Autonomous Plankton Classification from Reconstructed Holographic Imagery by $L1$-PCA-assisted Convolutional Neural Networks

Kavita Varma,*‡ Lisa Nyman,† Konstantinos Tountas,* George Sklivanitis,* Aditya R. Nayak,† and Dimitris A. Pados*

*I-SENSE and Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University, Boca Raton, FL 33431, USA
E-mail: {kvarma2018, ktountas2017, gsklivanitis, dpados}@fau.edu

†Department of Ocean and Mechanical Engineering and Harbor Branch Oceanographic Institute
Florida Atlantic University, Ft. Pierce, FL 34946, USA
E-mail: {lnyman2012, anayak}@fau.edu

Global Oceans 2020: Singapore - U.S. Gulf Coast | 978-1-7281-5446-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEEECONF38699.2020.9389240

***Abstract*—Studying and monitoring plankton distribution is vital for global climate and environment protection as they are the most elementary part of oceanic eco-systems. However, the conventional methods and techniques used for understanding the planktons are slow and lacks precision and therefore, in modern day scientific and engineering implementations, Convolutional Neural Networks is extensively used in deep learning and machine learning applications as it outperforms traditional manual approach. Dynamic nature of oceans make it very challenging to monitor these microscopic organisms. Our approach here is to generate a powerful automated plankton recognition system to autonomously identify them and improve the D-CNN for classification of the Plankton holographic imagery curated with the method of Data Conformity Evaluation. The performance of D-CNN classifier is improved by various hyper-parameter tuning, regularization techniques and appending meta-data. Conformity evaluation is based on a matric that's calculated on a continuously refined sequence of calculated L_1-norm tensor subspaces of the Plankton images. We note that our classifier performs accurately where our results improve performances from contemporary Deep Learning classifier alone.**

***Index Terms*—Autonomous Classification, D-CNN, Data Conformity, L1-PCA HOLOCAM.**

I. Introduction

In recent years, large public image repositories such as ImageNet [1] and emerging high-performance graphics processing units (GPUs) accelerated the adoption of deep neural networks (DNNs) as means to carry out visual object detection for several applications including gesture recognition, web search, medical and aerial imaging, to name a few. In particular, deep convolutional neural networks (D-CNNs) [2], [3] are used extensively to improve accuracy in large-scale image classification systems. When neural networks train on noisy or mislabeled data, they often (over-)fit to the noise measurements and faulty labels, which leads to significant performance degradation. In this paper, we consider a new method based on L1-norm principal-component analysis (PCA) to improve the quality of labeled data sets that are used for training a custom version of the D-CNN described in [2] known as miniVGG [4]. We test experimentally the classification accuracy of miniVGG using imagery acquired from a novel submersible microscopic holographic imaging system known as the HOLOCAM [5]. Specifically, we consider a three way tensor data set generated from the fixed size (128×128) grayscale subscenes from the HOLOCAM images, which are pre-categorized into corresponding classes of plankton by human domain experts. The conformity of the tensor data per class is evaluated through iterative projections on robust, high-confidence data characterizations per class that are returned by L1-norm tensor subspaces [6]–[9]. Non-conforming tensor slabs are likely to be contaminated by excessive noise or examples mislabeled due to mistakes made during data entry and are automatically removed from the data set. In this work, we conduct experiments with four classes of plankton (see Fig. 1a). Each class is contaminated with images of background scenery as well as images of other plankton species (Fig. 1b). We show that L1-norm tensor-conformity curation of the data identifies and removes inappropriate class instances from the training set and drastically improves/restores the classification accuracy of the miniVGG.

‡Principal corresponding author. The work of K. Varma, K. Tountas, G. Sklivanitis and D. A. Pados was supported in part by NSF under Grants CNS-1704813 and ECCS-1462341. The work of D. A. Pados was also supported by the Schmidt Family Foundation. The work of all authors was also supported by a 2019 Seed Grant from the College of Engineering & Computer Science and I-SENSE, FAU.

II. Problem Formulation

In situ digital imaging technologies are fast emerging as important means of recording spatial and temporal distributions of plankton in the ocean. In particular, digital holography is a promising technique that provides 3-D spatial distributions of particles and plankton within a given sample volume. We have obtained a large training data set consisting of subscenes of holographic imagery captured *in situ* using the HOLOCAM [5], a submersible digital holographic imaging system.

