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

A Gaussian Data Augmentation Technique on Highly Dimensional, Limited Labeled Data for Multiclass Classification Using Deep Learning

Juan F. Ramirez Rochac and Lily Liang
Dept. of Computer Science & Information Technology
University of the District of Columbia
Washington, DC, USA
Emails:{jrochac, lliang}@udc.edu

Nian Zhang
Dept. of Electrical & Computer Engineering
University of the District of Columbia
Washington, DC, USA
Email: nzhang@udc.edu

Timothy Oladunni
Dept. of Computer Science & Information Technology
University of the District of Columbia
Washington, DC, USA
Email: timothy.oladunni@udc.edu

***Abstract*—In recent years, using oceans of data and virtually infinite cloud-based computation power, deep learning models leverage the current state-of-the-art classification to reach expert level performance. Researchers continue to explore applications of deep machine learning models ranging from face-, text- and voice-recognition to signal and information processing. With the continuously increasing data collection capabilities, datasets are becoming larger and more dimensional. However, manually labeled data points cannot keep up. It is this disparity between the high number of features and the low number of labeled samples what motivates a new approach to integrate feature reduction and sample augmentation to deep learning classifiers. This paper explores the performance of such approach on three deep learning classifiers: MLP, CNN, and LSTM. First, we establish a baseline using the original dataset. Second, we preprocess the dataset using principal component analysis (PCA). Third, we preprocess the dataset with PCA followed by our Gaussian data augmentation (GDA) technique. To estimate performance, we add k-fold cross-validation to our experiments and compile our results in a numerical and graphical using the confusion matrix and a classification report that includes accuracy, recall, f-score and support. Our experiments suggest superior classification accuracy of all three classifiers in the presence of our PCA+GDA approach.**

***Keywords*—Deep Learning, high dimensionality, limited labeled data, Principal Component Analysis, Gaussian Data Augmentation, Multiclass Classification**

I. INTRODUCTION

In recent years, oceans of datasets paved the path for expert-level performance in deep learning models. Deep learning has flooded journals and conferences with new approaches and applications. In the last decade deep machine learning has developed expert-level accuracy in image classification [1]-[3]. In [4], convolutional neural networks (CNNs) are able to produce dermatologist-level accuracy in skin cancer diagnosis. Moreover, [5] demonstrated the effectiveness of deep machine learning algorithms to the point of achieving ophthalmologist-level accuracy in the detection of referable diabetic retinopathy (RDR).

Deep learning is a growing family of models. At its core these models enable a system to automatically uncover hidden patterns needed for multiclass classification from the data without any preprocessing. Deep machine learning is nowadays the state-of-the-art classification technique with proven applications in a variety of fields from security monitoring to remote sensing to computer-aided prognosis, and from image to voice recognition. Applicability of deep machine learning algorithms is fueled by data. The more data, the better learning. However, hyperspectral datasets put a stiff told on deep learning models, due to its dimensionality and imbalance nature of the datasets.

Principal Component Analysis (PCA) presents one of the most commonplace tools [6]-[8] to deal with highly dimensional data. The work in [9]-[10] present techniques that extract features adaptively using local and global information and deal with imbalance hyperspectral cubes. But, arbitrary selection of PCs may result in a poor representation and lack the data variability needed to accurately perform classification. Second, Data Augmentation (DA) is a commonplace tool to expand the limited number of training samples [11]-[13] present successful implementations of DA in the fields of image processing and molecular modelling. Together PCA and DA thus prove a promising option to deal with the composite problem of both high dimensionality and limited labeled datasets in DNN. However, regular DA cannot be applied to PT-IS datasets, because any distortion of the original image would result in a distortion of the signature of the underling substance.

In this paper, we study how much a two-folded technique (PCA+DA) impacts the performance of three deep machine learning models, namely, Multilayer Perceptron (MLP), Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). The goal is to achieve better classification results with limited training data and thus reduce the need for more data. We also propose a Data Augmentation variant (GDA) which integrates signal-to-noise ratio (SNR) with an Additive white Gaussian noise (AWGN) to generate derived

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

978-1-7281-0015-9/19/$31.00 ©2019 IEEE

data samples suited for multiclass classification in various deep neural networks models. This paper continues with a description of the PT-IS dataset. Then, Section 3 presents each component of our proposed methodology and performance metrics. In Section 4, you may find the results of our experiments. And, in Section 5, the paper closes with final remarks and future directions.

