GPU-Based Normalized Compression Distance for Satellite Images

Mihai STOICA and Mihai COCA

***Abstract*—The measurement of the normalized compression distance (NCD) between two objects is computed by counting the number of common patterns that appear in the representation of compressed objects. Classical Lempel-Ziv (LZ) algorithms do not achieve high compression speeds on images because they require sequentially traversing the linearized version of object to build the corresponding dictionary, i.e., dynamic approach to optimize the encoding based upon the particular input. This paper proposes a faster approach for computing the NCD metric in a parallel manner by estimating compressions on GPU and multi-thread schemes. The goal of calculating distances is to use them in clustering and classification of input images. The tests are run on the UC Merced Land Use and BigEarthNet datasets, which are both well-known and frequently used in the field of remote sensing.**

***Index Terms*—NCD, Image Clustering, Image Classification, Image mining, nvCOMP, BigEarthNet, UC Merged, JPEG.**

I. INTRODUCTION

Nowadays, a massive amount of data is gathered on a regular basis to make future judgments in the process stages. Intelligence of data mining methods made possible the extraction of useful information from data for analytic uses. The items under analysis in data mining related papers are images, which is why the term image mining is intensely used. At first glance, it would seem to be just an extension of data mining to image domain, but it is an area that makes use of knowledge in image retrieval, computer vision and image processing [1].

Image mining is a field that draws on the expertise of a wide range of areas, including image processing, machine learning, and computer vision. Clustering or classification is one of the steps of an image mining system, which aims to group dataset samples according to certain criteria. Classification uses predefined labels for training classifiers that predict, based on the information gathered in the training phase, the category to which the test objects belong to. Clustering involves grouping according to the degree of similarity, without using predefined labels. Both approaches – supervised, respectively unsupervised, are important in image mining because they leverage characteristics taken from the image to extract information about the content.

One of the most used methods of grouping data is to assess the degree of similarity between two objects. NCD is a generic compression-based similarity measure with the primary benefit of a powerful parameter-free method. The NCD [2] metric is preferred due to its robustness [3], which comes from the ability to work with a variety of data formats. CompLearn [4], a suite of compression techniques applied to the process of discovering and learning patterns, is robust to resolution changes in terms of images, which is essential since it allows for the use of lower quality in compressed formats.

Few papers proposed methods that compute NCD on satellite images. In [5], a brief SAR image change detection method based on NCD is projected. This method was further improved and applied for the detection of changes in TerraSAR-X images [6]. The distance matrix generated by applying NCD is used as input to supervised and unsupervised methods in order to obtain a change map in flooding scenarios. The procedure mainly consists in dividing two-time series images into patches, computing a collection of similarities corresponding to each pair of patches, and generating the change map with a histogram-based threshold. In [7], in the context of satellite image retrieval, the authors made a comparison between assessing image similarity through NCD metric and Bag-of-Words (BoW) feature extraction method. They concluded that NCD is computationally very expensive and infeasible to be applied in real applications. Starting from this issue, our article examines NCD outcomes in image clustering and classification with the classic compressors and proposes a faster method that reduces computing time by utilizing a GPU-based compressor. The suggested method is endorsed by technical advancements and by the improved quality and representation of images.

This paper is organized as follows. Section II introduces NCD theoretical background and information about nvCOMP [8], a compressor that uses GPU processing. There are also presented other works in this area, relevant to our proposed method. Section III gives an overview of our proposed approach, NCD-NVCOMP. Section IV presents the results obtained by applying the NCD metric using three compressors, zlib, JPEG and nvCOMP, on two known datasets, UC Merced Land Use [9] and BigEarthNet [10]. The conclusions are drawn in Section V.

II. THEORETICAL ANALYSIS

NCD starts from the definition of Normalized Information Distance (NID), a notion introduced in Information Theory that estimates information shared by two objects. Its definition is expressed in terms of the complexity of Kolmogorov $K(\cdot)$, whose value represents the length of the shortest program *q* used by a universal Turing machine to calculate that object as output (*x*).

M. STOICA is with Cyber Command, Bucharest, Romania (e-mail: mihaistoica1998@gmail.com).
M. COCA is with the Computer Science and Cyber Security Department, "Ferdinand I" Military Technical Academy, Bucharest, Romania (e-mail: mihai.coca@mta.ro).

