10th International Conference on Intelligent Control and Information Processing
Marrakesh, Morocco, December 14-19, 2019

A Spectral Feature Based CNN Long Short-Term Memory Approach for Classification

[1]Juan F. Ramirez Rochac and [2]Nian Zhang
[1]Dept. of Computer Science & Information Technology,
[2]Dept. of Electrical and Computer Engineering
University of the District of Columbia
Washington, D.C., USA 20008
Email: jrochac@udc.edu, nzhang@udc.edu

Jiang Xiong
College of Computer Science and Engineering
Chongqing Three Gorges University
Chongqing, China 404000
Email: xjcq123@sohu.com

***Abstract*—This paper presents a Gaussian data augmentation-assisted deep learning using a convolutional neural network (PCA18+GDA100+CNN LSTM) on the analysis of the state-of-the-art infrared backscatter imaging spectroscopy (IBIS) images. Both PCA and data augmentation methods were used to preprocess classification input and predict with a comparable degree of accuracy. Initially, PCA was used to reduce the number of features. We used 18 principal components based of the cumulative variance, which totaled 99.92%. GDA was also used to increase the number of samples. CNN-LSTM (long short-term memory) was then used to perform multiclass classification on the IBIS hyperspectral image. Experiments were conducted and results were collected from the K-fold cross-validation with K=20. They were analyzed with a confusion matrix and the average accuracy is 99%.**

***Keywords*— convolutional neural networks; principal component analysis; data augmentation; multiclass classification**

I. INTRODUCTION

Deep learning approaches have been applied in image recognition for years while it is gaining more attention in the research community over time due to the robustness and wide applicability [1]. Recently there is a huge shift from traditional neural network approach to deep neural network, convolutional neural network (CNN), and many hybrid algorithms [2]-[3]. A deep neural network (DNN) is based on an artificial neural network with deep layer structure. Its input layer is represented as a composition of primitives. The extra layers make use of features from lower layers, thus potentially reduce the number of units needed for modeling complex data than a shallow network.

In deep learning, CNN known as ConvNet is one way to solve the issue with DNN using convolution rather than matrix multiplication [4]. Convolutional Neural Network (CNN) has gotten admirable performance in the domain of image recognition [5]. It is a common knowledge that a deep learning based algorithm would be more effective when accessing more training data. Previous studies have demonstrated the effectiveness of data augmentation through minor modifications to the available training data, such as image cropping, rotation, and mirroring [6]. However, these transformations are not suitable for the infrared backscatter images, because any cropping, rotation, and mirroring of the original image might generate a different signature that is associated with a different material.

In this paper, we present a novel data augmentation technique that determines the suitable amount of white Gaussian noise (AWGN) to be added to the original data samples based on the desired signal-to-noise ratio (SNR). We study how Gaussian data augmentation techniques would impact the performance of a CNN for infrared backscatter images classification. The rest of this paper is organized as follows. In Section 2, the infrared backscatter imaging spectroscopy (IBIS) technique and the convolutional neural networks (CNN) architecture are introduced. In Section 3, a near lossless PCA is introduced. A Gaussian data augmentation method which can increase the number of samples 100 times is presented. The CNN-LSTM (long short-term memory) for multiclass classification is described. In Section 4, the experimental results are demonstrated. In Section 5, the conclusions are given.

II. BACKGROUND

A. *Infrared Backscatter Imaging Spectroscopy (IBIS) for Standoff Chemical Detection*

Infrared backscatter imaging spectroscopy (IBIS) system utilized a quantum cascade laser (QCL) as an illumination source and collects the reflection of laser beam onto a focal plane array. Spectroscopy is implemented by continuously tuning the wavelength of the QCL. The signals in the collected frames are binned and turned into a hypercube image. Spectra are generated from the IBIS hypercube images by obtaining the signal from a spot of interest from every frame in the image stack.

The synthetic data set was composed of 40 different analytes on various different substrates. For each of the 450

This work was supported in part by the DoD grant, #W911NF1810475, National Science Foundation (NSF) grants, #1505509, #1533479, and #1654474.

permutations of analyte, substrate, a range of mass loading levels were produced, bringing the total number of spectra in the training set to 18,000. Various amounts of Gaussian noise were also added to each spectrum to further simulate experimental data. Since the existing CNNs we chose were originally designed for image recognition and classification, the spectra in the training set had to be presented as images. Fig. 1 is a plot of the spectrum of avobenzone, an active ingredient used in sunscreen products to absorb the full spectrum of UVA rays.

