning technique can be used to preprocess the images to improve the accuracy of the SVM classification as reported in Hu's previous work [12], our focus is on the validation of machine learning algorithms in general but not on a specific domain application. The SVM based classification results are used for validating the results of the deep learning classification on the similar data set.

C. A Deep Learning based Classifier

The limited number of GLCM features used in the SVM based classification are not the complete features representing the 3D morphology information captured in the diffraction image. Deep learning [4] [1] is a natural selection as an alternative to SVM. We have acquired large amount of p-DIFC images of cells and developed a deep learning algorithm of convolutional neural networks (CNNs) for automated classification of diffraction images. We utilized a CNN called AlexNet [2] and deep learning execution environment Caffe [5], to perform deep learning research on a GPU workstation with dedicated NVIDIA Tesla K40 board. AlexNet has a total of 12 layers with 5 convolution layers (CONV), 3 max-pooling layers (POOL) and 4 fully connected layers with fast rectified linear unit (ReLU) neurons. The convolutional layers have variable receptive field sizes from 5x5 to 20x20 and stride length from 1 to 10. Since our purpose is to demonstrate an approach for validating a deep learning framework but not the development of a deep learning algorithm, AlexNet was not modified for this research in case any bug is introduced into the net by the modification. The size of the input image to AlexNet is 227 * 227 pixels, but the size of a p-FIFC diffraction image is 680 * 480 pixels. Therefore, each diffraction image has to be resized to 227 * 227 pixels. In addition, several thousands diffraction images are far not enough for training a deep learning classifier. We need approaches for producing large amount of diffraction images from the original data set. Two different approaches we adopted to produce diffraction images to train the AlexNet for classifying the cell images into the three categories: cells, debris and strips. One approach is to produce many small images through pooling an original image with different sizes of the pooling windows and the stride distance. The other approach is to crop many small images from an original image in different sections. The detail of the two approaches will be discussed in Section III-C.

D. Metamorphic Testing

Metamorphic testing was first proposed by Chen [16] [17] for solving oracle problems in "non-testable" programs. A "non-testable" program is a program that is hard to find a test oracle [18]. For example, complex scientific software or artificial intelligent systems that implement sophisticate machine learning algorithms are typical "non-testable" programs. Metamorphic testing has been successfully applied for testing several domain applications such as bioinformatics systems, machine learning systems, compilers, partial differential equations solvers, large-scale databases, and online service systems. Metamorphic testing could become even more important for testing machine learning algorithms since they are becoming more and more complex. It aims at verifying the satisfiability of a Metamorphic Relation (MR) among the outputs of the corresponding MR related tests, rather than checking the correctness of the individual outputs [16] [17]. If a violation of an MR is found, the system under test (SUT) must have defects [16]. Specifically, metamorphic testing creates tests according to an MR and verifies the predictable relations among the actual outputs of the related tests. 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 software under test, but satisfiability of an MR does not guarantee the absence of defects. In this paper, metamorphic testing is used for validating a deep learning framework. We developed MRs for validating deep learning framework on three different levels: in the system level MRs are defined on alternative machine learning algorithms; in the data set level MRs are defined on metamorphic organization of the data set; and in the data item level MRs are defined on metamorphic processing of individual data items. A machine learning framework could be rigorously validated through testing the deep learning framework with MRs and their tests at the three different levels,

III. Metamorphic Validation of a Deep Learning Framework

The validation process of a deep learning framework using metamorphic testing is conducted in three levels. The validation approach is called metamorphic validation [19]. The deep learning framework to be validated includes deep learning architecture AlexNet, deep learning execution environment Caffe, and a data set consisting of diffraction images of cells, fractured cells, and debris. The MRs in system level are created based on the relation of the classification accuracy of deep learning and SVM. The MR is an extension of the definition of the classical MR for simplifying the introduction of the validation approach. The MRs in the data set level are created on the relation of the classification accuracy of reorganized training data sets. The MRs in the data item level are created on the relation of the classification accuracy of re-produced individual images. The classification accuracy is measured by NFCV and confusion matrix. The data set used in this section include majority of the images described in Section II-A, but some new images were added, and some low quality images were removed. The original data include 2232 normal cell images, 3642 debris images and 1645 strip images.