II. DATASET

The dataset was collected and normalized in [14]. The collection apparatus consisted of two parts. First, the substance or analyte samples are illuminated with a quantum cascade laser. Second, hundreds of images are collected with an infrared camera. The collected images correspond to multiple infrared wavelength channels. Fig. 1 shows the entire system, from data collection by law enforcement officers and military personnel to detection and classification.

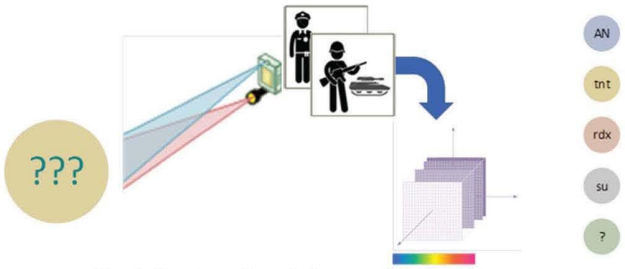

Fig. 1. Quantum Cascade Laser and Infrared Camera

By using this two-part apparatus, we collected a three-dimensional array of images, which we called a hyperspectral cube, as shown in Fig. 2. For each pixel in the front image there are not only RGB values, but also layers corresponding to different infrared intensities. With this cube, it is possible to identify certain analytes and substrates based on the unique pattern of intensities. This pattern is called fingerprint or spectral signature. By matching this spectral signature, the underlying material or element can be uniquely identified.

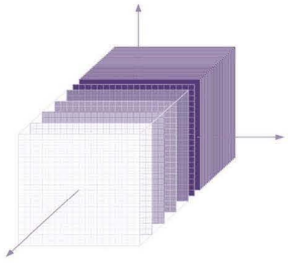

Fig. 2. Hyperspectral Cube

The matching process is possible on pure pixels because they consists of only one material. Classification or detection of these types of pixels is achieved by simply matching its corresponding analyte signature. But without the proper capturing resolution, cameras may lack the granularity or finesse to produce pure pixel images. Thus, single pixels can capture multiple signatures. The result pattern of intensities in each pixel will be a combination of two or more individual pure signatures.

The dataset is highly-dimensional (1254 features) and limited labeled (only 123 samples). It consists of eight classes, explosive analytes and non-explosive substrates. We focus on the multiclass classification problem of four analytes: AN, RDX, Sucrose and TNT. For further reading on the collection and normalization of this dataset, refer to [14].

III. METHODOLOGY

The proposed methodology is threefold. We first apply PCA. Second, we proposed a novel DA technique. Third, we explore the effects on various deep learning models for multiclass classification.

A. *Principal Component Analysis*

This section considers a well-known technique for dimensionality reduction. The main goal of using PCA is to reduce the high number of features yet maintain the majority of the variability [6]. PCA is a popular, widely implemented technique to reduce highly dimensional datasets into more manageable datasets, by remapping each data points from a high dimensional space into a lower dimensional space [7]. Data scientists find PCA very appealing mainly because PCs tend to represent the most important patterns first while removing anomalies and outliers [8].

Fig. 3 depicts a collection of data points on its original XY space as dots and tiny triangles on the left hand side. And it also depicts the first and second principal components as lines on the right hand side. Note that these principal components are orthogonal to one another and together they manage to capture and better represent the variability of all sample points.

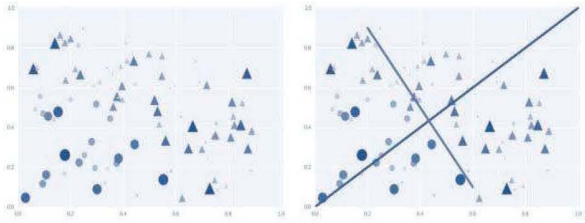

Fig. 3. Principal Component Analysis

The mathematics behind this dimensionality reduction technique are based covariance. Let $\mathbf{D}[n][m]$ be a matrix that holds the raw data, where the number of samples is represented by the index n and the number of features is represented by the index m. Assume D has zero mean. Then, let **COV(D,P)** be the expected value of $d_i * d_j$ which is an orthonormal transformation matrix and $\mathbf{P}[n][k]$, where the number of new features is represent by the index k and k is usually less than m and less than n.