The Kolmogorov complexity plays an important role in the field of compression, being the absolute limit of the best compression, which is why the definition of the NCD metric is directly related to the Kolmogorov complexity. The Kolmogorov complexity approximation [2] is referred to as

$$\mathrm{NID}(x,y)=\frac{K(x,y)-\min\left(K(x),K(y)\right)}{\max\left(K(x),K(y)\right)} \quad (1)$$

that is the size of the latest compression version of *x*, the best value a real compressor can get.

Thus $C(x)=K(x)+k$, where *C* represents the function of the compressor, and *k* is a constant. NCD [2] is defined as

$$\mathrm{NCD}(x,y)=\frac{C(x,y)-\min\left(C(x),C(y)\right)}{\max\left(C(x),C(y)\right)}, \quad (2)$$

where $C(x,y)$ represents the size of the file obtained by compressing the concatenation of x with y. NCD is explicitly computed between two sequences of bytes x and y to represent how different they are and to facilitate the use in various areas of information theory, e.g., pattern searching and data mining. In practice, the result of function NCD is a non-negative number representing how different the two files are. Smaller numbers represent more similar files. The NCD's value in the upper bound is 1+Ɛ, due to imperfections in compression techniques, but for most standard compression algorithms one is unlikely to see an Ɛ above 0.1 [2]. The NCD metric is used to generate a distance matrix, which is subsequently used to cluster and classify images.

Compression algorithms are of two core types: lossy and lossless. Lossless compression decreases the size of a file by searching for redundant data and does not affect the quality. Conversely, lossy compression involves sacrificing some quality to obtain better rate of size reduction. The most popular criteria to choose one of them are bitrate and compression rate. For compression rate should also be taken into consideration the PSNR (Peak Signal-to-Noise Ratio), to keep a version with acceptable quality. In the case of compression rate, it is clear that lossy compression is better but, in terms of bitrate it is harder to make a comparison. In [11] there were presented some image compressors that support both lossless and lossy compression. The results showed that a lossless compressor, named FLIF (Free Loseless Image Format), was the fastest. FLIF performances demonstrate that lossless and lossy compression are comparable in terms of bitrate. We made experiments with both compression algorithms, lossy, i.e., JPEG, respectively lossless, i.e., zlib and LZ4, in section IV.

CompLearn is a set of easy-to-use programs for applying compression methods to the process of pattern discovery and learning. The effectiveness of compression-based technique occurs from the fact that it can extract patterns in a variety of domains without the requirement for domain-specific parameters. CompLearn framework calculates similarities using NCD and includes common compressors, e.g., bzlib, zlib, and blocksort. The results obtained are dependent on the type of compressor used. Considering a file collection as data input, the result is a quadratic distance matrix with the size of collection cardinality. On the main diagonal there are placed values of compressed binary sequences length of each file with itself, and values of compressed concatenation length of any two binary sequences of distinct files are found in the rest of the matrix. Every item in a dataset is treated as a cluster in hierarchical clustering. Based on the distance matrix, each cluster is merged with its nearest neighbour. When all the clusters are merged, the iteration is complete, and the result is a dendogram. The CompLearn toolkit's maketree function, which is used to build phylogenetic trees, employs the cluster merging procedure described above in conjunction with Hill Climbing to find the location of leaves in the tree.

Calculation of a distance matrix for a small set of images takes a long time using current compression approaches based on the NCD metric [12]. To achieve fewer compression operations, Fast Compression Distance (FCD) [12] was used in our framework. FCD is a similarity measure that involves calculating distances based on dictionaries extracted by the LZW algorithm. LZW compression and decompression are hard to parallelize because they use dictionary tables created by reading input data one by one. A CUDA implementation of LZW compression has been presented in [13], achieving a much smaller compression time than the CPU implementation. nvCOMP is a CUDA library that features generic compression interfaces to enable developers to use high-performance GPU compressors and decompressors in their applications. LZ4 method can achieve up to 100 GB/s compression and decompression throughput depending on the dataset, attaining good compression ratios for arbitrary byte streams. One of the deciding factors in using LZ4 algorithm in our method is the minimum required size of the workspace that is allocated on the GPU.