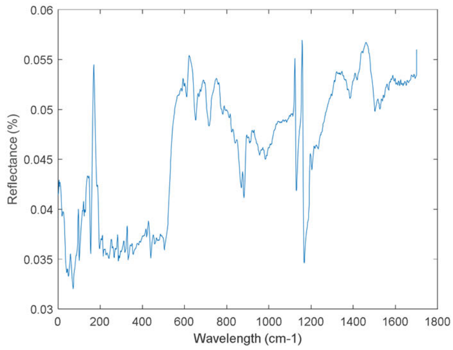

Fig. 1. Example representations of spectral image format used to train and test a CNN. The model spectrum plotted as a simple line in white space

A colorized image of the identical simulated spectrum by applying a series of high- and low-pass filters with different cutoff frequencies to the raw spectrum and transforming all of those values into an RGB color scale is represented in Fig. 2. To make these results fit better into a square image file, each spectrum was cut into three segments and placed one below the other. As a result, Fig. 2 contains three horizontal segments in each of which the center horizontal line carries the raw spectrum, the rows above carry low pass filtered versions and the rows below carry high pass filtered versions. The training data images presented in the form illustrated in Fig. 2 attempt to utilize the full frame of the image (as well as the dynamic range contained in the color scale) sent through the CNN as opposed to leaving a significant amount of white space as seen in Fig. 1.

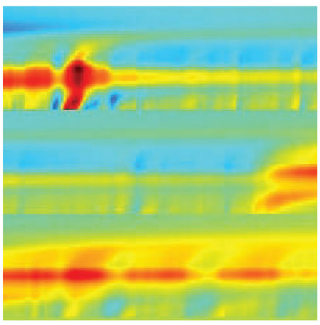

Fig. 2. An RGB color scaled intensity plot of the same spectrum run through high and low band pass filters and segmented into three horizontal regions stacked vertically

B. *Convolutional Neural Networks (CNN) Architecture*

CNN also known as ConvNet is one way to solve the issue with DNN using convolution rather than matrix multiplication [7]. * denotes convolution between two functions *x(t)* and *w(a)* and symbolized as *S(t)* in (1), which is also known as convolution integral in literature. The CNN architecture incorporates local receptive fields to ponder the spatial information, shared weights, and pooling to consider the summary statistics in the output. A two-dimensional CNN for image *(I)* recognition with kernel *K* is defined in (2) where *m, n* are image dimensions and *i, j* are kernel parameters (Fig. 3).

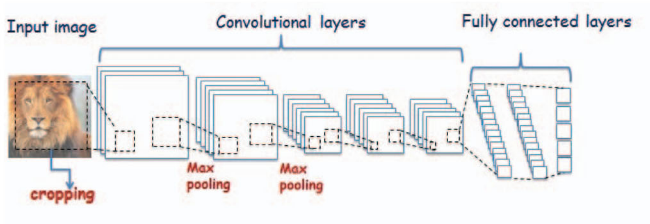

Fig. 3 Simplified architecture of DNN

$$S(t) = (x * w)(t) = \int_{-\infty}^{\infty} x(a)w(t-a)da \quad (1)$$

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \quad (2)$$

C. *Principal Component Analysis*

Principal component analysis (PCA) is a popular and effective dimensionality reduction method that can be applied before the classification task [8]. It aims to reduce the number of irrelevant features while maintaining the most important features and variability [9]. In addition, PCA can also remove outliers and anomalies from the raw or original data [10].

D. *Data Augmentation*

Data augmentation is a well-known method to increase the number of sample size by using different mathematical transformations [11]. A derived image can be obtained by cropping, rotating or mirroring the original image. In addition, we can add noise to the original images, or increase the contrast on the original image. Some examples are illustrated in Fig. 4. The left most image is the original image. The other six images are derived from the original image using the above transformations.