Fig. 4 presents the Scree Plot for our dataset. The Scree Plot depicts the cumulative variance in percentage over the number of principal components. In Fig. 4 we observe that eighteen PCs (k=18) are sufficient to model almost all the variability in our dataset. Therefore, during our experiments, only the first 18 PCs were used.

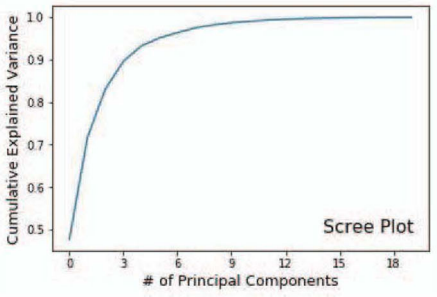

Fig 4. Scree Plot

B. *New Data Augmentation Method*

Traditional data augmentation (DA) has showed its capability to improve the performance of image classification with better levels of accuracy [11]. It is used to expand the limited samples to large-scale data set by generating additional data. These synthetic data points are the result of a mathematical transformation of all or some features. And, they automatically inherit the original label.

Equations, kernels and/or a rule-based mapping allows us to derive and augment the original set of images [12]. For each original image, we automatically generate multiple derived images. Each image is a different version or variant of the original image. By stretching, shrinking, zooming in, zooming out, flipping, rotating, or applying a different color or contrast, DA augments the original dataset size. Then, the original and derived images are used in the learning stage. However, there are two basic ways to transform the images: 1) using affine transformations or 2) using pixel-wise transformations to generate a derived images. Affine transformations consist of flipping, rotating or shifting the original image. Pixel-wise transformations consist of adding noise, increasing the contrast or/and blur on the original image. Fig. 5. Presents a summary of these DA transformations. On the top left corner, we have the image. The middle row shows higher contrast, blur and edges only. The bottom row depicts the flip transformation and it also shows the effects of stretching and shrinking the image over the horizontal axis. All nine images portray a person in front of a computer regardless of the transformation.

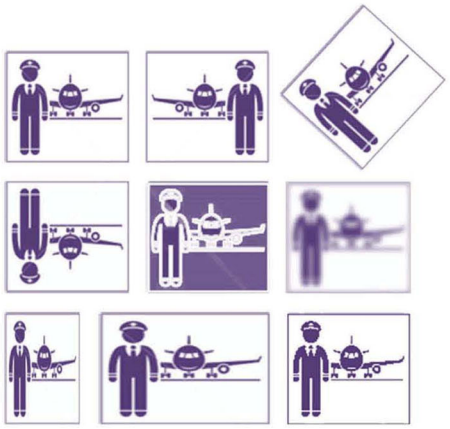

Fig. 5. Data Augmentation Transformations

Unfortunately, the affine transformations presented in Fig. 5 cannot be directly applied to raw hyperspectral cubes or principal components because any remapping of the original features would result in a different hyperspectral signature and a mismatching class label.

To solve this problem, we propose a Gaussian based DA (GDA) technique in which we use an additive white Gaussian noise (AWGN) to generate new images. Inspired by the communication systems concept, the received signal, *x(t)* will be represented as the sent or original signal, y*(t)* contaminated with some noises, *n(t)* (Equation 1) [15]. Both the *x*(*t*) and *y*(*t*) which are random signals while *n*(*t*) follows a Gaussian distribution with mean equal to zero (Equation 2) and a parametrized variance σ^2 which is calculated using the signal-to-noise ratio (SNR). Equation 3 presents the formula for SNR. We performed a set of experiments to investigate the impact of SNR on the training accuracy of the PCA18 (PCA with 18 principal components) + GDA10 (Gaussian data augmentation with 10 derived samples per each original sample, so that data will be 10 times larger) approach. The target value will not hinder the classification accuracy.

$$x(t) = y(t) + n(t) \tag{1}$$

$$p(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2}{2\sigma^2}} \tag{2}$$

$$SNR_{dB} = 20 \cdot log_{10} \frac{\sigma_{signal}}{\sigma_{noise}} \tag{3}$$

Gaussian data augmentation (GDA) has two parameters: SNR and the Multiplier. SNR serves as a noise tolerance limit. It tells us how much additive white Gaussian noise (AWGN) can be added to the original image without compromising classification accuracy. Preliminary experiments were used to determine the appropriate value of SNR. In Table I, the "Noise" column denotes the percentage of noise added to the original signal. For example, row 5, SNR=20 and Noise=10% denotes that the noise loudness is 10% of the signal's loudness.