Classification is a task that demands machine learning algorithms usage in learning how to assign a class label to samples from the data domain. Among the classification algorithms, in the context of high dimensional space application, Support Vector Machine (SVM) is a robust candidate. The classifier aims to determine the hyperplane parameters that best separate the data. Support vectors are points close to the plane that influence its position and orientation, intending to keep a high margin for robustness reasons. A cost function is used to maximize the edge between the support vectors and the hyperplane based on a training dataset. SVM was chosen because it allows the use of a customizable kernel, radial basis function with NCD, so the distance matrix can be used as a feature vector. The distance matrix provides, in the case of images, texture and color palette information [14].

III. Faster Approach for NCD Computation

In this work, we aimed at combining NCD metric and GPU compressors performance, while maintaining similar classification accuracy as CompLearn framework. Hence, the most computationally expensive process in (2) is compression. As classical algorithms used by CompLearn achieve a low-speed rate in compression [15], the first intention was to parallelize the way in which compression is performed.

We proposed a framework to classify multispectral satellite imagery using parallel computing through offloading the computation of a compression matrix on GPU or multi-thread CPU, using afterwards the resulted NCD distance matrix as feature input to SVM classifier.

 Digital Object Identifier 10.32754/JMT.2022.1.07

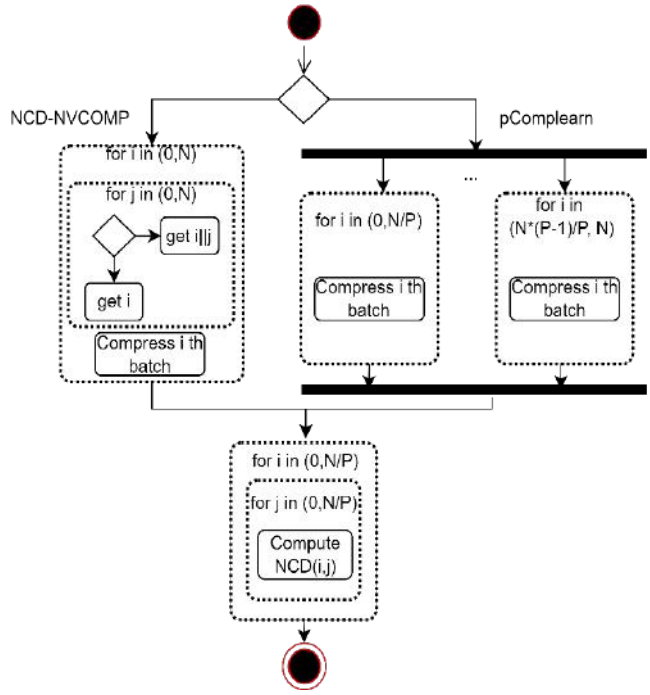

Figure 1. Framework workflow

Fig. 1 illustrates the workflow of the methods integrated into the suggested framework. There are two methods, with differences in compression mode: a) the first method is a thread-local data scheme in which image compressions are run in parallel, hereinafter referred to as pCompLearn, and b) the second approach is a GPU-based compression scheme, hereinafter referred to as NCD-NVCOMP. Each branch generates a compression matrix populated with the byte sequence corresponding to each compressed image. The final output of this workflow is a distance matrix obtained by applying NCD metric on compression matrix content.

In NCD-NVCOMP method, the low-level version of the nvCOMP algorithm allows batch-level compression of images, the number of images being dependent on the internal memory capacity of the GPU. For a big dataset of high-quality images, the set could be split into fixed-size batches, to avoid the overflow of GPU memory.

In pCompLearn method, batch compression computation is run on N files using P threads. When launching P threads for N-quadratic distance matrix calculation, NCD function will be called N/P times. The compression algorithm is chosen from those available in CompLearn based on the criterion of minimum computation time.

IV. Experimental Evaluation

A subset of 100 images was used to choose the optimal number of threads for the pCompLearn method. The results are highlighted in Fig. 2. The time needed for a thread is computationally comparable to the classical one, i.e., running the built-in NCD function from CompLearn framework. The number of threads is dependent on the machine used for testing, and, for our computer, the best solution was to use 8 threads, as is described in Fig. 2. For all the following tests, pCompLearn was distributed on 8 threads.