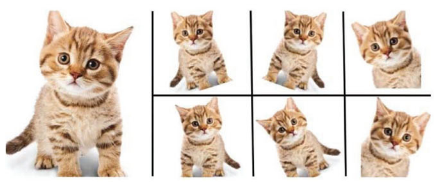

Fig. 4 Data augmentation transformation

III. Methodology

Our methodology has three components followed by an evaluation component. First, we present a lossless principal component analysis. Then, we explain our noise-based data augmentation technique. Third, we present a deep learning classifier for multiclass classification based on convolution. And finally we list the performance metrics that will be used to evaluate our approach.

A. *Lossless Dimensionality Reduction*

The first component of our methodology is a PCA variant. A near lossless PCA is used to reduce the high number of features with the idea to compute only necessary eigenvectors to avoid losing information. To find the minimum number of eigenvectors or principal components needed to represent the majority of the spectra data, we perform a cumulative variance analysis on the original data points. This analysis is further explained in the Experimental Results section.

B. *Gaussian Data Augmentation*

After PCA, the next component of our methodology is a data augmentation (DA) variant. A noise-based DA is used to increase the number of samples. Our proposed data augmentation (GDA) method adds white noise (AWGN) to the original image to produce derived images [12]. Let the derived data, original data and the effect of the noise, be random variables: $Y(t)$ $X(t)$ and $N(t)$, respectively, where $N(t)$ is a weighted probability distribution. This weighted random signal is a Gaussian distribution, see Equation 4, with mean equal to zero and standard deviation equal to sigma, $N(\mu,\sigma)$ where μ is mean and σ is standard deviation. In our approach, we derive the value of σ using the signal-to-noise ratio (SNR_{dB}) value in Equation 5.

$$Y(t) = X(t) + N(t) \tag{3}$$

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{4}$$

$$SNR_{dB} = 20 \cdot \log_{10}\frac{\mu_{signal}}{\sigma_{noise}} \tag{5}$$

What makes our approach different from traditional data augmentation using noise it that our noise is not just random. Instead, we employ an AWGN to generate our noise. AWGN will generate more derived samples that are closer to the original samples (within one sigma) and less derived samples that are far from the original samples while random/uniform noise will generate derived samples without any priority. Fig. 5 depicts the original signal in red and the signal with AWGN when SNR is 20.

For instance, let us use DA to generate 10 derived samples for each original sample. Given the original integer value of 25 labeled as class C, a noise-based DA will generate: 23,23,24,24,25,25,26,26,27,27, where all derived values have the same probability. A classifier will learn that 23, 24, 25, 26 and 27 belong to the same class C. Now, our GDA will generate: 23,24,24,25,25,25,25,26,26,27, where derived values have a probability based on the closeness to the original value. This probability is the one that we want our deep learning classifier to learn. So that, a classifier will learn that 25 definitely belongs to class C, that 24 and 26 probably belong to class C, and finally that 23 and 27 might or might not belong to class C.

To generate derive samples based on the degree of closeness to the original samples, we need to determine a target SNR. We will investigate how much loudness can our noise be and yet accurately perform classification for a particular SNR. Preliminary experiments were used to study the effect of different level of loudness to the classification accuracy using the PCA10, GDA10 and Epoch = 5 in a basic CNN. The results are shown in Table I. SNR denotes the difference in loudness between the original signal *X* and the noise *N*. For example, SNR=0 means that there is no difference between the loudness of the signal *X* and the loudness of the noise *N*. SNR=20 means that the signal *X* is twenty decibels louder than the noise *N*. SNR=100 means that the signal *X* is one hundred decibels louder than the noise *N*. Hence, as the signal *X* overpowers the noise *N*, the noise become unnoticeable and the derived signal *Y* becomes equal to the original signal *X* which causes overfitting in our model.

TABLE I. Signal-to-Noisy Analysis

SNR (dB)	Training (%)	Validation (%)	Test (%)
0	8.49	8.84	0.00
20	50.39	54.59	45.97
40	85.14	90.46	87.81
60	90.55	95.47	92.23
80	**90.93**	**95.95**	**93.41**
100	90.76	95.77	92.43
120	90.50	95.25	92.02
140	91.13	95.55	92.37
160	91.16	95.35	91.20
180	91.12	95.30	92.43
200	91.03	95.75	92.13