TABLE I. SIGNAL-TO-NOISY ANALYSIS

SNR (dB)	Noise (%)	Training (%)	Validation (%)	Test (%)
0	100%	99.03	96.73	86.96
5	56.2%	99.27	97.09	86.96
10	31.6%	99.52	98.91	91.30
15	17.8%	99.64	100.00	95.65
20	10.0%	99.76	100.00	100.00
25	5.6%	99.88	100.00	100.00
30	3.2%	100.00	100.00	100.00
40	1.0%	100.00	100.00	100.00
60	0.1%	100.00	100.00	100.00
100	0.0%	100.00	100.00	100.00

The second parameter, the multiplier values indicates how many additional samples will be generated by GDA. Based on [1], we set our Multiplier value equal to 100. With this value will provide sufficient augmented samples to achieve expert-level classification accuracy.

C. *Deep Learning Models*

Deep learning is currently the best performing family of techniques for uncovering the defining patterns needed for classification from oceans of big datasets. A deep neural network (DNN) is one technique under such umbrella. DNN is en essences an artificial neural network (ANN) with not just one or two layers, but instead with multiple hidden layers. By using these "deeper" models, more complex patterns and non-linear relationships can be modeled. Three DNN models are studied.

MLP is a basic deep learning model which consists of fully-connected feedforward layers with a *ReLU* activation function. Fig. 6 shows our implementation of MLP. The input data is shown as a purple cube. The dense layers are shown in gray and the dropout layer in orange. The output consists of a *softmax* activation function for multiclass classification.

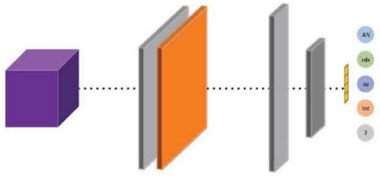

Fig. 6. One-layer MLP for Multiclass Classification

CNN is a convolutional neural network which consists of at least one convolution kernel with a *ReLU* activation function. Fig. 7 shows our implementation of CNN. The 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. We flatten it and use two fully-connected layers before reaching the final *softmax* activation function for multiclass classification.

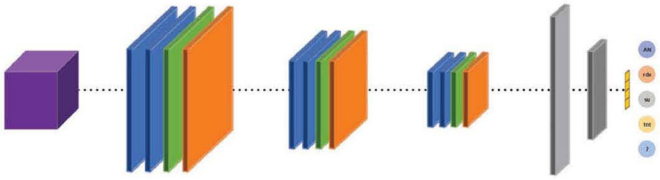

Fig. 7. Three-layer CNN for Multiclass Classification

LSTM is a recurrent neural network which consists of at least one long-short term memory layer with an activation function that uses *tanh*. Fig. 8 shows our implementation of CNN-LSTM, for which the input transformations and recurrent transformations are both convolutional. The core LSTM layers are shown in yellow. For consistency, the 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. We still flatten it and use two fully-connected layers before reaching the final *softmax* activation function.

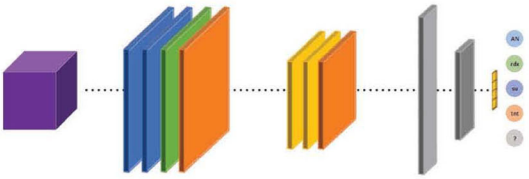

Fig. 8. Two-layer CNN+LSTM for Multiclass Classification

D. *Performance Metrics*

To evaluate the impact of our pre-processing approach on the classification accuracy, the confusion matrix was computed for each classifier without and with PCA, and without and with PCA+GDA. Table II presents the basic structure of the confusion matrix and depicts how true positives, true negatives, false positives and false negatives are defined.

TABLE II. CONFUSION MATRIX

	Predicted Class	
True Class	***Detected***	***Not Detected***
Detected	True Positive (TP)	False Negative (FN)
Not Detected	False Positive (FP)	True Negative (TN)

Another tabular performance metric is the classification report. Table III presents the basic structure of this report, which includes the *Precision*, *Recall*, *F1-score* and *Support* for each class, as well as their average values. To compile these metrics the below equation are used:

- *Precision* is the ratio between number of true positives over the sum of true positives and false positives.
- *Recall* is the ratio between number of true positives over the sum of true positives and false negatives.
- *F1-score* is number of true positives over the sum of true positives, ½false positives and ½false negatives.
- *Support* refers to the number of samples in a Class.

