N-Gram: PToPCNN : An Improved Image Classification Using Patch-To-Patch Convolutional Neural Network With Sparse Based N-Gram Feature Extraction Method

P.DOLPHIN DEVI1 & K.CHITRA2
1Vice Principal,Kalvi Matriculation Higher Secondary School, Dindigul, India
2Professor, Department of computer science Government Arts College Melur, Madurai, India

Abstract

Multisensor fusion is of abundantsignificance in Earth observation related applications. For instance, multispectral images (MSIs) deliverfull spectral data while light detection and ranging (LiDAR) data deliverheight information, and using MSI and LiDAR data together can attainimproved classification performance. In this work, an unsupervised feature extraction model, calledas N-gram along with classifier called as patch-to-patch convolutional neural network (N-Gram: PToP CNN), is proposed for collaborative classification of mutispectral and LiDAR data. Sparse coding, or sparse dictionary learning, is an unsupervised learning algorithm, and is skilled of mining features based simply on how well those features can be familiar torebuild the original image. With respect to imagepatches, we acquire sparse dictionaries for n-grams, constant sequences of bytes, of various sizes.More specific, a three-tower PToP mapping is first established to find an accurate illustration from MSI to LiDAR data, aiming at merging N-gram features between two different images. Then, by combining hidden layers of the designed PToP CNN, extracted features are projected to retain deeply fused characteristics. Hence, features from various hidden layers are merged into a stacked vector and provide into three fully inter connected layers. To prove the efficiency of the proposed classification structure, experiments are implemented on benchmark remote sensing data sets. The experimental results prove that the proposed method offersgreater performance when compared with some state-of-the-art classifiers, such as two-branch CNN and context CNN.

Keywords: Mutispectral image, classification,. N-gram, CNN, patch to patch

I.INTRODUCTION

In digital world, content-based analytics are often required to classify imagepatches in the lack of otherrecognizingdata. During image recovery, for example, thepatches of images on damaged media or memory dumpsresults in images that appear corrupted or missing while thedata is present [18]. For imagemodel, in order to recreate thecomplete images from these patches, analysts must determinewhich patches might go with which image. Since the searchspace of where each patch may possiblyfit to be solarge, automated tools are necessary in creating this timeconsuming process practical.

Sensortechnology has knowledgeablesignificant advances [1]–[4] lately, letting us to degree various aspects of the items on the surface of Earth. Remotelysensed hyperspectral images (HSIs) delivercomplete spectral material to exclusivelycategorize various materialsof interest, leading to greater classification of land-coverclasses [5]–[8]. Though, in assuredsituations, it maybe essential to resort to different source to complement theinfodelivered solely by hyperspectral instrumentsfor additionalsuccessful and/or refining classification. For thispurpose, a series of methods have been examined inthe literature for merging of data together from differentsources [9], [10]. Light detection and ranging (LiDAR) data,which deliveraltitude information about the measured area,are very useful source for perfecting the informationdelivered solely by HSI [11], [12]. Collaborative classificationof HSI and LiDAR has been widelydeployed in variousapplications, for example complex area classification [13], forestfire management [14], etc., due to its fine performance. Severalstudies have specified that classification performance can bedeveloped after incorporating HSI and LiDAR data. For example,in [15], LiDAR was intended for the scene segmentation andHSI data for classifying the segmented regions; in [16],Ghamisi et al. exploited morphological extinction-profiles tomine both HSI and LiDAR features; and in [17], Rasti et al.developed extinction profiles for joint feature extraction, followed by total variation component analysis for additionalfusion.

In compare to hand-engineering features, we suggest anmethod for computerized feature extraction from image patchesusing sparse coding, also known as sparse dictionary learning(details in Section II-A). This method has a number ofbenefits over hand-engineered features. Mostly, the features that are extracted over sparse coding are establish in anunsupervised manner, as opposed to features that might needto be laboriously created to be proper for a specificdomain [19]. Also, because the features mined bythis methodologyreduce reconstruction error, they capture aimportant amount of information about the domain withoutrequiring prior domain knowledge. Additionaladvantages of thisapproach, due to the sparsity constraint, is the extraction offocused features directed to each particular file type, or evenwithin file types comprising more complex internal structure(e.g. doc, pdf, zip) [20].