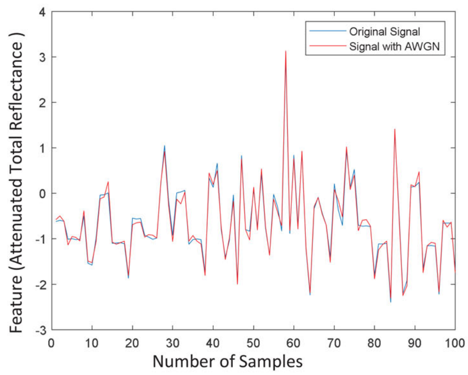

Fig. 5. Comparison of the original signals and the AWGN signals

C. *Convolutional Neural Networks*

After PCA and GDA, the third component of our methodology consist of a Convolutional Neural Networks

(CNN) classifier. CNN is used to learn from the pre-processed data samples. Fig. 6 depicts the dataflow from dimensionality reduction to data augmentation to deep learning classification. Our experiments were implanted on the PCA+GDA+CNN model as shown in Fig. 6. In this model, input data will be reduced using PCA first and will be augmented using GDA before using deep learning neural networks. The CNN-LSTM hyperparameters are listed in Table II.

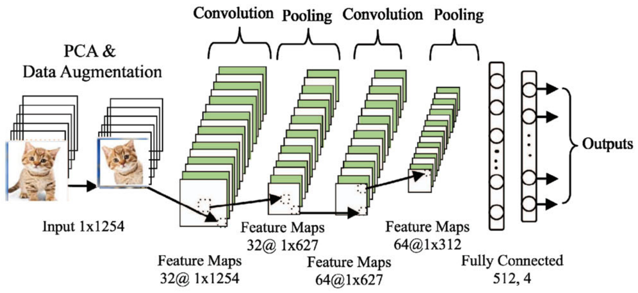

Fig 6. Overview of PCA+GDA+CNN

TABLE II. CNN-LSTM HYPERPARAMETERS

Hyper-parameter	Value
Convolutional Layer	[10, 10, 32, 128]
Strides of Convolutional Layers	[5, 5]
Padding	Same
Activation	ReLU
Long-Short Term Memory Layers	[10, 5, 64], stateful=T
Fully Connected Layer	3104 × 12352 × 20
Activation	Softmax

The experiments were implemented using Python, Keras [13] and TensorFlow. Each iteration has 20 Epochs and the batch size is ten. The platform was a state-of-the-art NVIDIA Tesla k80 GPU accelerator consisting of 2496 CUDA cores and a main memory of 12GB GDDR5.

D. *Performacne Metrics*

Fig. 7 is a representation of the data partition for a five-fold cross-validation, where 20% of the entire original dataset is used for testing and the remaining 80% is used for learning/fitting the model. In order to diminish overfitting, each fold will work in a round robin fashion. During the fitting/learning phase, that remaining 80% is augmented using GDA and further divided into two sets: one for deep learning training (75%), and one for validation (25%). The test samples are used in the prediction phase and a confusion matrix is computed to compare overall accuracy.

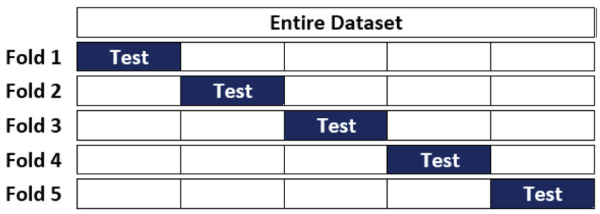

Fig 7. Five-fold Cross-Validation Partitions

IV. EXPERIMENTAL RESULTS

The goal is to measure performance improvement in multiclass classification accuracy using our proposed system: PCA18+GDA100+CNN-LSTM. To this end, first we define each steps of our system. This is followed by a 5-fold cross-validation on three different datasets. The block diagram of the proposed approach is shown in Fig. 8.

Fig 8. Block diagram of the proposed Gaussian data augmentation approach

Step 1: PCA18

We found 18 principal components have represented 99.92% of the data variability based of the cumulative variance analysis, as shown in Fig. 9. Therefore, the 18 most significant eigenvectors for the spectra from each pixel in the image were calculated. Next, all the spectral points were projected along these axes.

