Fall 2021

Sparse Coding for Data Augmentation of Hyperspectral Medical Images

Rojin Zandi
San Jose State University

SPARSE CODING FOR DATA AUGMENTATION OF HYPERSPECTRAL MEDICAL IMAGES

A Thesis

Presented to

The Faculty of the Department of Electrical Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Rojin Zandi

November 2021

The Designated Thesis Committee Approves the Thesis Titled

SPARSE CODING FOR DATA AUGMENTATION OF HYPERSPECTRAL MEDICAL IMAGES

by

Rojin Zandi

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

SAN JOSÉ STATE UNIVERSITY

November 2021

Birsen Sirkeci, Ph.D.	Department of Electrical Engineering
Robert Morales-Zaragoza, Ph.D.	Department of Electrical Engineering
Chang Choo, Ph.D.	Department of Electrical Engineering

ABSTRACT

SPARSE CODING FOR DATA AUGMENTATION OF HYPERSPECTRAL MEDICAL IMAGES

by Rojin Zandi

Hyperspectral imaging presents detailed information about the electromagnetic spectrum of an object in three dimensions. The significant point about the hyperspectral images is that it contains tens or hundreds of spectral layers, which provide precise data about the composition of the studied material. Therefore, hyperspectral images have been popular in many fields of study, such as medical diagnostic imaging. Speed and precision are key points to save human life in disease diagnosis, and applying machine learning techniques to medical hyperspectral images helps answer this need. Convolutional neural networka are one of the most popular machine learning methods for classifying medical images. However, training neural networks, in general, requires a large dataset, and the small size of medical imaging datasets results in a problem. In this thesis, we propose sparse coding algorithms to regenerate the hyperspectral data and feed it to the CNN model for training. This issue can be solved with the help of sparse coding algorithms. We focus on a colon cancer hyperspectral image dataset and different sparse coding methods utilizing K-SVD and A+ (with and without patching) as dictionary learning methods. The new reconstructed images have been added to the original image set and provided three new training sets with doubled number of images (246) for training the CNN. Using the augmented datasets, the test accuracy has risen to 86.53%, which is 30.13% higher than the original dataset (56.4%). We have also generated another dataset which is a mixture of the three reconstruction methods, and increased the number of training images to 266. Using the mixed dataset, the accuracy has reached 94.23%, and the difference between the test and training accuracy has dropped by 15.42%. Also, the precision has increased to 100%, which means there is no non-malignant image classified as a lesional image.

ACKNOWLEDGMENTS

Above all, I am deeply greatful of Prof. Birsen Sirkeci, for her memorable support at each step of this study. Her immense knowledge and plentiful experience have encouraged me in all the time of my academic research.

Many thanks to members of my committee, Professor Robert Morales Zaragoza and Professor Chang Choo, for providing a number of helpful comments and suggestions.

I would also like to thank my family. Without their support, I would have never been able to complete this thesis and pursue my dreams.

DEDICATION

I would like to dedicate my thesis to Pouneh Gorji and all other 175 passengers and crew of the PS-752 flight (Tehran-Kyiv), who have been shot by two missiles. Pouneh was a graduate student at the University of Alberta, and her last work has been cited in this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Hyperspectral imaging is a technological breakthrough for collecting imagery data with more spectral details. Each hyperspectral image contains three dimensions, including two spatial and one spectral dimension [1], as shown in Figure.1. The spectral dimensions are similar to RGB (Red-Green-Blue) images, but the spectral dimension contains more than three layers or channels. Increasing the number of spectral channels provides more data about the electromagnetic spectrum of the scene. Every material has a specific spectral response, and hyperspectral images aim to capture these responses, which helps us to understand the material composition of an object [2]. Furthermore, RGB images are not able to tap the spectral information of the target object because of being highly quantized. Hence, hyperspectral images are preferred in many study cases in comparison with RGB images.