A. Validation on the System Level

We have developed an SVM classifier for automated classification of diffraction images for cells, fractured cells and debris, and the best classification accuracy for selected data set was around 85%. We hope the classification accuracy of a deep learning classifier should be equal or better than the accuracy of the SVM classifier for similar data set. Otherwise, investigation has to be conducted on the net model, the data set or the execution environment.

MR1: The classification accuracy of the deep learning classifier for the diffraction images is expected to be better than the classification accuracy of the corresponding SVM classifier.

1) Producing data set: AlexNet only accepts the image at size 227 * 227 pixels, which is much smaller than the size of the original diffraction image at 680 * 480 pixels. Caffe takes a larger image than AlexNet allows, but it cuts it into the appropriate size. The data set only has several thousands of diffraction images, which is not large enough for training a deep learning classifier. From diffraction imaging theory, we know partial of a diffraction image includes enough information of the whole image as soon as it includes enough textual section of the image. One can crop small sections from an original diffraction image to produce lots of new diffraction images at size 227 * 227 pixels. The cropping technique would resize the original image as well as produce enough training data.

The generation of new data set from the original images is automated with a Python program. First, a 5*5 pixels sliding window is defined, and the average of the intensity of each window is calculated for the whole image to find the window that has the maximal average intensity in the image. The intensity of a pixel is defined by the its pixel value. Second, choose the window that has the largest average intensity value as the center of a 227 * 227 pixels window of the image to be cropped. The cropped image is added to the new generated data set. If there are multiple sliding windows having the same largest average intensity values, the one closer to the center of the original image is chosen as the center for cropping. Third, move the 227 * 227 pixels window centering around the selected center in 8 different directions with a 5 pixels stride. At each round, 8 image instances are generated, and new instances will be produced based on last generated 8 windows following the same procedure in the second round. It can produce up to 512 new 227 * 227 pixels images from one 680 * 480 pixel image after 3 rounds of processing, and 4096 new images for 4 rounds of processing. Before an image window is cropped from the original image, it is important to ensure the whole new image is located within the original one; otherwise, the cropped image is discarded. The new generated image is labelled as the same category of the original image.

2) Training and testing: In the first experiment, the training data set of the AlexNet classifier includes 124,650 diffraction images: 40,140 normal cell images, 64,386 debris images, and 20,124 strip images. The validation data set includes 13,853 images, and the test data set includes 12896 normal cell images, 12724 debris images, and 10704 strip images. The test result shows the accuracy for classification of normal cells is 95.04%, debris is 95.5%, and strips is 66.5%. The classification accuracy of strips was not good enough according to MR1. We added additional 20,000 new strips images into the training data set to retrain the classifier, and the experiment result showed the classification accuracy of strips was significantly improved from 66.5% to 83.45%. The result shows the necessary of enough number of images needed for the training data set and the balance of the number of images in each category. We increased the number of images in the training data set to 305,580 images that includes 107,040 normal cell images, 100,156 debris images, and 98,384 strips images. The number of images in the validation data set was increased to 58,317, and the test data set was the same. However, comparing to the second round of experiment results, the classification accuracy of each category was not much improved, which was 95.34% for normal cells, 93.25% for debris, and 80.0%

for strips. The NFCV showed the similar results. The best classification accuracy of the SVM classifier is around 85% with improved data. Therefore, MR1 was not satisfied. Since the best classification of the SVM classifier was conducted on the improved data set, we should check the quality of the training data set as we did for SVM classifier. We removed the images that were incorrectly classified from the training data set in the third experiment. It was completed through the 8-fold cross validation process. The training data set, validation data set and test data set were combined and then divided into 8 equal groups/folds. At each round of experiment, the image that was incorrectly classified was removed from the data set. After we removed the incorrectly classified images, we also randomly removed some images of each category from the data set to ensure the balance of the images in each category, and finally the training data set includes 166,024 images consisting of 56,420 normal cells, 54,077 debris, and 55527 strips. The 8FCV showed the classification accuracy of cells was 97.77%, debris was 99.69% and strips was 96.24%. The error rate of the classification of each category to other two categories was almost equal.