To choose the best compressor available in CompLearn suite, we considered a subset of 100 images from the UC

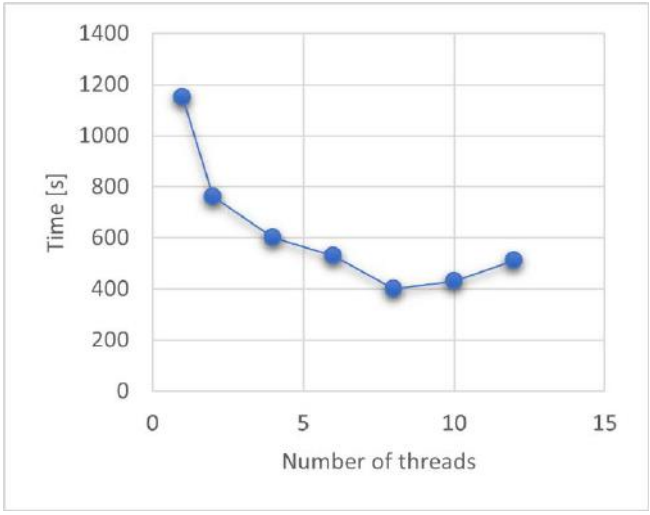

Figure 2. Variation of pCompLearn runtimes with changing the number of threads

Merced Land Use dataset, for which we generated distance matrices using the zlib and bzlib algorithms. Bzlib took 3 minutes and 47 seconds and zlib took 2 minutes and 32 seconds. With better computation time, we implied zlib compressor in our experiments, as the default compressor of the CompLearn suite.

We tested only lossless compressors because CompLearn is built as a general approach, capable of working with any type of file. Based on the results obtained by modern lossless compressors (an example is described in [11]), a comparison between our lossless compressors and a lossy compressor was made. The chosen lossy compressor was JPEG because of its popularity and common usage. For testing JPEG compression, we used Pillow [16]. We tested JPEG performances using the same set of 100 images from UC Merced Land Use dataset and obtained the distance matrix in 1 minute and 13 seconds. This means two times faster than zlib.

To test our methods, we used two well-known remote sensing datasets, UC Merced Land Use and BigEarthNet. All experiments were run on a machine with an i7-8750H processor, with 8 GB RAM and Nvidia GeForce 1050 GPU. It is worth mentioning that the images were not preprocessed, and the linearization of the image to produce the byte sequence was accomplished simply by reading the bytes from left to right, passing over each line.

UC Merged is a land-use remote sensing image dataset with 21 classes and 100 images per class. The images were extracted from the USGS National Map Urban Area Imagery collection and represent various urban and rural areas.

In our first experiment on UC Merced dataset, the computation time of the distance matrix is tracked. To test the computational speed, the entire dataset, i.e., 2100 images divided into 21 classes, was used. The size of an image is approximately 200 KB. The time obtained by NCD-NVCOMP was 8828 seconds, while pCompLearn obtained 84967 seconds, about 10 times longer.

The next experiment on UC Merged dataset tracked the accuracy obtained from clustering images based on distances. The clustering operation was performed using the maketree module, available in the CompLearn utility suite, which generates the phylogenetic tree based on pre-computed distances between objects.