978-1-7281-5446-6/20/$31.00 ©2020 IEEE
Authorized licensed use limited to: Florida Atlantic University. Downloaded on October 18,2021 at 04:06:58 UTC from IEEE Xplore. Restrictions apply.

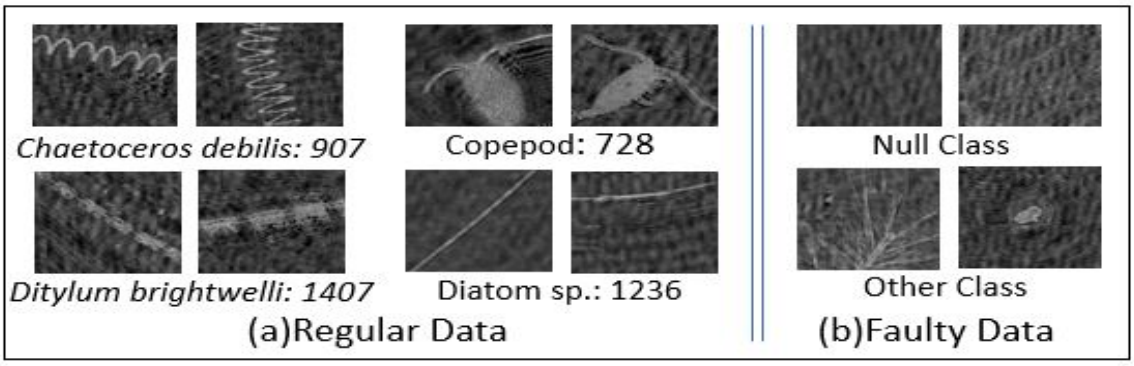


Fig. 1: (a) Examples of images of plankton for the four classes; (b) Examples of mislabeled images.

To acquire images, the HOLOCAM is lowered into the water column a rate of 4-6 cm/s from an anchored ship in order to observe zooplankton and phytoplankton colonies in their natural environment. The holographic data are extensively processed in three major steps: background subtraction, reconstruction, and image plane consolidation. A mean image is computed for each HOLOCAM cast and that mean is subtracted from each hologram within that cast. This effectively eliminates imaging artifacts and nonuniformities associated with variations in background intensity. Next, the hologram is reconstructed at varying depths using the Fresnel diffraction formula [10] at incremental steps of 500 μm, resulting in 70 planes per image, where particles at a particular depth come into focus at the given plane, generating a 3-D spatial distribution of particles within the sample volume. Finally, a composite "flattened" image is generated such that every particle in a given hologram is presented at an optimal depth where it is in focus. These 2D images are 2048 $\times$ 2048 pixels in size at a spatial resolution of 4.59 μm/pixel [11], [12].

Tens of thousands of images are processed and reconstructed. From these, four commonly seen classes of plankton are identified either to class, sub-class or species level as: (1) Copepod; (2) an unknown thin elongated diatom chain, denoted as "Diatom sp."; (3) *Chaetoceros debilis*; and (4) *Ditylum brightwelli*. Both *Chaetoceros debilis* and *Ditylum brightwelli* are diatoms which have a distinct enough morphology to be identifiable to a species level. Upon identification within the large flattened image, the holographic image is cropped to 128 $\times$ 128 pixels ($\sim$ 588 $\times$ 588 μm) centered on the organism of interest. This process produced 907 images of *C. debilis*, 1236 images of Diatom sp., 1407 images of *Ditylum brightwelli*, and 728 images of copepods. Copepods are not as plentiful in the data but have a very distinctive body shape. In order to produce a sufficient number of copepod images, copepods were also identified from an additional dataset with a lower spatial resolution. The images from the lower resolution set (at 4.68 μm/pixel) were interpolated to a 4.59 μm/pixel resolution, which is the resolution of the diatom images.