3) Validation of the results: MR1 was satisfied in the validation, and it shows deep learning is a prefer option for building the classifier for diffraction images. However, the deep learning classifier requires resizing the original images and large amount of training data. SVM classifier takes the GLCM feature values calculated from each image directly, and the training is much faster than deep learning classifier. Cropping the diffraction images for resizing the images and producing training data for the deep learning classifier may not be feasible for other domain applications if every part of the image is important to the classification. In that case, new strategies such as pre-pooling or calculating features from original data can serve the same purpose.

B. Metamorphic Validation on the Data Set Level

The classification accuracy was validated with 8FCV and confusion matrix in the previous section. However, n-fold validation is not sufficient to validate the classification algorithm according to experiment results reported by Xie *et al.* [20]. Confusion matrix is used to describe the classification accuracy of each category and how the error rate is distributed on other categories. It is a representation of cross validation. Therefore, we developed a group of MRs on metamorphic processing of the data set to test the deep learning framework. The data set of the deep learning framework includes a training data set, a validation data set, and a test data set. The first group of MRs are built on adding data items to or removing data items from the three data sets.

MR2: Add 10% of new images into each category of the training data set should not affect the classification accuracy.
MR3: Duplicate 10% of images of each category in the training data set should not affect the classification accuracy.

The original training set include around 160,000 images as we discussed in Section III-A2, and each category includes around 55,000 images. Then around 5,500 labelled images were added into each category each time, and the classification accuracy of the classifier retrained with the three different training sets (all new images or duplicated images) was only slightly different to the result of the classifier trained with the original training data set, which is acceptable for the validation. 10% used in this MR as well as in other MRs is only an experience value.

MR4: Add 10% of images into each category of the validation data set should not affect the classification accuracy.
MR5: Add 10% of images into each category of the test data set should not affect the classification accuracy.

The 10% images added into the validation set and test data set include new images and duplicated images, the classification accuracy was almost identical to the result calculated from the original validation and test data sets. In addition, the experiments were also conducted on removed 10% of images from each data set, and the classification accuracy was almost same to the original results.

The deep learning classifier under validation is used for classifying three categories of diffraction images: cells, debris and strips. The second group of MRs are built on adding new categories to or removing some categories from the classifier.
MR6: Remove one category of the data from the data set should not affect the classification accuracy of the remaining categories.

In this experiment, we trained the classifier with the training data set and validation data set that only include two categories of the images such as cells and strips images or strips and debris images. The classifier was also tested with the test data set that only includes the two categories of the images. Three different combinations, which were normal cells and debris, normal cells and strips, and debris and strips, were validated and the classification accuracy was almost same to the original result that contained three categories. We didn't conduct an experiment for adding new categories into the data set due to the nonavailability of the data for the new category.

MR7: Add one category of the diffraction images through duplicating one existing category of data in the data set should not affect the classification accuracy.

For example, we duplicate the data set of *debris* and labelled them as *noise* category, then training the classifier with the new training data. The classification is correct if a debris image is classified as debris or noise.

C. Metamorphic Validation on the Data Item Level

In last two sections, the data set was produced through cropping images from original images. It is necessary to validate the small images used for the classification are sufficient to represent the original large images. For example, we may crop images with different sizes of sliding windows or different strides or different moving directions or even different resizing techniques. Are all of the small images produced using different methods enough to train a deep learning classifier for classifying the three different categories

of diffraction images? MRs defined on different methods for producing the small images were used for testing the deep learning framework. We expect the images produced with different methods should be good enough for the classification.

MR8: The classification accuracy of the classifier trained on the data sets cropped from original images using different stride distances should be almost the same.

MR9: The classification accuracy of the classifier trained on the data sets cropped from original images using different moving directions should be almost the same.

A small image can be produced through moving the 227 * 227 pixels window with a stride distance such as 7 pixels, 14 pixels or 17 pixels from the selected center of the original image along a direction. We produced small images with stride distances as 7 pixels, 14 pixels and 17 pixel along 8 different directions. The original data set used in Section III-A2 was created with mixed distances and all 8 directions. We separated the data set into three groups and each group only includes the small images produced with the same stride distance. New images were added to each group to ensure the number of images in the data set is same to the original one. Another data set was created using mixed distances but only 4 directions. The classification accuracy had no significant difference among the classifier trained and tested with the four different data sets. Passing MR8 and MR9 shows the small images are good enough for the classification of the three categories of diffraction images.