The rest of the sections of the paper are ordered asfollows: in Section II we deliver background on the sparsecoding algorithm as well as the patch-to-patch convolutional neural network classifier; in SectionIII we define our experiments on how we use our sparsecoding method to extract features from imagepatches data andtrain our classifier to differentiate among multiple images;in Section IV we deliver and deliberate the results from ourexperiments; and in Section V we make concluding remarks.

II.BACKGROUND

A. Sparse coding

Sparse coding, or sparse dictionary learning, is a way of modeling data by decomposing it into sparse linear combinations of elements of a given basis set [19], [21]. That is, a data vector $x \in R_m$ may be approximated as multiplying a dictionary matrix $D \in R^{m \times k}$ with a sparse representation vector $r \in R^k$: $x \approx D_r$. Here, a vector is said to be sparse when only a small

fraction of the entries are nonzero. Althoughit is called sparse dictionary learning, the dictionary D is notnecessarily sparse.

When dictionary D is known (e.g. a wavelet basis), acommon method for finding an associated sparse vector isthrough regression analysis, which may be formulated as theL_1-regularized optimization problem,

$$l(x, D) = \min_{r \in \mathbb{R}^k} \frac{1}{2} \|x - Dr\|_2^2 + \lambda \|r\|_1 \qquad (1)$$

where the cost may be understood as the contributions of thereconstruction error $\frac{1}{2}||x-Dr||_2^2$ and a sparsity penalty$\lambda\|r\|_1$and λ is a regularization parameter. This particular formulationis also known as the lasso [22].

When dictionary D is not known, or it is more desirable tolearn a dictionary more representative of the data, an extendedformulation to the above optimization problem gives the sparsecoding cost function,

$$f_n(X) = \min_{D \in \mathbb{R}^{m \times k}, R \in \mathbb{R}^{k \times n}} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2} \|x_i - Dr_i\|_2^2 + \lambda \|r_i\|_1 \right) \qquad (2)$$

where X = $\{x_1, x_2, \ldots x_n\}$are the data vectors from which wewant to learn a dictionary. Because the optimization occursover both the dictionary D and the set of sparse representation vectors R =$\{r_1, \ldots r_n\}$, most approaches to sparsecoding iteratively fix one variable while minimizing the other.For learning dictionaries from large input data sets, onlineor streaming methods have been developed [23]. Regardinginitialization, a common method for initializing the dictionary**D** is seeding it with random elements from the training set,and the sparse representation vectors ***R***are initialized as zerovectors.

B. End-to-End Architecture for Image Analysis

As we see, the most corporate usage of convolutional networks is for classification tasks [24], where the output to an image is an only class label. Though, in numerous visual tasks, the preferred output should contain localization; such as, a class label is made-up to be allocated to each pixel. End-to-end designs of existing networks, which expect dense outputs from arbitrary-sized inputs, are trained end-to-end and pixel-to-pixel in image analysis, e.g., semantic segmentation. Both learning and suggestion are implemented whole-image-at-a-time by dense feed-forward computation and back-propagation. In the end-to-end network, the subsampled combining operations allow the learning process, while up-sampling layers allow pixel-wise expectation [25]. In other words, the end-to-end architecture for image segmentation normallycontains an encoder path for setting capturing and a symmetric decoder path for exact localization. Broadly used deep architectures for segmentation have same end-to-end structure as illustrated in Fig. 4, butcontrast in the form of the encoder–decoder network scheme and training strategy. Such as, Long et al. [25] constructed fully convolutional networks that acquired input of random size and createdindividually sized output with well-organized inference and learning. Furthermore, in [24], an architecture called as U-Net consisted of an increasing path and andevelopment path, which could be trained end-to-end using very few sample images.

III.PROPOSED WORK

In above-mentionedapproaches, feature extraction approaches withencoder–decoder designexclusivelycontain one type of datasource. Clearly, the encoder–decoder structure holds twoqualities: 1) feature extraction and 2) classification.Such as, image segmentation tasks always map an imagefrom one area to another area. Different from existing single-source feature extraction techniques, we efforts onencoder–decoder design to incorporate two-domain translation during feature extraction process, thereforeallowing theseamless fusion of HSI and LiDAR data.