In a case where the organism is larger than the 128 $\times$ 128 pixel window at 4.59 μm/pixel resolution, centering the window in the middle of the organism may not show a representative sample of the unique shape of that organism. To solve this issue, a window containing an identifying characteristic of the organism is selected. A window selection for a large copepod would contain characteristic features such as long antennae or the caudal ramus (a pair of tail-like spiny protrusions). This is shown in Fig. 2. This allows the neural network to be trained on the characteristic features of each plankton type, even if some samples are too large to fit in their entirety in the 128 $\times$ 128 pixel window.

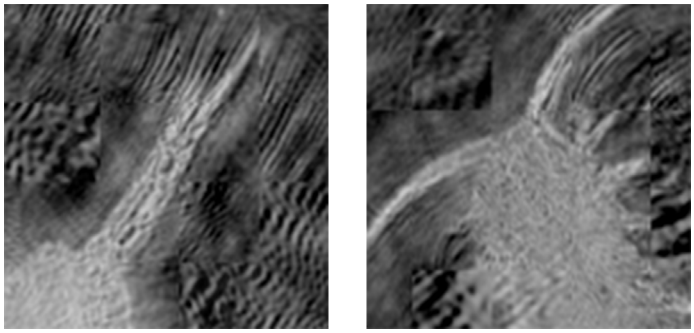


Fig. 2: Training data selection window on copepods that are too large to fit in the 128 $\times$ 128 window. The left image highlights the caudal ramus and the right image focuses on the long antennae.

A total of $N = 4278$ subscenes that contain these plankton from the HOLOCAM images are manually annotated (as shown in Fig. 1a. The examples of images per class i, $i = 1, 2, 3, 4$, are organized in a three way tensor $\mathcal{X}_i \in \mathbb{R}^{128\times128\times N_i}$. We test and demonstrate for the first time in the literature the application of novel tensor data conformity evaluation tools [6]–[9] on $\mathcal{X}_i$, $i = 1, 2, 3, 4$, to improve the training and classification accuracy of CNNs for autonomous plankton classification.

III. System Description

We assess the quality of labeled plankton images for each of the four classes by first calculating robust subspace representations along each tensor dimension by means of L1-norm tensor PCA [6]–[9]. Projecting tensor data on the calculated L1-norm subspaces produces a soft score in the $[0, 1]$ range, which measures the pixel-wise conformity of the entire tensor

Authorized licensed use limited to: Florida Atlantic University. Downloaded on October 18,2021 at 04:06:58 UTC from IEEE Xplore. Restrictions apply.

for each class. The conformity of each image is calculated by averaging the pixel-wise conformity along each 2-D image. Images with high score are highlighted as non-conforming. Removing inappropriate class instances from the training set by a thresholding operation leads to improvements in the quality of data, which drastically increases the classification accuracy of supervised learning models. In this work, we

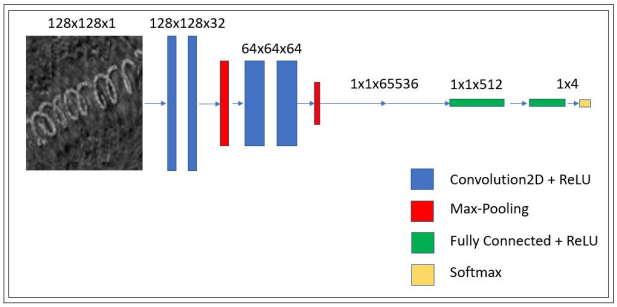


Fig. 3: D-CNN Architecture.

Layers	Output Size
Input	128x128x1
Conv2D (32 filters)	128x128x32
Activation (ReLU)	128x128x32
Batch normalization	128x128x32
Conv2D (32 filters)	128x128x32
Activation (ReLU)	128x128x32
Batch Normalization	128x128x32
Max-Pooling (2x2)	64x64x32
Dropout (0.25)	64x64x32
Conv2D(64 filters)	64x64x64
Activation (ReLU)	64x64x64
Batch Normalization	64x64x64
Conv2D (64 filters)	64x64x64
Activation (ReLU)	64x64x64
Batch Normalization	64x64x64
Max-Pooling (2x2)	32x32x64
Dropout (0.25)	32x32x64
Fully Connected/Dense (512)	512
Activation (ReLU)	512
Batch Normalization	512
Dropout (0.5)	512
Fully Connected (Classes)	Classes
Softmax	Classes

Fig. 4: D-CNN Architecture Layers & Outputs.