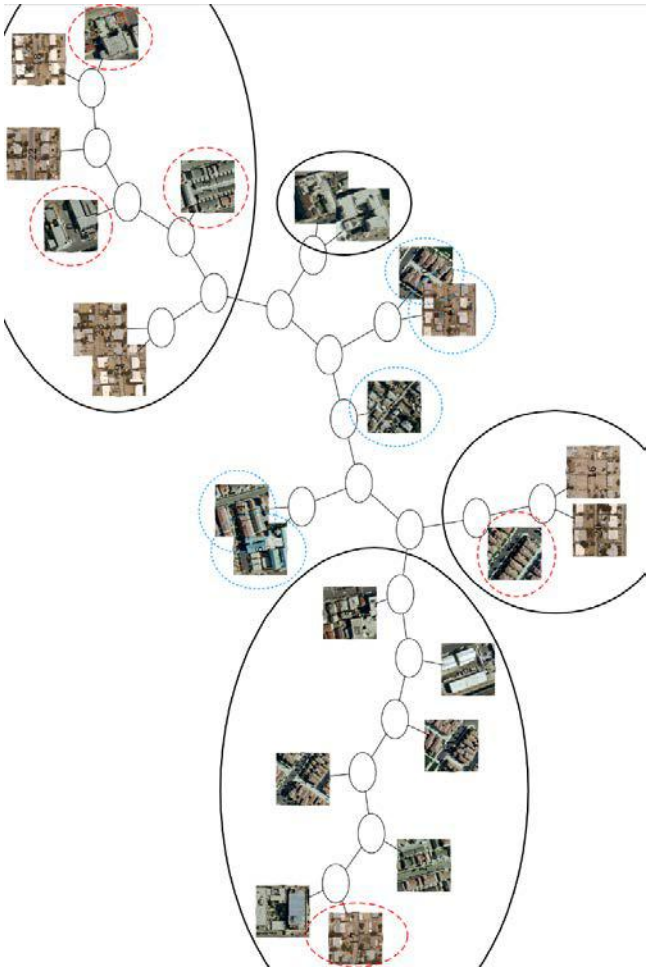

Figure 3. Unrooted tree computed using pCompLearn for UC Merced Land Use dataset. Black solid line represents groups; blue dotted line represents confusion; red dashed line represents false alarm

For this experiment a subset of 24 images was used, equally divided into 3 classes (buildings, dense residence and medium residence). This subset of 24 samples contains urban areas with low inter-class spectral variance and was used to test the clustering accuracy of the NCD-NVCOMP and pCompLearn methods.

The phylogenetic tree was analyzed visually in Fig. 3, looking at how objects in the same category are grouped. A false alarm is considered for each image subset lying in a branch related to another class. A confusion in separation means that the image is not well separated. A false alarm is represented by a red circle and confusion is marked by a blue circle. The only areas of the tree where images were correctly associated are black circled, the rest represented false alarms or confusion (Fig. 3).

Fig. 4 presents a phylogenetic tree in which a single confusion is identified, the rest of the images being correctly associated in the proper cluster.

In the last experiment applied to this set, our goal was to analyze the accuracy of the classification. Binary classification involves using distances to train an SVM model, based on which images belonging to a rural or urban area will be classified. In the case of this experiment, pCompLearn achieved an accuracy of 60%, while NCD-NVCOMP confirmed an accuracy of 75%. In this experiment, we have also tested the accuracy obtained using the distance matrix generated by the JPEG compressor. The result showed an accuracy of 68%.

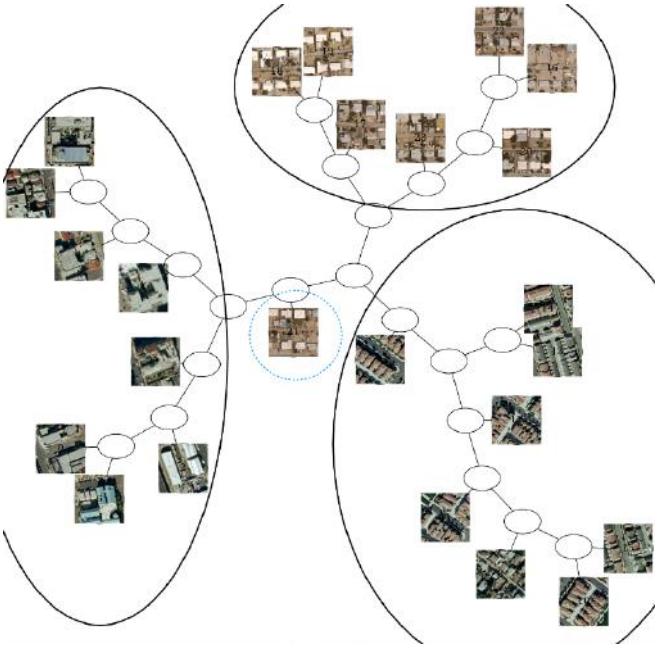

Figure 4. Unrooted tree computed using NCD-NVCOMP for UC Merced Land-set. Black solid line represents groups; blue dotted line represents confusion; red dashed line represents false alarm