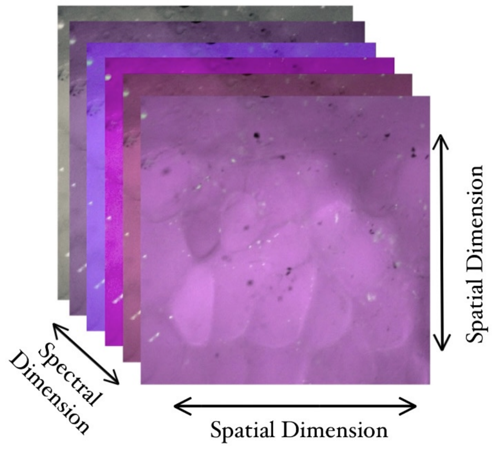

Fig. 1. Dimensions of a hyperspectral image.

The ability to collect spectral details of an object in an image has been desirable in different applications, such as agriculture [3], crime scene detection [4], remote sensing [5], and medical imaging [6], [7]. In medical applications, hyperspectral images can provide precise and nearly real-time information about the studied biomarker by the characteristics of the reflected spectrums. This aspect makes hyperspectral images helpful in diagnosing diseases, such as cancers [8], heart and artery diseases [9], shock [10], and retinal diseases [11]. In this thesis, our focus is on cancer detection.

The process of diagnosing cancer disease depends on the cancer type and its symptoms, and the medical imaging method is chosen based on these factors. Computed Tomography (CT), Magnetic Resonance Imaging (MRI), ultrasound, Positron Emission Tomography (PET), Mammography, Single-Photon Emission Computed Tomography (SPECT), and optical imaging are the main imaging methods for cancer detection [12], and they are usually used for early detection or after removing the tumor and chemotherapy to prevent the recurrence of the disease. Despite the fact of being commonly used, these methods have different limitations and even may cause side effects for the patient [13].

As mentioned above, hyperspectral imaging is another technique used for cancer diagnosis, which can obtain more spectral information about the tumor or lesioned area. Although medical hyperspectral images are more challenging to obtain and process, they have been achieving promising results, which is because of the development of machine learning algorithms and computational power [14].

In 2006, Seong G. Kong et al. [15] used hyperspectral images with 21 spectral layers for detecting cancerous tumors on mouse skin. The spectral range of images was 440 nm to 640 nm with 10 nm spectral resolution, which provided sufficient information for classifying the malignant skin tumors without biopsy. Another cancer diagnosis research via hyperspectral imaging was done on breast tumors, which is the most common cancer type among American women [16]. [17] studied 156 hyperspectral cubes of 56 female rats with the wavelength 450 to 700 nm that resulted in higher sensitivity and specificity in comparison with the histopathological method. As shown in Fig.2, after the female breast (11.7%) and lung (11.4%), prostate cancer has the most cases (7.3%) worldwide [18]. A prostate cancer diagnosis study has been done on 11 mice using hyperspectral imaging [19]. The dataset of this research contains images with the spectral range of 450 to 950 nm with 31 layers, and there are 1.4 million pixels in each image. To classify the data, they have applied least squares support vector machine, and it resulted in 96.9% specificity and 92.8% sensitivity.