evaluate experimentally, the impact of applying tensor conformity evaluation to the HOLOCAM data on the classification accuracy of a customized version of VGG –a typical CNN architecture widely used in computer vision also known as miniVGG [4]. Convolutional Neural Networks (CNNs) are a category of neural networks designed for image recognition and classification. Convolutional Neural Networks apply a network of hierarchical filter to the grid of pixels in a manner inspired by Hubel and Weisel's experiments of visual cortex's response on visual signal [13]. The filters are then convolved against the input image, i.e., performing element-wise multiplication between the filter and the region of the image that it covers for every possible region in the image. MiniVGG has two units of CONV-CONV-POOL blocks followed by a set of FC-FC layers, where CONV, POOL and FC denote convolution, max-pooling, and fully-connected dense layers, respectively. SOFTMAX activation follows FC layers for multi-class plankton classification. The network layers are shown in Fig. 3. The POOL block is also known as Max-Pooling layer which, also called as subsampling or down sampling, reduces the dimensionality of each feature map by picking up the maximum number in a 2×2 matrices of feature map. It makes the system, invariant to transformations, translations and distortions. Activation Functions such as softmax are used to introduce nonlinearity to the system. The numerical calculations complexity can be seen in Fig. 4

IV. Proposed Methodology

We consider 4 three-way tensors $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{128\times128\times N_i}, i = 1,2,3,4$ each containing N_i grayscale subscenes from the HOLOCAM images. Our goal is to use a pre-processing tool called data conformity evaluation [8], [9] to find miss-classified or anomalous images inside the datasets. Data conformity evaluation converts the original tensor data to a new tensor of the exact same dimensions, where each new tensor entry measures the conformity of that entry with respect to all other data points. The conformity metric will take values from the $[0,1]$ set of real numbers, with conformity values close to 1 indicating "misbehaving" data points, and values close to 0 corresponding to nominal data points. This is achieved, by utilizing iteratively refined L_1-norm (absolute-error) data subspaces [6]–[9], [14]. Detection of non-conforming data entries enables the identification of contaminated data slabs. With respect to the cth data tensor, we first unfold the tensor along its columns (rearrange the columns), creating a data matrix $\mathbf{X}_{c,(1)} \in \mathbb{R}^{128\times128N_i}$ and we calculate the r_1 L1-norm principal components $\mathbf{Q}_1^{(0)} \in \mathbb{R}^{128\times r_1}$ by solving the following maximization problem [15], [16]

$$\mathbf{Q}_1^{(0)} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{128\times r_1},\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_{r_1}}}{\operatorname{argmax}} \left\|\mathbf{X}_{c,(1)}^T\mathbf{Q}\right\|_1. \tag{1}$$

The resulting basis describes accurately the subspace spanned by the columns of the original tensor $\boldsymbol{\mathcal{X}}$ that contain nominal data. Columns that are contaminated with anomalous data are not spanned by the resulting basis vectors. The conformity of the data columns is computed by projecting each column $\left[\mathbf{X}_{c,(1)}\right]_{:,i_1}, i_1 = 1,2,\ldots,128N_i$ on the calculated subspace $\mathbf{Q}_1^{(0)}$ as

$$d_{1,i_1}^{(1)} = \left\|\mathbf{Q}_1^{(0)}\mathbf{Q}_1^{(0)^T}\left[\mathbf{X}_{c,(1)}\right]_{:,i_1}\right\|_2^{-1} \quad \forall i_1 = 1,2,\ldots,128N_c. \tag{2}$$

Large $d_{1,i_1}^{(1)}$ values are expected if $\left[\mathbf{X}_{c,(1)}\right]_{:,i_1}$ is an anomalous data vector and small $d_{1,i_1}^{(1)}$ values if $\left[\mathbf{X}_{c,(1)}\right]_{:,i_1}$ is a nominal data vector. After the calculation of the projection of each

Authorized licensed use limited to: Florida Atlantic University. Downloaded on October 18,2021 at 04:06:58 UTC from IEEE Xplore. Restrictions apply.