BigEarthNet is a benchmark archive, consisting of 590,326 pairs of Sentinel-1 and Sentinel-2 image patches. We used BigEarthNet with Sentinel-2 image patches, i.e., BigEarthNet-S2, which consists of 125 Sentinel-2 tiles acquired between June 2017 and May 2018 over the 10 European countries. Each image patch was annotated with multiple land-cover classes, i.e., multi-labels, that were provided from the CORINE Land Cover database of the year 2018 [10]. For our experiments, subsets with size between 50 and 500 images were picked to test the computational speed and the accuracy of the proposed methods.

The subsequent experiment aimed to track the computation time of the distance matrix as the dataset was increasing. The size of an image is 385 KB and all 12 bands contained in a Sentinel-2 image patch were used. The computation time obtained was calculated as the average of 10 successive runs.

The last experiment involved the construction of the unrooted tree for 25 multispectral multi-label BigEarthNet images. The evaluation consisted in determining the number of common classes of images derived from the same parent node. In every resulted unrooted tree, we represented a cluster with a black circle. Above the circled area there were listed the common classes of the associated group (Fig. 7). The division into groups was done under supervision, combining the items that contain common and very similar labels, to highlight that the association depends on all classes.

To evaluate our results, we defined false alarm (red circle) as the assignment of a sample to an improper cluster and confusion (blue circle) as an item with similar labels, but not integrated into the correct cluster (Fig. 7 and Fig. 8).

For example, if we had 4 samples grouped in a subtree, the labels for that cluster would be the common labels between those samples. If one does not have common labels with the others, it will be marked as a false alarm. Confusion, in this scenario, intervenes when a sample, that is not integrated into any cluster, does not have a common parent with the samples from a neighboring cluster, but has common labels with them.

 Digital Object Identifier 10.32754/JMT.2022.1.07

Fig. 5 reveals the computational time obtained by the algorithms for batches of different sizes. The compression rate for NCD-NVCOMP, on average, was almost ten times better than the one obtained by pCompLearn.

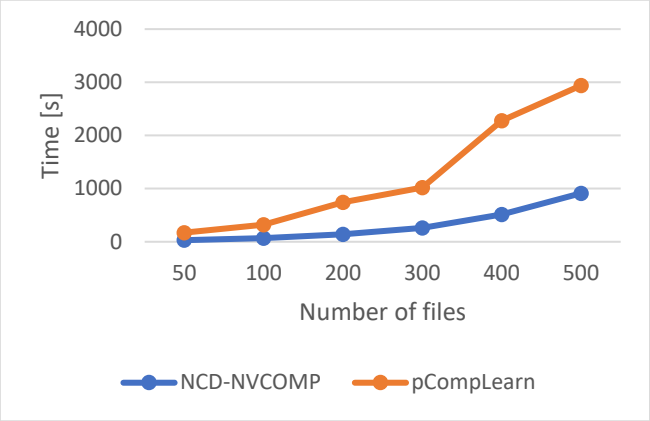

Figure 5. NCD-NVCOMP and pCompLearn comparison in computational time in relation to the cardinality of the input BigEarthNet dataset

In Fig. 6 there are listed images that have been used for the experiment. On the bottom of every image there were displayed labels, which have been explained in the legend. For example, the first image contains non-irrigated arable land, land mainly occupied by agriculture, and complex cultivation patterns. The information about the labels was taken from BigEarthNet creators.