Cropping is not a good approach for producing new data instances if the whole image is important to the classification. In this case, we may produce new images using pooling technique. In the pooling, first a sliding window such as 5*5 pixels is created and then find the maximal (max-pooling) or average intensity value (average-pooling) of the window, and the window is downsampled to one pixel which has the intensity just calculated from the window. Through sliding the window along row and column with a stride distance to produce a 227 * 227 pixel image. Using different sizes of the sliding window such as 3*3 pixels, 5*5 pixels and 7*7 pixels combining with different stride distances such as 2 pixels, 3 pixels and 5 pixels, one can produce many small images from the original images. The sliding windows probably don't cover 100% of the whole original images since we need make sure the new produced image must be 227*227 pixels.

MR10: The classification accuracy of the classifier trained on the data set cropped from original images and the data set pooled from the original images should be almost the same.

We selected the same original diffraction images discussed in Section III-A2, and then conducted average-pooling on each image with different stride distances and sliding windows to produce similar number of images as the cropping data set. 8FCV showed the classification accuracy of cells, debris and strips was 85.7%, 98.7% and 94.3%, respectively. The classification accuracy of cells was about 11% lower than the cropping method, and the confusion matrix showed the 9.8% of normal cells were incorrectly classified as debris. The only difference between the textual pattern of normal cell images and debris images is the size of the speckles in the diffraction images, but average-pooling could mitigate the size difference of the speckles in the two categories of images. Other pooling algorithm could improve the classification accuracy.

MR11: The classification accuracy of the classifier trained on the data sets pooled from original images with different pooling functions should be almost the same.

Following the experiments for validating MR10, we developed a new data set using max-pooling functions. The classification accuracy of cells were slightly improved to 88.5%, and the accuracy for other two categories were almost the same. The pooling layers in AlexNet are max-pooling, and max-pooling might be a better choice than average-pooling for textual patterns based classification such as the classification of diffraction images.

D. Discussion

In this section, we described an empirical study of validating a deep learning framework for classifying three categories of diffraction images to explain the approach for systematically validating a deep learning framework using metamorphic testing. The metamorphic testing were conducted by checking the classification accuracy between different classification algorithms, different data sets and different data items to ensure the validity of the net architecture, the data set and the execution environment. Different to other validation approaches, the validity of the data sets and data items were explicitly checked together with checking the net architecture. Checking the validity of the data set is extremely important to ensure the quality of deep learning since it normally requires a huge amount of data, but checking the quality of the data is challenging. The approach introduced in this section could be extended for checking the data quality in other deep learning applications. The validation of the execution environment could be conducted using regular software validation approaches. We also found not all deep learning frameworks are as effective as the deep learning framework based on AlexNet and Caffe for classifying the diffraction images. For example, we have tried VGG [21] and GoogLeNet [22] (Inception V3 and V4), the classification accuracy for the same data set used in Section III-A2 was less than 60%.

IV. Related Work

N-fold cross validation and confusion matrix have been widely used for the validation of machine learning algorithms. However, NFCV and confusion matrix are good for measuring the accuracy and stability of a machine learning algorithm under validation, it is not good enough for validating the performance of the algorithm, the quality of the data set or the correctness of the implementation or the execution environment. For example, if an algorithm was implemented with some defects that cause the learning accuracy lower than the correct implementation. the defects would be difficult to be detected by NFCV and confusion matrix. If noise data items are uniformed distributed in the data set, NFCV and confusion matrix would not find the problem. Metamorphic

testing as a novel technique for alleviating oracle problem for testing "non-testable" program is a nature choice for validating machine learning algorithms. Murphy *et al.* proposed six MRs for testing machine learning algorithms [23]. The MRs are quite simple and the ability of the MRs for detecting defects probably is not better than NFCV. However, it defined a general framework for others to develop more comprehensive MRs to better testing machine learning algorithms. Xie *et al*. defined a set of MRs for validating machine learning algorithms and verification of the implementation [20]. Giannoulatou *et al*. reported a case study of verification and validation of bioinformatics software using metamorphic testing [24]. Metamorphic testing has also been used for validating simulation software [19] and scientific software [25] [26]. Comparing to existing metamorphic validation of machine learning algorithms, our approach focuses on comprehensive validation of a machine learning framework that includes the machine learning algorithms, the data set and the execution environment. The validation of the data set using metamorphic testing is the unique contribution of this paper.