column on the subspace, the conformity values are converted to a tensor $\mathcal{W}_{c,1}^{(1)} \in \mathbb{R}^{128\times128\times N_c}$

$$\mathcal{W}_{c,1}^{(1)} = \text{tensorization}\left(\left[d_{1,1}^{(1)},\ldots,d_{1,128N_c}^{(1)}\right]^T \odot \mathbf{1}_{128\times128N_c}, 1\right) \tag{3}$$

where $\mathbf{1}_{128\times128N_c}$ stands for an all-ones matrix of dimension $128 \times 128N_c$, and the *tensorization*$(\cdot)$ operation converts the matrix unfolding to the original three-mode tensor form (reverting the unfolding process). The tensor $\mathcal{W}_{c,1}^{(1)}$ contains the conformity values corresponding to each column of the original tensor $\mathcal{X}$. We repeat the above process for the rest of the modes of the original data tensor, and calculate the conformity tensors $\mathcal{W}_{c,2}^{(1)}$ and $\mathcal{W}_{c,3}^{(1)}$. We calculate the final conformity tensor $\mathcal{W}_{c}^{(1)}$ by combining the above calculated tensors in an additive fashion as well as normalizing the tensor so that each element is in the $[0,1]$ range

$$\mathcal{W}_{c}^{(1)} = \frac{\sum_{k=1}^{3}\alpha_k\mathcal{W}_{c,k}^{(1)} - \min\left(\sum_{k=1}^{3}\alpha_k\mathcal{W}_{c,k}^{(1)}\right)}{\max\left(\sum_{k=1}^{3}\alpha_k\mathcal{W}_{c,k}^{(1)}\right) - \min\left(\sum_{k=1}^{3}\alpha_k\mathcal{W}_{c,k}^{(1)}\right)} \tag{4}$$

where $\alpha_1,\alpha_2,\alpha_3 \in \mathbb{R}^+, \sum_{k=1}^{3}\alpha_k = 1$ correspond to weights for each mode of the tensor that model the weighting of the corresponding dimension of the original tensor. The final conformity tensor $\mathcal{W}_{c}^{(1)}$ enables element-wise conformity of the original tensor data. The data conformity values can be iteratively refined until numerical convergence of the data conformity tensor $\mathcal{W}_{c}^{(l)}$.

In order to identify slabs that are contaminated by anomalous or miss-classified images, we calculate the mean data conformity value per slab as

$$\bar{w}_{c,n} = \frac{1}{128 * 128}\sum_{d=1}^{128}\sum_{l=1}^{128}[\mathcal{W}_c]_{l,d,n}, n = 1,2,\ldots,N_c. \tag{5}$$

Slabs with high conformity values contain contaminated images, while low-conformity slabs contain nominal images. We choose to remove slabs with conformity value $\bar{w}_{c,n}$ above a pre-defined threshold $t_{c,\text{cutoff}}$ from the received tensor $\mathcal{X}_c$, resulting in a new tensor $\mathcal{X}' \in \mathbb{R}^{128\times128\times N_c'}$, where $N_c' < N_c$. The new tensor $\mathcal{X}'$ contains only images that are not anomalous, which results in better classification performance.

V. Experiments

The D-CNN runs on Ubuntu OS on a graphics processing unit (GPU) NVIDIA RTX-2070. We implement the D-CNN in Python using Tensorflow 2.0/Keras that runs on the CUDA framework. The D-CNN runs for 10 distinct instances of the network Training experiments. For each instance of the D-CNN, we split the data into "training" and"test" data according to the following ratio: $80:20$. The "test" data is never seen by the D-CNN classifier during it's training process. The number of samples used for the experiments is 4278 of size 128 $\times$ 128 gray pixels. The Keras ImageDataGenerator class in python is used to artificially expand the size of the training data set and to increase the generalizability of our classifier. These augmented images are the modified forms of the original images by applying simple geometric transforms such as random translations like shifting horizontally and vertically by a factor of 0.1, random rotations of ±20 degrees, changes in scale, shearing by 0.2, zoomed by uniformly sampling in the range $[0.8, 1.2]$. Applying some of these transformations to an input image changes its appearance slightly, but it does not change the class label – thereby making data augmentation a natural method to apply to deep learning for computer vision tasks. The data augmentation was limited to just reproducing two random transformed copies of each images in order to avoid over-fitting. Additionally, the labels for the images were "one-hot encoded" as it does a better job in prediction by not assigning values to labels.