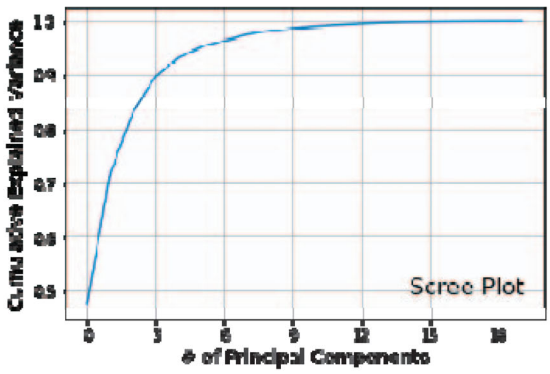

Fig 9. Scree Plot

The Cumulative Explained Variance values for the first 20 Principal Components are in Table III:

TABLE III. CUMULATIVE EXPLAINED VARIANCE

#PCs	CumVar	#PCs	CumVar
1	47.66%	11	99.04%
2	71.66%	12	99.32%
3	83.12%	13	99.52%
4	89.62%	14	99.69%
5	93.24%	15	99.79%
6	95.12%	16	99.85%
7	96.38%	17	99.89%
8	97.49%	**18**	**99.92%**
9	98.19%	19	99.94%
10	98.70%	20	99.95%

Step 2: GDA100

In the second step, GDA is used to increase the number of samples. We generated 100 derived images per sample using a SNR equal to 80dB. This means that we increased the number of samples 100 times.

Step 3: CNN+LSTM

In the third and final step, CNN-LSTM (long short-term memory) is used to perform multiclass classification. This deep learning model consists of two types of neural networks. CNN is a convolutional neural network which consists of at least one convolution kernel with a *ReLU* activation function. LSTM is a recurrent neural network which consists of at least one long-short term memory layer with an activation function that uses *tanh*. The complete architecture is shown in Fig. 10.

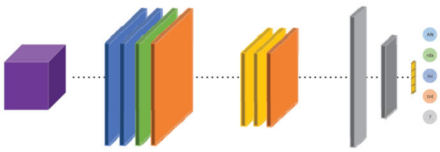

Fig 10. *CNN-LSTM* architecture

The pre-processed input data is shown as a purple cube. The convolutional layers are shown in blue, the pooling layers in green, and the dropout layer in orange. The core LSTM layers are shown in yellow. Fully-connected layers or dense layers are shown in gray. A *softmax* activation function is used for multiclass classification.

We use an IBIS data set to explore the capability of the proposed method. This data set consists of 40 analytes on various different substrates. For each of the 450 permutations of analyte, substrate, a range of mass loading levels were produced, bringing the total number of spectra in the training set to 18,000. The data set is divided by multiple batches, with the batch size of 10. 60% of the data are randomly chosen as the training set, 20% are used a test data, and 20% are validation data.

For 60% training data samples and 20% validation data samples, Fig. 11 and Fig. 12 show the accuracy and loss in percentage as a function over 20 epochs.

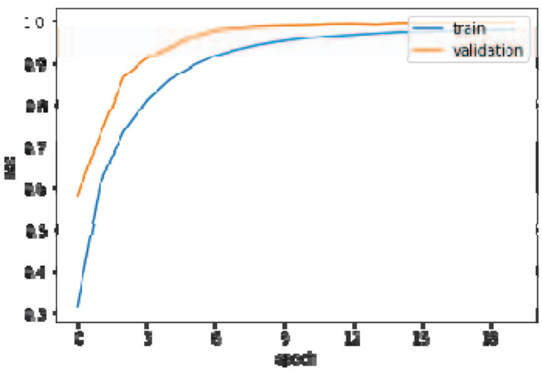

Fig 11. Accuracy of PCA18+GDA100 +CNN-LSTM

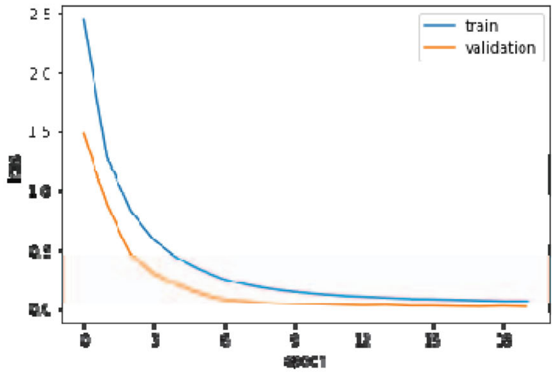

Fig 12. Loss of PCA18+GDA100 +CNN LSTM

For the 20% test data samples, Fig. 13 shows the confusion matrix for PCA18+ GDA100+CNN-LSTM on 40 different analytes. The real class labels are on the vertical axis while the predicted class labels are on the horizontal axis. The intensity of the shade of blue denotes the number of samples that were labeled as that class. Less intensity or lighter blues denote less number of samples. Higher intensity or darker blues denote greater number of samples. Therefore, the perfect classifier will only depict high intensity blues on the main diagonal.