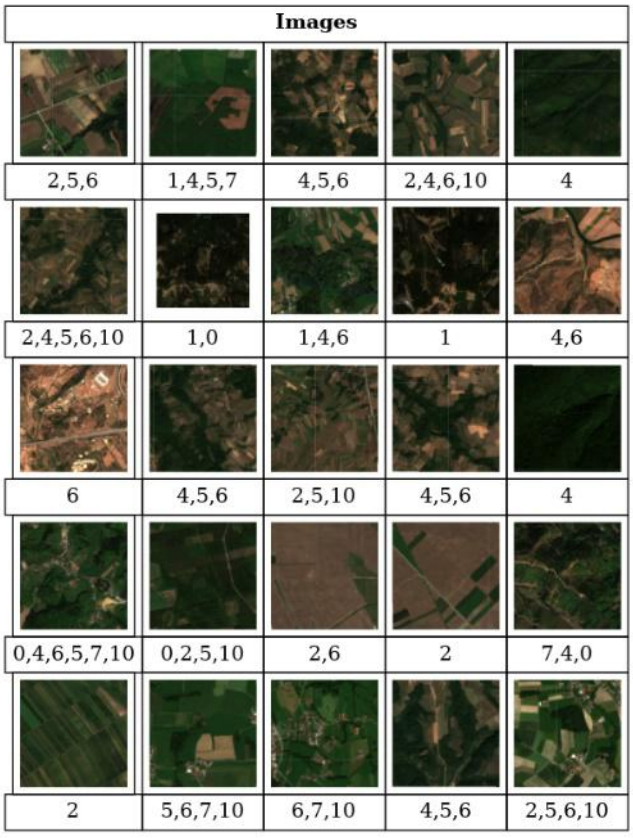

Figure 6. Subset of selected BigEarthNet samples; label 0 represents mixed forest; label 1 represents coniferous forest; label 2 represents non-irrigated arable land; label 3 represents transitional woodland; label 4 represents broad-leaved forest; label 5 represents land principally occupied by agriculture; label 6 represents complex cultivation patterns; label 7 represents pastures; label 10 represents discontinuous urban fabric

Fig. 7 shows the result of applying the maketree utility on the distance matrix calculated from the subset BigEarthNet using pCompLearn. Four false alarms and three confusions were obtained. Only one image in the false alarms was incorrectly associated with the image the parent leaf has in common with, the one in the bottom right group with classes 5 and 7.

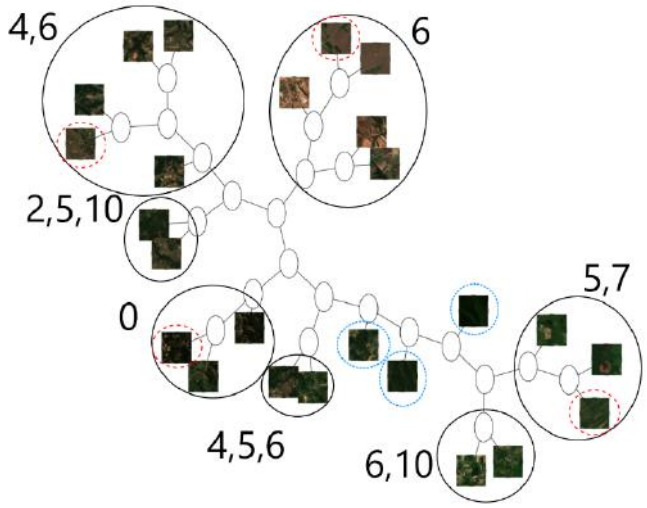

Figure 7. Unrooted tree computed using pCompLearn for BigEarthNet subset. Black solid line represents groups; blue dotted line represents confusion; red dashed line represents false alarm

In the phylogenetic tree presented in Figure 8, generated using the NCD-NVCOMP method, three false alarms and two confusions were identified. Compared to the result obtained using the pCompLearn method, a lower number of false alarms was obtained. We mentioned here that the images were correctly associated next to each other, but the three false alarms came from the small number of labels compared to the rest of the groups.

False alarms that were identified in Fig. 8 were also identified in Fig. 7 because those images have only one label. The number of labels for a multi-spectral sample is important because it increases the chances of an item to share labels with more than one cluster.