1) Model Tuning - Hyperparameters: Hyperparameters play a significant role in controlling the classifier to train as accurately as possible. This particular model comprises of two blocks of layers each comprising of a set of two convolution layers along with dropout layers. Fluctuations of certain hyperparameters such as *batch size*, number of *epochs* per iterations and *learning rate schedulers* were deployed to bring out the best tentative hyperparameter combinations for accurate classification. An *epoch* is a measure of the number times the network has seen a training sample in order to update the classifier weights. An epoch can comprise of one or more batches. *Batch size* is a hyperparameter specific to network training which can be defined as the amount of individual training samples that the classifier take in batches to perform in one epoch. Batch size of value 128 is used in the network. *Learning rate scheduler* is an optimization hyperparameter that optimizes the learning rate based on specified criteria. The criteria used in the network is reduction of learning rate by 20% after every 10 epochs.

2) Model Tuning - Regularization: Regularization is a method to help the model learn better in order to improve model performance to predict an unseen data. One such technique that's quite common in practice is "dropout regularization" which can be defined as the percentage of drops in "dropout layer". Dropout specifies the probability at which outputs of the layer are dropped out randomly, or inversely, the probability at which outputs of the layer are randomly retained. There are three dropouts layers in the architecture with different percentages. Data Augmentation is another regularization technique that's been discussed earlier.

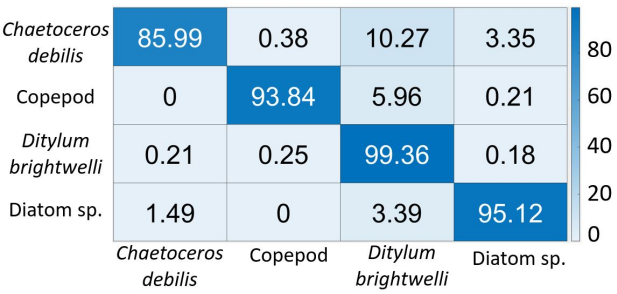
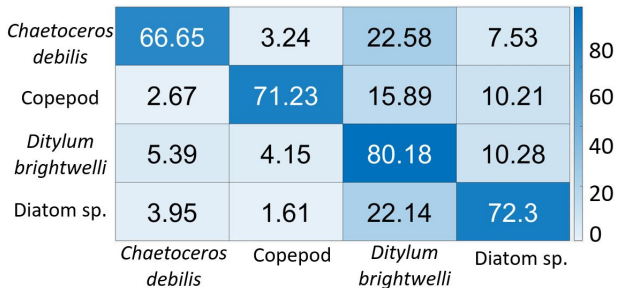
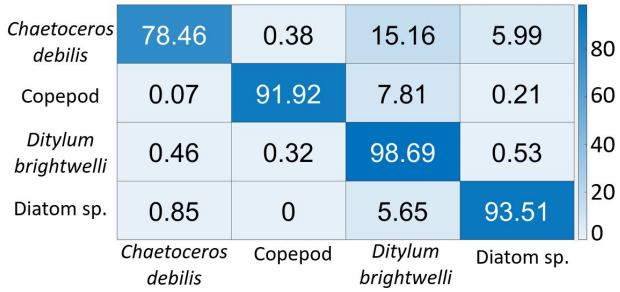


Fig. 5: Classification Error Rate on "pure" plankton data ($N = 4278$).

Authorized licensed use limited to: Florida Atlantic University. Downloaded on October 18,2021 at 04:06:58 UTC from IEEE Xplore. Restrictions apply.



(a) Classification Error Rate on Contamination A.



(b) Improvement seen on Cleaning the Contamination A.