TABLE III. CLASSIFCATION REPORT

Classes	**Precision**	**Recall**	**F1-score**	**Support**
Class$_i$	$\frac{TP}{(TP+FP)}$	$\frac{TP}{(TP+FN)}$	$\frac{TP}{(TP+\frac{1}{2}FP+\frac{1}{2}FN)}$	$TP+FP+FN+TN$

Finally, *classification accuracy* is calculated as the ratio between the number of correctly classified samples, which includes both true positives and true negatives, over the *support*. Moreover, to reduce the effect of overfitting, twenty-fold cross-validation is used to obtain an overall accuracy, as the average for all runs and classes. Fig. 9 present a graphical data partition for five-fold cross-validation, where the test set

consists of ⅕ of the data (blue squares), the validation set also consists of ⅕ of the data (orange squares) and the training set consists of the remaining ⅗ of the data (white squares). And, in every run, different partitions are used for training, validation and testing.

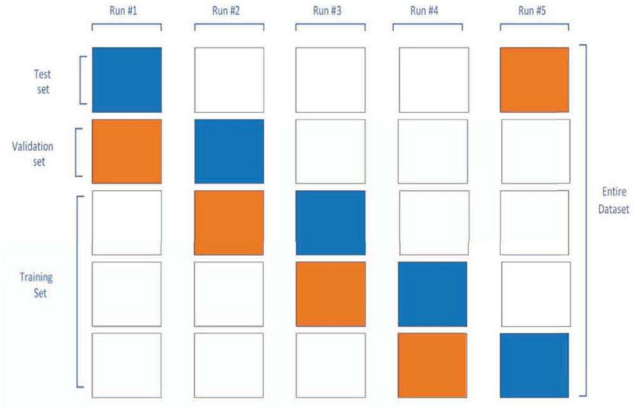

Fig. 9. K-fold Cross Validation Partitions

IV. Experimental Results

This section presents our experimental results in a graphical and tabular way. We start with MLP, CNN and CNN+LSTM without any pre-processing. Then, we present the results of adding PCA to the models. And, we add our GDA approach to the models as a second layer of pre-processing. The code was written in Python. Keras and TensorFlow [16] packages were also employed. A Jupyter notebook environment was used to run all experiments on a cloud-based virtual machine sporting a Tesla k80 GPU, which has a total of 2496 CUDA cores and main memory size of 12GB GDDR5.

A. *Multilayer Perceptor (MLP)*

This subsection presents a baseline for a Multilayer Perceptor (MLP) classifier. Fig. 10 shows two graphs for MLP, one depicts classification accuracy percentage on the right column and one shows classification loss on the left column. Both graphs are functions of the number of epochs. The blue line represents the training set while the orange line represents the validation set. One dropout layer was used to deal with overfitting as depict in section III.C.1. Note that MLP seems to suffer from overfitting. Yet we are interested only in the impact of PCA+GDA.

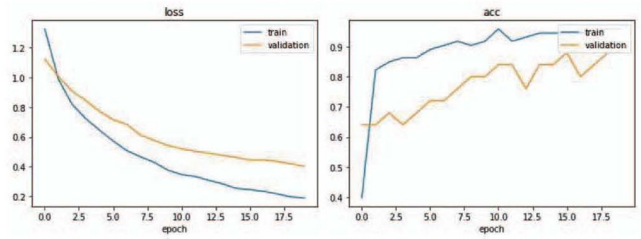

Fig. 10. MLP for Multiclass Classification

B. *Convolutional Neural Network (CNN)*

This subsection presents a baseline for Convolutional Neural Network (CNN). Fig. 11 shows two graphs for CNN, one depicts classification accuracy percentage on the right column and one shows classification loss on the left column. Both graphs are functions of the number of epochs. The blue line represents the training set while the orange line represents the validation set. Three dropout layers were used to deal with overfitting as depict in section III.C.2.

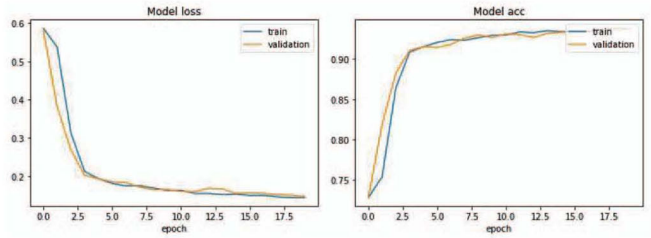

Fig. 11. Principal Component Analysis

C. *Long-Short Term Memory (LSTM)*

This subsection presents a baseline Long-Short Term Memory (LSTM). Fig. 12 shows two graphs for LSTM, one depicts classification accuracy percentage on the right column and one shows classification loss on the left column. Both graphs are functions of the number of epochs. The blue line represents the training set while the orange line represents the validation set. Two dropout layers were used to deal with overfitting.