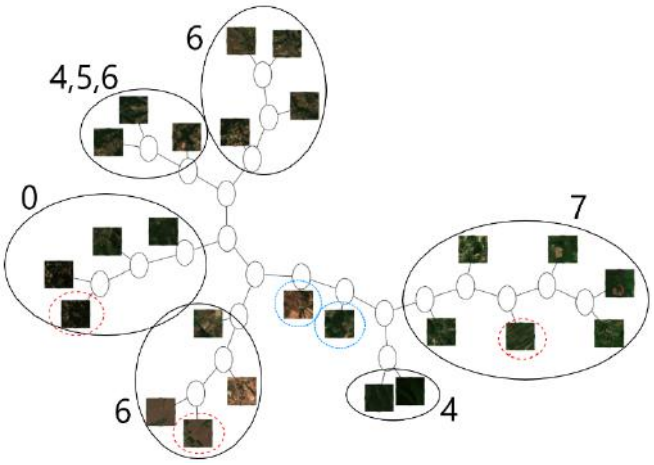

Figure 8. Unrooted tree computed using NCD-NVCOMP for BigEarthNet subset. Black solid line represents groups; blue dotted line represents confusion; red dashed line represents false alarm

V. Conclusion

The paper presented a faster approach to computing NCD metrics using compression solutions that run on GPU and multi-thread CPU. This approach, by using nvCOMP as

compressor, made it possible to obtain NCD metrics faster than in the original CompLearn, while keeping similar accuracy. In terms of clustering and classification, the nvCOMP compressor achieved better results than zlib, the default compressor used in CompLearn.

Future work will involve using the NCD-NVCOMP method with other types of objects in areas such as malware analysis, where, based on the distance matrix, executable files will be classified as benign or malicious. Due to the large volume of data that needs to be compressed in most cases, the usage of the NCD-NVCOMP algorithm to calculate the distance matrix in an optimum time is a robust candidate.

REFERENCES

[1] J. Zhang, W. Hsu and M. Lee, "Image Mining: Issues, Frameworks and Techniques," August 2003.

[2] R. Cilibrasi and P. M. B. Vitanyi, "Clustering by compression," *IEEE Transactions on Information Theory*, vol. 51, pp. 1523-1545, 2005. doi: 10.1109/TIT.2005.844059

[3] M. Cebrián, M. Alfonseca and A. Ortega, "The normalized compression distance is resistant to noise," *IEEE Transactions on Information Theory*, vol. 53, p. 1895–1900, 2007. doi: 10.1109/TIT.2007.894669

[4] R. Cilibrasi, *Complearn*, 2015.

[5] M. Coca, A. Anghel and M. Datcu, "Normalized Compression Distance for SAR Image Change Detection," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018. doi: 10.1109/IGARSS.2018.8518126

[6] M. Coca, A. Anghel and M. Datcu, "Unbiased Seamless SAR Image Change Detection Based on Normalized Compression Distance," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, pp. 2088-2096, 2019. doi: 10.1109/JSTARS.2019.2909143

[7] S. Cui and M. Datcu, "A comparison of Bag-of-Words method and normalized compression distance for satellite image retrieval," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 4392-4395. doi: 10.1109/IGARSS.2015.7326800

[8] D. L. Nikolay Sakharnykh and B. Karsin, *nvCOMP*, 2020.

[9] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010. doi: 10.1145/1869790.1869829

[10] G. Sumbul, M. Charfuelan, B. Demir and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019. doi: 10.1109/igarss.2019.8900532

[11] B. David, "Comparison of Lossless Image Formats," *arXiv preprint arXiv:2108.02557*, 2021, doi: 10.48550/ARXIV.2108.02557.

[12] D. Cerra and M. Datcu, "A Fast Compression-Based Similarity Measure with Applications to Content-Based Image Retrieval," *J. Vis. Comun. Image Represent.*, vol. 23, p. 293–302, February 2012. doi: 10.1016/j.jvcir.2011.10.009

[13] K. Shyni, "Lossless LZW Data Compression Algorithm on CUDA," *IOSR Journal of Computer Engineering*, vol. 13, pp. 122-127, January 2013. doi: 10.9790/0661-131122127

[14] D. Cortés, M. Nakano, H. Koga and H. Perez, "Evaluation of Image Descriptors for Urban-Rural Classification of Aerial Images.," in *SoMeT*, 2017, doi: 10.3233/978-1-61499-800-6-204.

[15] R. Borbely, "On normalized compression distance and large malware: Towards a useful definition of normalized compression distance for the classification of large files," *Journal of Computer Virology and Hacking Techniques*, vol. 12, pp. 1-8, December 2015. doi: 10.1007/s11416-015-0260-0

[16] A. Clark, *Pillow (PIL Fork) Documentation*, 2015.