Fig. 6

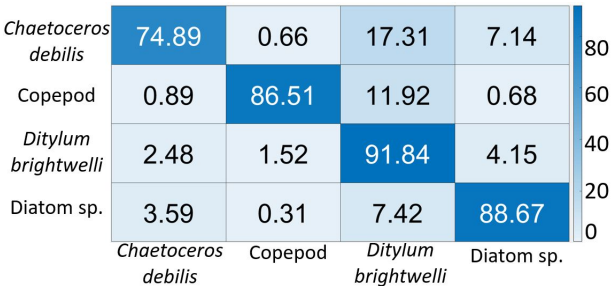
A. Experiment 1: No Contamination

The first experiment is performed on the $N = 3417$ ("trainings" images = 80% of total images) "pure plankton images" comprising of 724 images of *C. debilis*, 967 images of Diatom sp., 1090 images of *Ditylum brightwelli*, and 636 images of copepods. These "training" examples of plankton images per class are further split into (75%) and (25%) for training and validation purposes in the D-CNN network. After tuning the model's hyperparameters and performing regularization techniques, the training loss of $2{\cdot}10^{-3}$ and validation accuracy of 93% is achieved over 100 to 125 epochs. The loss function that is generally used for optimizing classification models, one such as above, is the "categorical cross entropy loss". Cross-entropy is a measure from field of information theory for finding the difference between two probability distributions for a given random variable or set of events. The interpretation of how well the model is doing is based on the loss that is calculated and summed on training and validation. Fig. 5 shows the performance of the D-CNN classifier on the four classes of plankton (a.k.a pure plankton data) on the "test" data averaged over 10 distinct "training-testing" experiments. The elements of the confusion matrix are the summed result of the individual elements of the individual confusion matrices across all distinct training-testing instances and divided by the total number of distinct instances.

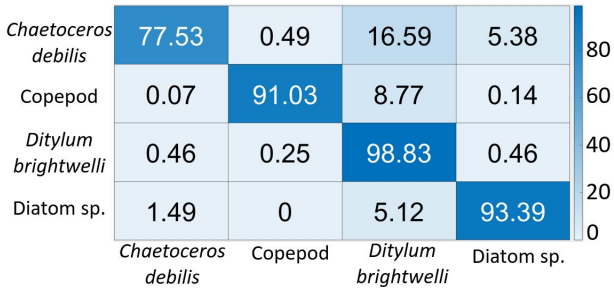
B. Experiment 2: Contamination A

A next step taken in this approach is by considering the data that is randomly contaminated by the images from the remaining three classes individually. The 25% of the images from each class is replaced by the images from the other three classes (among the four considered classes of "pure plankton"). Experiment is performed on the $N = 3417$ images, also known as "25% contaminated plankton images: Contamination A". The dataset comprises of 724 images in *C. debilis* class, 741 images in Diatom sp. class, 1090 images in *Ditylum brightwelli* class, and 636 images in copepods class with each class being 25% contaminated. These "training" examples of plankton images per class are further split into (75%) and (25%) for training and validation purposes in the D-CNN by the network.

After tuning the model's hyperparameters and performing regularization techniques for training, the classification error rate is calculated for each distinct instance. Fig. 6a shows the class-confusion matrix of the D-CNN on this contaminated data. These contaminated "training" data ($N = 3417$ images) are cleaned by the proposed methodology of L1-PCA faulty-data removal using the data conformity method which spits out the anomalous images which are not conforming with the "pure" images from that particular class. The classifier re-runs the whole process on this cleaned data. This cleaning process results in 547 images in *C. debilis* class, 967 images in Diatom sp. class, 861 images in *Ditylum brightwelli* class, and 441 images in copepods class. Fig. 6b shows the classification error rate of the "cleaned" data averaged over 10 distinct instances of "training-testing" experiments. The "true" label and the "predicted" label of the confusion matrices are represented by vertical and horizontal axes of the figures, respectively.



(a) Classification Error Rate on Contamination B.



(b) Improvement seen on Cleaning the Contamination B

Fig. 7

C. Experiment 3: Contamination B

This experiment is a another method of contamination of the "pure images" wherein we purposefully contaminate it by using the images from "null class" and "other classes" (Fig. 1 (b)). Each class under interest consists of 25% contamination.

Authorized licensed use limited to: Florida Atlantic University. Downloaded on October 18,2021 at 04:06:58 UTC from IEEE Xplore. Restrictions apply.