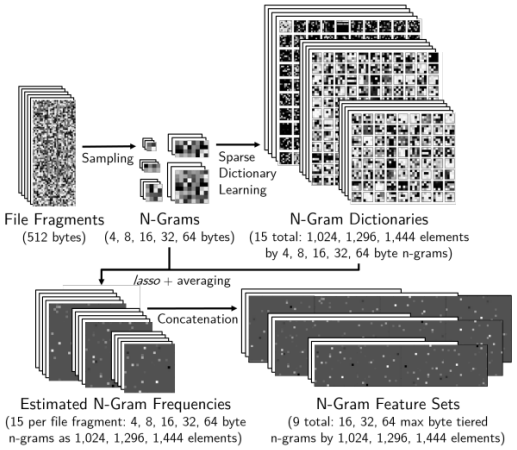

Fig 1 Feature Extractor extraction of N-gram method

In this section, an unsupervised feature extraction methodfor image data is defined in detail,and network training is explained. As shown in Fig. 1, theproposed structure includes a three-tower feature extractorcalled N-Gram: PToPCNN, followed by a classifier that containing fullyconnected layers with softmax loss.

A.LEARNING DICTIONARIES OF N-GRAM

In sparse coding, or sparse dictionary learning, the goal is often to obtain an overcomplete basis set from the data vectors [26]. From the file fragment datasets described above, we learn a dictionary over n-grams similar to that of learning a dictionary from image patches or slices of other natural signals. Unlike natural signals which contain a great deal of redundancy, byte data is often represented in a relatively concise format for efficiency. As a result, the ratio of the elements in our learned dictionaries to the size of the n-grams is significantly higher than what is more commonly found in the sparse coding literature [19]. For example, given an n-gram size of 64 bytes, a dictionary size of 1024 would give a ratio of 16 : 1.

For learning the dictionary, only n-grams from file fragments set aside for training were used. The dictionaries were learned using the spams sparse coding library which provides a scalable online algorithm for dictionary learning based on stochastic gradient descent [23]. Learning occurred over two epochs (i.e. passes over the training set) and was performed by iterating over randomly shuffled and balanced batches of n-grams using a sparsity penalty of λ = 0:15 (a default value in spams). For each batch, 25 files per file type were randomly sampled, for a total of 500 file fragments per file type. From each file fragment, 64 randomly sampled n-grams were used for dictionary learning, giving a total of 32,000 n-grams per file type, or 576,000 n-grams per batch. The sampling of N-grams from file fragments was performed without replacement, but allowing for overlap between different n-grams. The choice of randomly sampling 64 n-grams, as opposed to using all possible (overlapping) n-grams from each fragment, was done more as a practical matter to save time during training.

In total, 15 dictionaries were learned, differentiated by the number of elements in the dictionary (1,024, 1,296, and 1,444 elements) as well as the size of the n-gram (4, 8, 16, 32, and 64 bytes). A sample learned dictionary is shown in figure Fig. 2. Although there are a few elements with identifiable patterns, the vast majority of the elements appear to be random. This result is not surprising as the dictionary is learned from byte patches from files that have both low entropy (e.g. html) and high entropy (e.g. gz) with respect to their byte distributions.

B.N-GRAM FREQUENCIES

Once the dictionaries have been learned, we can use the standard lasso method to transform the file fragments from a collection of n-grams to a collection of sparse feature vectors. Here, unlike when sampling for dictionary learning, all n-grams from the file fragment are used. The transformed sparse feature vectors may then be averaged together to provide an estimate of n-gram frequencies based on the dictionary elements, allowing for significantly larger n-gram sizes than are typically found in existing methods which suffer from combinatorial explosion. Here, instead of representing each possible n-gram combination as a "one-hot" entry in a 256n feature vector, the n-gram frequencies are approximated according to the dictionary elements. This is because the sparse feature vectors provide a measure of how much each dictionary element contributes to the reconstruction of the n-grams in the file fragment.

To construct the feature sets to be used in classification, the n-gram frequencies for different n-gram sizes (4, 8, 16, 32, and 64 bytes) were concatenated. Instead of examining each possible combination, we constructed feature sets according to multiple tiers ordered by maximum n-gram size. This was done such that each progressively larger tier contained a superset of features with respect to the tiers before it (e.g. a feature set with n-gram frequencies up to 32 bytes also contained the frequencies for n-grams of 4, 8, and 16 bytes). Following this construction, we used three different tiers with maximum N-gram size of 16, 32, and 64 bytes.