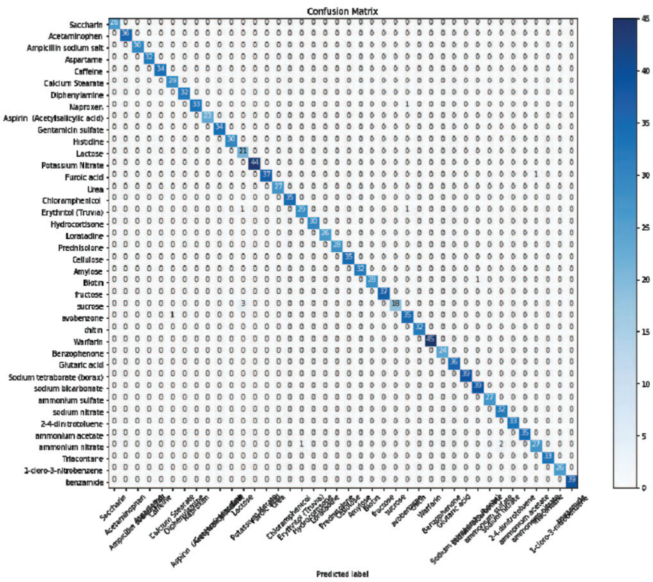

Fig. 13. Confusion Matrix

Note that in Fig. 13, we cannot visualize any blue squares under or above the main diagonal, suggesting that accuracy is near perfection. Quantitatively, our PCA18+GDA100+CNN-LSTM approach achieved 99% accuracy. To rule out overfitting on training, validation and test data, cross-validation was applied.

Table IV presents in detail the results of our performance measurements for our PCA18+GDA100+CNN-LSTM system on 40 different analytes (classes). Results were collected from our twenty-fold cross-validation. *Precision* indicates the number of accuracy labeled test samples. *Support* represents the number of samples for a particular class inside the test set and the weighted-average is based on the total support per class.

TABLE IV. CLASSIFCATION REPORT

PCA18+GDA100 +CNN LSTM (Accuracy 99%)				
Classes	**Precision**	**Recall**	**F1-score**	**Support**
Saccharin	1.0	1.0	1.0	26
Acetaminophen	1.0	1.0	1.0	36
Ampicillin sodium salt	1.0	1.0	1.0	30
Aspartame	1.0	1.0	1.0	32
Caffeine	1.0	1.0	1.0	34
Calcium Stearate	0.97	1.0	0.98	29
Diphenylamine	1.0	1.0	1.0	32
Naproxen	1.0	0.97	0.99	34
Aspirin	1.0	1.0	1.0	23
Gentamicin sulfate	1.0	1.0	1.0	34
Histidine	1.0	1.0	1.0	30
Lactose	0.84	1.0	0.91	21
Potassium Nitrate	1.0	1.0	1.0	44
Furoic acid	1.0	0.97	0.99	38
Urea	1.0	1.0	1.0	27
Chloramphenicol	1.0	1.0	1.0	35
Erythritol	0.97	0.94	0.95	31
Hydrocortisone	1.0	1.0	1.0	30
Loratadine	1.0	1.0	1.0	26
Prednisolone	1.0	1.0	1.0	28
Cellulose	1.0	1.0	1.0	35
Amylose	1.0	1.0	1.0	32
Biotin	1.0	0.97	0.98	29
fructose	1.0	1.0	1.0	37
sucrose	1.0	0.86	0.92	21
avobenzone	0.95	0.97	0.96	36
chitin	1.0	1.0	1.0	32
Warfarin	1.0	1.0	1.0	45
Benzophenone	1.0	1.0	1.0	24
Glutaric acid	1.0	1.0	1.0	36
Sodium tetraborate	1.0	1.0	1.0	39
sodium bicarbonate	0.97	1.0	0.99	39
ammonium sulfate	1.0	1.0	1.0	27
sodium nitrate	0.94	1.0	0.97	32
2-4-dinitrotoluene	1.0	1.0	1.0	33
ammonium acetate	1.0	1.0	1.0	35
ammonium nitrate	0.96	0.9	0.93	30
Triacontane	1.0	1.0	1.0	33
1-cloro-3-nitrobenzene	1.0	1.0	1.0	26
benzamide	1.0	1.0	1.0	39
Accuracy	**0.99**	**0.99**	**0.99**	**1280**
macro avg	**0.99**	**0.99**	**0.99**	**1280**
weighted avg	**0.99**	**0.99**	**0.99**	**1280**