Experiment is performed on a total of $N = 3417$ images, also known as "25% contaminated plankton images: Contamination B". The dataset comprises of 724 images in *C. debilis* class, 741 images in Diatom sp. class, 1090 images in *Ditylum brightwelli* class, and 581 images in copepods class with each class being 25% contaminated. These "training" examples of plankton images per class are further split into (75%) and (25%) for training and validation purposes in the D-CNN by the network. After tuning the model's hyperparameters and performing regularization techniques for training, the classification error rate is calculated for each distinct instance. These contaminated images are sent through the proposed L1-PCA Faulty removal algorithm, which returns the clean data as a result. D-CNN is re-run on this clean data for 10 distinct instances of "training-testing". Fig. 7a and 7b depict the class-confusion matrix performance averaged over 10 iterations for the "25% contaminated plankton images: contamination B", and the data set "cleaned" after L1-PCA faulty-data removal, respectively.

The "true" label and the "predicted" label of the confusion matrices are represented by vertical and horizontal axes of the figures, respectively. The improvement in classification error rate is clearly seen comparing Fig. 7b against Fig. 7a, and both against the baseline of ground-truth training in Fig. 5.

CONCLUSION

We evaluate training data set curation method is a novel method of curating the data which were possibly contaminated (by human/machine error) by using our Data Conformity method. Apart from this, the results show that few convolution layers, when their hyper-parameters tuned up appropriately, yield accurate classifications. The accuracy-improving techniques applied to this dataset are not only applicable to automated plankton classification but can also be applied to any other type of imagery-based dataset.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Miami, FL, 2009.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. CVRP*, Boston, MA, 2015, pp. 1–9.

[4] A. Rosebrock, *Deep Learning for Computer Vision with Python*, 3rd ed. PyImageSearch.com, 2019.

[5] A. R. Nayak, M. N. McFarland, J. M. Sullivan, and M. S. Twardowski, "Evidence for ubiquitous preferential particle orientation in representative oceanic shear flows," *Limnology and oceanography*, vol. 63, no. 1, pp. 122–143, 2018.

[6] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L1-subspace signal processing," *IEEE Trans. Signal Processing*, vol. 62, pp. 5046–5058, Oct. 2014.

[7] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Trans. Signal Processing*, vol. 65, pp. 4252–4264, Aug. 2017.

[8] K. Tountas, D. A. Pados, and M. J. Medley, "Conformity evaluation and L_1-norm principal-component analysis of tensor data," in *Proc. SPIE Big Data: Learning, Analytics, and Applications Conf., SPIE Defense and Commercial Sensing*, vol. 10989, Baltimore, MD, Apr. 2019, pp. 1–11.

[9] K. Tountas, G. Sklivanitis, D. A. Pados, and M. J. Medley, "Tensor data conformity evaluation for interference-resistant localization," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2019, pp. 1582–1586.

[10] J. Katz and J. Sheng, "Applications of holography in fluid mechanics and particle dynamics," *Annual Review of Fluid Mechanics*, vol. 42, pp. 531–555, 2010.

[11] A. R. Nayak, M. N. McFarland, M. S. Twardowski, and J. M. Sullivan, "On plankton distributions and biophysical interactions in diverse coastal and limnological environments," in *Ocean Sensing and Monitoring X*, vol. 10631. International Society for Optics and Photonics, 2018, p. 106310P.

[12] A. R. Nayak, M. N. McFarland, J. M. Sullivan, and M. S. Twardowski, "Evidence for ubiquitous preferential particle orientation in representative oceanic shear flows," *Limnology and oceanography*, vol. 63, no. 1, pp. 122–143, 2018.

[13] D. H. Hubel, "Single unit activity in striate cortex of unrestrained cats," *The Journal of physiology*, vol. 147, no. 2, p. 226, 1959.

[14] N. Tsagkarakis, P. P. Markopoulos, G. Sklivanitis, and D. A. Pados, "L1-norm principal-component analysis of complex data," *IEEE Transactions on Signal Processing*, vol. 66, no. 12, pp. 3256–3267, 2018.

[15] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L_1-subspace signal processing," *IEEE Transactions on Signal Processing*, vol. 62, pp. 5046–5058, Oct. 2014.

[16] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L_1-norm principal-component analysis via bit flipping," *IEEE Transactions on Signal Processing*, vol. 65, pp. 4252–4264, Aug. 2017.

Authorized licensed use limited to: Florida Atlantic University. Downloaded on October 18,2021 at 04:06:58 UTC from IEEE Xplore. Restrictions apply.