C. CLASSIFICATION PROCESS OF PTOP CNN

For the PToP CNN, training patches are collected by adopting a sliding window in an unsupervised way, as depicted in Fig. 1. Both HSI and LiDAR patches are collected through the process shown in Fig. 2, and each patch-pair of HSILiDAR is acquired over the same area, thus ensuring the high correlation between two-source data for further joint feature extraction. The value of S (moving step length) is set to 2 with 11 × 11 window size, and the total number of training samples stands at about [(Width×Height)/S2] (Width and Height are the spatial size of the image), ensuring sufficient training samples. For the hierarchical fusion stage and final classification, a simple but effective data augmentation method is utilized, which produces additional data without introducing extra labeling costs. Specifically, a random seed is generated for controlling counter-clockwise rotation angle, 90◦, 180◦, 270◦, and 360◦ in the training phase.

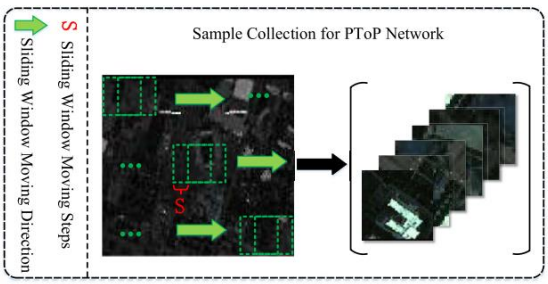

Fig 2 Process of constructing training patches from original image

Then, the training process is divided into three stages as shown in Fig. 1. In the first stage, training patches derived from Source I (HSI data) are fed into the designed PToP network, and the detailed network configuration is depicted in Fig. 4. After that, input samples flow through the PToP network to obtain features of different hierarchy (different filter scales, different layers). Therefore, the input of "hierarchical fusion module" is acquired, where the detailed parameter setting and network structure are shown in Fig. 3. The right half of Fig. 3 illustrates one of the branches, a specific layer L, including some fixed operation, convolution and batch normalization. Since optimizing parameters in 8 branches simultaneously is difficult, each branch of hierarchical fusion module is trained separately. When the 8 branches are merged, the pre-trained feature extractor extracts the corresponding features from input data with their fully connected layer and softmax prediction layer being removed. The remaining layers in 8 branches are fixed or trainable with a small learning rate of SGD rule. All the branches are concatenated to generate the final informative feature vector.

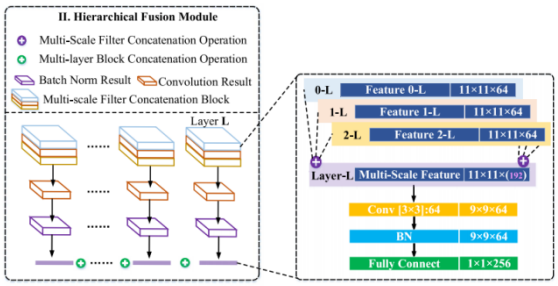

Fig 3 Architecture of hierarchical fusion module

During the learning process, all data are normalized to a range of 0-1 for accelerating convergence process of the network. Weights and bias of all the convolutional layers are initialized with Glorot normalization, and then updated with small learning rate.

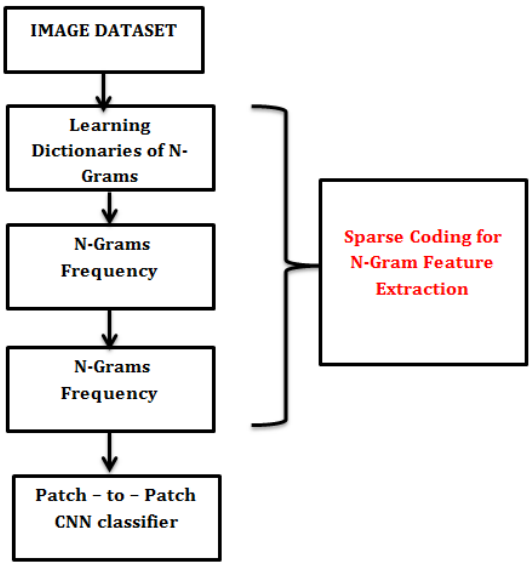

Fig 4Implementation flow of proposed method

IV.RESULT AND DISCUSSION

The entire experimental setup fromsampling imagepatches from the dataset to classification wasperformed over 10 independent images. From the classificationresults, we evaluated the performance of our approach using anumber of metrics. In addition to the raw prediction accuracy,we also computed the F1 score, which provides a weightedaverage of the precision and recall of the classifier.

Compared to existing work on file fragment classification as well as found during our replication studies, we found that the features obtained using our sparse coding approach performed significantly better, especially when the features were used in supplement to existing hand-engineered features. It can be described in Fig 7.