V. CONCLUSIONS AND FUTURE WORK

A Gaussian data augmentation-assisted deep learning using a convolutional neural network (PCA18+GDA100+CNN LSTM) was designed and explored on the state-of-the-art infrared backscatter imaging spectroscopy (IBIS) datasets. Both PCA and data augmentation methods were used to preprocess classification input and predict with a comparable degree of accuracy. Initially, PCA was used to reduce the number of features. We used 18 principal components based of the cumulative variance, which totaled 99.9%. This means that 18 PCs are enough to represent most of the data variability. GDA was also used to increase the number of samples. We generated 100 derived images per sample. This means that we increased the number of samples 100 times. CNN-LSTM (long short-term memory) was then used to perform multiclass classification on the IBIS hyperspectral image.

Experimental results were compiled from the cross-validation with twenty runs each. They were analyzed with a confusion matrix and the average accuracy is 99%. In the future, we will apply the proposed PCA18+GDA100+CNN-LSTM on the noisy infrared backscatter imaging spectroscopy (IBIS) and compare the performance with the state-of-the-art deep learning approaches.

REFERENCES

[1] G. Zhong, X. Ling, and L. Wang, "From shallow feature learning to deep learning: benefits from the width and depth of deep architectures," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 9 (1), Jan/Feb 2019.

[2] G. Zhong, K. Zhang, H. Wei, Y. Zheng, and J. Dong, "Marginal deep architecture: stacking feature learning modules to build deep learning models," IEEE Access, vol. 7, pp. 30220 - 30233, March 2019.

[3] H. Zou, H. Zhang, X. Li, J. Liu, and Z. He, "Generation textured contact lenses iris images based on 4DCycle-GAN," in Proceedings of the IEEE 24th International Conference on Pattern Recognition (ICPR), pp. 3561-3566, 2018.

[4] A. Gulli, and S. Pal, "Deep Learning with Keras," 1st ed, Packt Publishing, 2017.

[5] K. Wu, X. Wang, and M. Li, "Better Performance of Memristive Convolutional Neural Network Due to Stochastic Memristors," The 16th International Symposium on Neural Networks (ISNN 2019), Moscow, Russia, July 10-12, 2019.

[6] L. Perez, J. Wang, "The Effectiveness of Data Augmentation in Image Classification Using Deep Learning. arXiv preprint arXiv:1712.04621, 2017.

[7] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.

[8] I. T. Jolliffe, "Principal Component Analysis," Springer-Verlag, 2nd Edition, October, 2002.

[9] M. A. Carreria-Perpinan, "A review of dimension reduction techniques," Technical Report CS-96-09, Dept. of Computer Science, University of Sheffield, January 1997.

[10] I. K. Fodor, "A survey of dimension reduction tehniques,"Technical Report UCRL-ID 148494, LLNL June 2002.

[11] E. Jannik Bjerrum, M.Glahder, T.Skov, "Data augmentation of spectral data for convolutional neural network (CNN) based deep chemometrics," Cornell University, arXiv.org, 2017.

[12] Juan F. Ramirez Rochac, Lara Thompson, Nian Zhang, and Timothy Oladunni, "A Data Augmentation-assisted Deep Learning Model for High Dimensional and Highly Imbalanced Hyperspectral Imaging Data", The 9th International Conference on Information Science and Technology (ICIST 2019), Hulunbuir, Inner Mongolia, China, 2019.

[13] F. Chollet, et al.: Keras. https://github.com/fchollet/keras, 2015.