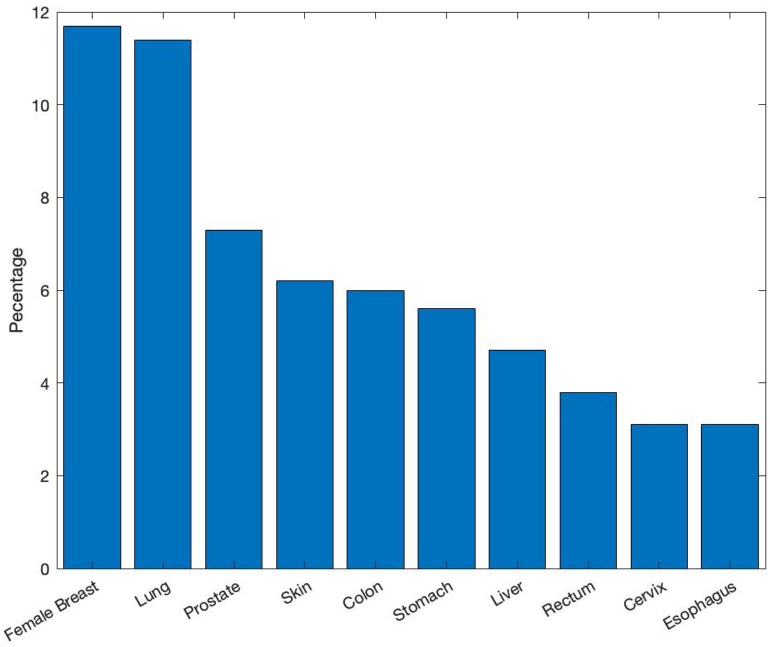

Fig. 2. Percentage of 10 most common cancer cases in 2020

In medical computer vision, deep learning methods are popular for disease diagnosis and classifying biomarkers. Mostly these methods require large datasets to train an accurate model which is critical for medical applications. Unfortunately, medical datasets are usually small and contain a few hundred images; hence researchers have suggested different methods to tackle this issue, such as probabilistic labels [20] and Generative Adversarial Networks (GAN) [21]. Furthermore, although hyperspectral images are precise and provide rich spectral information, they are expensive and complicated to obtain [22]. Collecting a medical hyperspectral image dataset is a challenging and prolonged process.

In this thesis, we aim to classify the lesional and non-lesional hyperspectral images of the colon cancer dataset [23]. This dataset contains 175 images of 13 patients in lesional and non-lesional classes. Previous work, done by Mobilia et al. [24], has applied an optimized Convolutional Neural Network (CNN) for hyperspectral image classification. Due to lack of enough data, the accuracy of the proposed model was not satisfying even though the dataset was enlarged using simple image position augmentation. In this thesis, we propose sparse coding methods for hyperspectral image data augmentation. Sparse coding algorithms aim to represent the data as a linear combination of a sparse representation and a dictionary. In Chapter 2, we explain sparse representation and dictionary learning algorithms and compare them. Chapter 3, reviews the colon cancer dataset and the work done by Mobilia et al. [24]. Chapter 3 also contains the pre-processing, model development, and their results. In Chapter 4, we apply different sparse coding methods to reconstruct the hyperspectral images of the colon cancer dataset and add the reconstructed images to the training set. We compare the performance of each reconstruction method and study their effect on classification. Finally, in Chapter 5, we present the conclusion and future work.

2 SPARSE CODING

Sparse coding methods are unsupervised algorithms that allow us to learn an over-complete set of K basis vectors. A sparse coding problem is divided into two parts, known as sparse signal representation and dictionary learning. In this chapter, we explain the structure of sparse coding and discuss the most applicable algorithms, which are required for sparse modeling, such as matching pursuit [25], orthogonal matching pursuit [26], least squares orthogonal matching pursuit [27], basis pursuit [28], and least absolute shrinkage and selection operator [29]. In addition, method of optimal direction [30], K-means [31], K-SVD [32] and A+ [33] methods will be discussed as dictionary learning methods.

2.1 Sparse Representation

Sparse coding has recently become popular for compressing, collecting, and reconstructing signals and images. It is advantageous in multifarious areas of machine learning, and signal processing, such as:

- Image classification [34]
- Image reconstruction [35]
- Edge detection [36]
- Clustering [37]
- Background subtraction [38]
- Image super-resolution [39]
- Face recognition [40]

A sparse coding algorithm is a simulation of a mammalian's visual cortex. When we see an image, our visual cortex generates an accurate representation of that object. Hence, sparse coding algorithms aim to extract statistically independent and meaningful structures in the image using a robust solution [41]. Let $\boldsymbol{x} \in R^n$ be our original image, using sparse representation, and every image can be modeled as a linear combination of:

$$\boldsymbol{x} \approx \boldsymbol{D\alpha} \tag{1}$$

where $\boldsymbol{D}$ is a set of basis vectors, also known as dictionary with K×N dimension size, and $\boldsymbol{\alpha}$ is the sparse representation of the image.