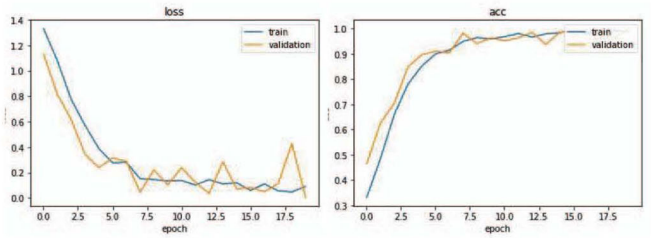

Fig. 12. Loss and accuracy for LSTM

D. *Deep Learning Models without Preprocessing*

To establish the classification accuracy without preprocessing, CNN was initially selected as the better classifier. Table IV presents the experimental results on the test set without incorporating any pre-processing. Precision and Recall is given in percentages. F1-score and Support was also calculated for CNN. All values were collected from our twenty-fold cross-validation experiments. The classification accuracy for each of our models (MLP, CNN and CNN+LSTM) without preprocessing is 78%, 81% and 44%; correspondingly.

TABLE IV. CLASSIFICATION REPORT WITHOUT PREPROCESSING

No Preprocessing				
Classes	***Precision***	***Recall***	***F1-score***	***Support***
AN	100%	57%	0.73	7
RDX	44%	100%	0.62	4
Sucrose	100%	83%	0.91	6
TNT	100%	88%	0.93	8

E. Deep Learning Models with PCA

To evaluate the classification accuracy with PCA preprocessing, CNN was initially explored. Table V presents the experimental results on the test set incorporating PCA. Precision and Recall is given in percentages. F1-score and Support was also calculated for PCA+CNN. All values were collected from our twenty-fold cross-validation experiments. Moreover, the classification accuracy for each of our models (MLP, CNN and CNN+LSTM) with PCA is 78%, 84% and 60%; correspondingly.

TABLE V. CLASSIFICATION REPORT WITH PCA

Principal Components				
Classes	***Precision***	***Recall***	***F1-score***	***Support***
AN	100%	67%	0.80	6
RDX	78%	88%	0.82	8
Sucrose	100%	100%	1.00	8
TNT	50%	67%	0.57	3

F. Deep Learning Models with PCA and GDA

To establish the classification accuracy with PCA and GDA preprocessing, CNN was initially evaluated. Table VI presents the experimental results on the test set incorporating both PCA and GDA. Precision and Recall is given in percentages. F1-score and Support was also calculated for PCA+GDA+CNN. All values were collected from our twenty-fold cross-validation experiments. The classification accuracy for each of our models (MLP, CNN and CNN+LSTM) with both PCA and our GDA is 94%, 99% and 96%; correspondingly.

TABLE VI. CLASSIFICATION REPORT WITH PCA+GDA

Augmented Components				
Classes	***Precision***	***Recall***	***F1-score***	***Support***
AN	100%	100%	1.00	5
RDX	100%	100%	1.00	6
Sucrose	100%	100%	1.00	6
TNT	100%	100%	1.00	8

Table VII compiles a tabular comparison of the classification accuracy for all deep learning models and different preprocessing approaches Each value in the table is the mean plus-minus standard deviation of the cross-validation results. Our experimental results suggest that the addition of PCA helps remove anomalies and outliers. Moreover, our experimental results also suggest that the use of PCA coupled with our GDA approach helps improve robustness to new never-seen-before data.

TABLE VII. CLASSIFICATION ACCURACY COMPARISON

Technique	**MLP**	**CNN**	**LSTM**
No preprocess	78.0 ± 9.0	80.7 ± 9.1	44.0 ± 9.2
PCA	77.6 ± 6.9	84.0 ± 6.5	60.0 ± 6.7
PCA + GDA	93.6 ± 4.7	99.0 ± 2.9	96.0 ± 3.8

The ideal confusion matrix will only have nonzero values on the main diagonal. Values greater than zero on the main diagonal are true positives and true negatives. Any nonzero value on the upper right corner is a false negative while any nonzero value on the lower left corner is a false positive. Fig. 13 shows the Confusion Matrix for MLP (left hand side chart), CNN (middle chart) and CNN+LSTM (right hand side chart). It is clear that for all three models, the classification accuracy showed an improvement with the integration of our preprocessing PCA+GDA approach.