Fig 5 Input image

Fig 6 Classified Image

Fig. 8 further lists the classification performance with different numbers of training samples to evaluate the sensitivity of all methods to the training-samples size. The percentage of training samples is changed from 20% to 100%. Obviously, the proposed method consistently outperforms other methods.This verifies the proposed framework is robust to small training sample sizes.

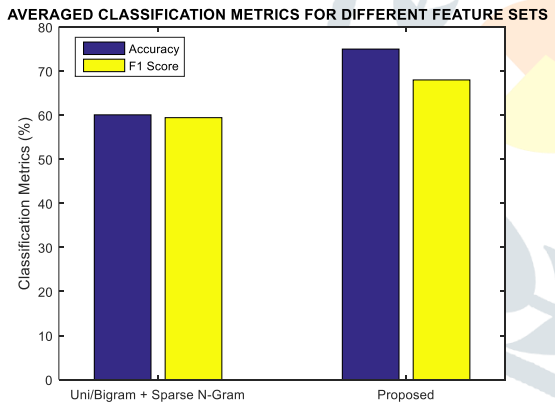

Fig 7 Performance Comparison based on feature set

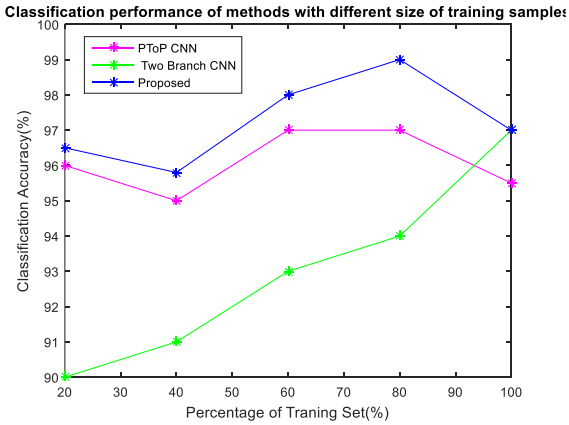

Fig 8 Performance Comparison based on classifier

V.CONCLUSION

We have proposed an approach for imagepatch classification using sparse dictionary learning to estimate n-gram frequencies along with PToP CNNclasifierfor a given file fragment. Notably, the dictionaries learned using this approach can be used to extend the feature of n-gram frequencies without suffering from combinatorial explosion because n-grams are represented as the reconstruction of sparse vectors as opposed to being expressed exactly. Although the features extracted by this approach are done so in an unsupervised manner, they are able to capture a significant amount of information present in the byte patterns without needing prior domain knowledge. In fact, due to the sparsity constraint, the approach tends toward the extraction of redundant byte patterns specific to individual image types. Experimentally, we found that these features yielded significantly better classification results with respect to existing methods, especially when the features were used in supplement to existing hand-engineered features.

Although the proposed sparse coding approach achieves competitive image patch classification on its own, leveraging domain expertise may yield still more accurate classifiers. The PToP CNN model was proposed for N-gram feature extraction, to take full advantages of very wealth spectral information and spatial/contextual information contained in HSI and LiDAR data. Experimental results demonstratedthat thefeature extractor N-gram in conjunction with the hierarchical fusion module could simultaneously utilize the information of HSI andLiDAR data to achieve excellent collaborative classification performance.

REFERENCES

[1] X. Lu, Y. Yuan, and X. Zheng, “Joint dictionary learning formultispectral change detection,” *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 884–897,Apr. 2017.

[2] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, “Joint reconstruction and anomaly detection from compressive hyperspectral images usingMahalanobis distance-regularized tensor RPCA,” *IEEE Trans. Geosci.Remote Sens.*, vol. 56, no. 5, pp. 2919–2930, May 2018.
[3] S. Jia, L. Shen, J. Zhu, and Q. Li, “A 3-D Gabor phase-based codingand matching framework for hyperspectral imagery classification,” *IEEETrans. Cybern.*, vol. 48, no. 4, pp. 1176–1188, Apr. 2018.

[4] M. Zhang, W. Li, and Q. Du, “Diverse region-based CNN for hyperspectral image classification,” *IEEE Trans. Image Process.*, vol. 27, no. 6,pp. 2623–2634, Jun. 2018.

[5] W. Li, E. W. Tramel, S. Prasad, and J. E. Fowler, “Nearestregularizedsubspace for hyperspectral classification,” *IEEE Trans. Geosci. RemoteSens.*, vol. 52, no. 1, pp. 477–489, Jan. 2014.