V. Summary and Future Work

The metamorphic validation introduced in this paper can be used for ensuring the quality of other machine learning framework, but it is more important to deep learning frameworks since their algorithms are much more complex and they require a large amount of data. A deep learning framework normally has to be tuned for a specific domain application. Therefore, validation of a deep learning framework is often needed. Metamorphic validation is an easy to use approach. However, identification of MRs are critical to ensure the quality of metamorphic testing. In this paper, three levels of MRs that were defined on alternative machine learning algorithms, on data sets and on individual data items can validate a deep learning framework in a comprehensive way. The three levels of MRs were used for directly validating both the algorithm and the quality of the data set, and indirectly validating the execution environment. In the future, we will study metamorphic testing of the net architecture of a deep learning, in particular on the identification of MRs based on the net architecture and the prediction accuracy.

Acknowledgment

The authors would thank Jiabin Wang and Min Zhang at East Carolina University for assistances of the experiments. This research is supported in part by grants #1262933 and #1560037 from the National Science Foundation. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. Xiaojun Kang is the corresponding author of this paper.

References

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

[2] 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.

[3] E. Gibney, "Google ai algorithm masters ancient game of go," *Nature*, vol. 529, pp. 445–446, Jan. 2016.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[5] (2016, Nov.) Caffe project. [Online]. Available: http://caffe.berkeleyvision.org/

[6] (2016, Dec.) Mxnet. [Online]. Available: https://github.com/dmlc/mxnet

[7] (2016, Dec.) Theano. [Online]. Available: http://deeplearning.net/software/theano/

[8] (2016, Dec.) Tensorflow. [Online]. Available: https://github.com/tensorflow/tensorflow

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

[10] Y. Feng, N. Zhang, K. Jacobs, W. Jiang, L. Yang *et al.*, "Polarization imaging and classification of jurkat t and ramos b cells using a flow cytometer," *Cytometry A*, vol. 85, no. 11, pp. 817–826, 2014.

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

[12] 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.

[13] R. M. Haralick, K. Shanmugan, and I. H. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 610–621, 1973.

[14] 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.

[15] C.-C. Chang and C.-J. Lin. (2016) Libsvm. [Online]. Available: https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[16] T. Y. Chen, S. C. Cheung, and S. Yiu, "Metamorphic testing: a new approach for generating next test cases," Tech. Rep. HKUST-CS98-01, Dept. of Computer Science at Hong Kong Univ. of Science and Technology, 1998.

[17] S. Segura, G. Fraser, A. Sanchez, and A. Ruiz-Cortés, "A survey on metamorphic testing," *IEEE Trans. on Software Engineering*, vol. PP, no. 99, pp. 1–1, 2016.

[18] M. D. Davis and E. J. Weyuker, "Pseudo-oracles for non-testable programs," in *Proceedings of the ACM '81 Conference*, 1981, pp. 254–257.

[19] M. Olsen and M. Raunak, "Metamorphic validation for agent-based simulation models," in *Proc. of the Summer Computer Simulation Conference*, 2016, pp. 33:1–33:8.

[20] X. Xie, J. W. K. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Y. Chen, "Testing and validating machine learning classifiers by metamorphic testing," *Journal of System and Software*, vol. 84, no. 4, pp. 544–558, Apr. 2011.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[23] C. Murphy, G. Kaiser, L. Hu, and L. Wu, "Properties of machine learning applications for use in metamorphic testing," in *Proc. of 20th Intl. Conf. on Software Engineering and Knowledge Engineering*, 2008, pp. 867–872.

[24] 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.

[25] 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.

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