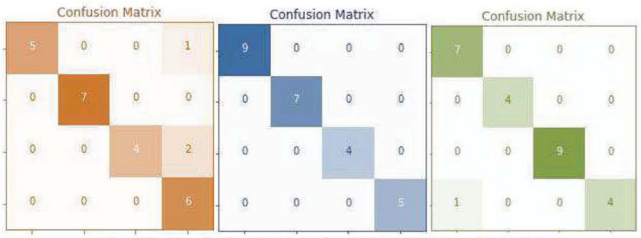

Fig. 13. Confusion Matrix for MLP, CNN and LSTM

V. CONCLUSION

Principal Component Analysis and our Gaussian Data Augmentation in deep learning models were explored on a highly dimensional and limited labeled dataset. Both PCA and GPA methods were used to preprocess multiclass classification input. Initially, three deep learning models were built. A baseline was established for each model. Then PCA was integrated to the models as a preprocessing step to reduce the number of features. Then GDA was also integrated to the models as a second preprocessing step to increase the number of labeled samples.

The experimental results, presented in this paper, suggest that our PCA+GDA approach has a positive impact in deep learning classification, particularly in the presence of highly dimensional, limited labeled data. Future works will focus on further exploring different types and sizes of datasets, including IBIS, MRIs, and Big Data sources.

ACKNOWLEDGMENT

The authors want to acknowledge the Naval Research Laboratory for providing the data sets.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT Press, 2016.

[2] J. Chmidhuber, "Deep Learning in Neural Networks: An Overview," Neural Networks, vol. 61, pp. 85–117, 2015.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[4] A. Esteva, B. Kuprel, and R. A. Novoa et al, "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, pp. 115–118, 2017.

[5] V. Gulshan, L. Peng, and M. Coram et al, "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs," Jama, vol. 304, pp. 649–656, 2016.

[6] M. A. Carreria-Perpinan, "A Review of Dimension Reduction Techniques," Technical Report CS-96-09, Dept. of Computer Science, University of Sheffield, Jan. 1997.

[7] I. K. Fodor, "A Survey of Dimension Reduction Tehniques," Technical Report UCRL-ID 148494, LLNL, Jun. 2002.

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

[9] J. F. R. Rochac and N. Zhang, "Feature Extraction in Hyperspectral Imaging Using Adaptive Feature Selection Approach," in The Eighth International Conference on Advanced Computational Intelligence (ICACI2016), Chiang Mai, Thailand, 2016, pp. 36–40.

[10] N. Zhang and L. Thompson, "An Intelligent Clustering Algorithm for High Dimensional and Highly Overlapped Photo-Thermal Infrared Imaging Data," in ASEE Mid-Atlantic Section Conference, 2016

[11] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," 2017.

[12] E. J. Bjerrum, M. Glahder, and T. Skov, "Data Augmentation of Spectral Data for Convolutional Neural Network (CNN) Based Deep Chemometrics," Cornell University, arXiv preprint arXiv:1710.01927 [cs.LG], 2017.

[13] E. J. Bjerrum, "SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules," arXiv preprint arXiv:1703.07076 [cs.LG], Mar. 2017.

[14] R. Furstenberg, C. A. Kendziora, J. Stepnowski, S. V. Stepnowski, M. Rake, M. R. Papantonakis, V. Nguyen, G. K. Hubler, and R. A. McGill, "Stand-Off Detection of Trace Explosives by Infrared Photo-Thermal Spectroscopy," Applied Physics Letters, Dec. 2008.

[15] J. F. R. Rochac, L. Thompson, N. Zhang, and T. Oladunni, "A Data Augmentation-assisted Deep Learning Model for High Dimensional and Highly Imbalanced Hyperspectral Imaging Data," in The 9th International Conference on Information Science and Technology (ICIST 2019), Hulunbuir, Inner Mongolia, China, Aug. 2019.

[16] G. Zaccone, M. R. Karim, and A. Menshawy, "Deep Learning with TensorFlow," Packt, 2017.