[6] X. Zheng, Y. Yuan, and X. Lu, “Dimensionality reduction by spatial–spectral preservation in selected bands,” *IEEE Trans. Geosci. RemoteSens.*, vol. 55, no. 9, pp. 5185–5197, Sep. 2017.

[7] Z. Wu *et al.*, “GPU parallel implementation of spatially adaptive hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ.Remote Sens.*, vol. 11, no. 4, pp. 1131–1143, Apr. 2018.

[8] L. Zhang *et al.*, “Simultaneous spectral–spatial feature selection andextraction for hyperspectral images,” *IEEE Trans. Cybern.*, vol. 48, no. 1,pp. 16–28, Jan. 2018.

[9] M. Khodadadzadeh, A. Cuartero, J. Li, A. Felicísimo, and A. Plaza,"Fusion of hyperspectral and LiDAR data using generalized compositekernels: A case study in extremadura, Spain," in *Proc. IGARSS*, Milan,Italy, Jul. 2015, pp. 61–64.

[10] M. Zhang, W. Li, and Q. Du, "Collaborative classification of hyperspectral and visible images with convolutional neural network,"*J. Appl.Remote Sens.*, vol. 11, no. 4, 2017, Art.no. 042607.

[11] J. Jung, E. Pasolli, S. Prasad, J. C. Tilton, and M. M. Crawford,"Aframework for land cover classification using discrete return LiDARdata: Adopting pseudo-waveform and hierarchical segmentation," *IEEEJ. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 2,pp. 491–502, Feb. 2014.

[12] M. Zhang, P. Ghamisi, and W. Li, "Classification of hyperspectral andLiDAR data using extinction profiles with feature fusion," *Remote Sens.Lett.*, vol. 8, no. 10, pp. 957–966, 2017.

[13] M. Dalponte, L. Bruzzone, and D. Gianelle, "Fusion of hyperspectraland LiDAR remote sensing data for classification of complex forestareas," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1416–1427,Jun. 2008.

[14] B. Koetz, F. Morsdorf, S. Van der Linden, and B. Allgöwer, "Multisource land cover classification for forest fire management based onimaging spectrometry and LiDAR data," *Forest Ecol. Manag.*, vol. 256,no. 3, pp. 263–271, Jul. 2008.

[15] D. Lemp and U. Weidner, "Improvements of roof surface classificationusing hyperspectral and laser scanning data," in *Proc. ISPRS*, Tempe,AZ, USA, Mar. 2005, pp. 14–16.

[16] P. Ghamisi, B. Höfle, and X. Zhu, "Hyperspectral and LiDAR datafusion using extinction profiles and deep convolutional neural network,"*IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6,pp. 3011–3024, Jun. 2017.

[17] B. Rasti, P. Ghamisi, and R. Gloaguen, "Hyperspectral and LiDARfusion using extinction profiles and totalvariation component analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3997–4007, Jul. 2017.

[18] S. L. Garfinkel, "Carving contiguous and fragmented files with fast object validation," Digital Investigation, vol. 4, Supplement, pp. 2 – 12, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287607000369.

[19] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparsecoding algorithms," in *Advances in Neural Information ProcessingSystems 19*, B. Schlkopf, J. Platt, and T. Hoffman, Eds. Cambridge,MA: MIT Press, 2006, pp. 801–808. [Online]. Available: http://books.nips.cc/papers/files/nips19/NIPS2006 0878.pdf.

[20] V. Roussev and S. L. Garfinkel, "File fragment classification-thecase for specialized approaches," in *Proceedings of the 2009 Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering*, ser. SADFE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 3–14. [Online]. Available:http://dx.doi.org/10.1109/SADFE.2009.21

[21] V. M. Patel and R. Chellappa, "Sparse representations, compressivesensing and dictionaries for pattern recognition," in *in Asian Conferenceon Pattern Recognition (ACPR)*, 2011.

[22] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[23] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrixfactorization and sparse coding," *Journal of Machine Learning Research*,vol. 11, pp. 19–60, January 2010.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutionalnetworks for biomedical image segmentation," in *Medical ImageComputing and Computer-Assisted Intervention*. Cham, Switzerland:Springer, Oct. 2015, pp. 234–241.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networksfor semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit.*,Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[26] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" Vision Research, vol. 37, no. 23, pp. 3311 – 3325, 1997.[Online]. Available: www.sciencedirect.com/science/article/pii/S0042698997001697