Figure. 3 shows the sparse coding problem, where the red pointers are the selected atoms for representing the data. The set of these atoms is known as support D_s. In this figure, the support set is $D_s = \{4,5,9\}$ and $L = 3$ which is the number of non-zero elements of $\boldsymbol{\alpha}$.

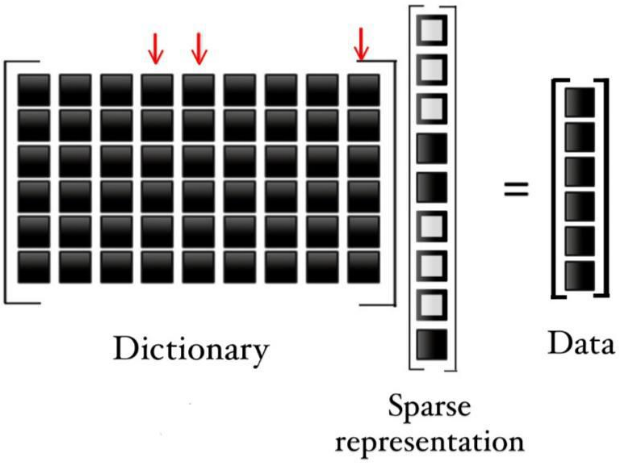

Fig. 3. Sparse coding: dictionary is multiplied with the sparse representation matrix. The red arrows show the non-zero values, which are the selected atoms (supports)

Considering that we are using approximation methods to model the data, there is a residual error ε in the equation:

$$\boldsymbol{x} = \boldsymbol{D\alpha} + \boldsymbol{\epsilon} \tag{2}$$

We aim to reduce the error, and the objective function of this optimization problem is:

$$\min_{\boldsymbol{\alpha} \in R^m} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D\alpha}\|_2 + \lambda G(\boldsymbol{x}) \tag{3}$$

where $\lambda G(\mathbf{x})$ is a regularization term that controls the sparsity of $\boldsymbol{\alpha}$, and choosing $G(\mathbf{x})$ depends on priorities of the problem such as smoothness, sparsity, and redundancy. In this thesis, we require smoothness and sparsity and the $G(\mathbf{x})$ function is defined by ℓ_0 norm:

$$\min_{\boldsymbol{\alpha} \in R^m} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D\alpha}\|_2 + \lambda \|\boldsymbol{\alpha}\|_0 \tag{4}$$

Considering that we use different norms, so it is useful to review their definition. In the following (Equations. 5 to 8) $\boldsymbol{a}$ and $\boldsymbol{b}$ are two vectors of which we aim to find the distance between d:

- ℓ_0 norm (Hamming distance):

$$d_0(\boldsymbol{a},\boldsymbol{b}) = \|\boldsymbol{a}-\boldsymbol{b}\|_0 = \sum_{i=1}^{d} \mathbb{I}(\boldsymbol{a}=\boldsymbol{b}) \quad \textit{where} \quad \mathbb{I}(\boldsymbol{a}=\boldsymbol{b}) = \begin{cases} 1 & \text{if } \boldsymbol{a}=\boldsymbol{b} \\ 0 & \text{if } \boldsymbol{a}\neq\boldsymbol{b} \end{cases} \tag{5}$$

  In later sections, ℓ_0 gives us the number of non-zero values in the sparse representation. Notice that ℓ_0 is non-convex and it makes the sparse coding problems more complex.

- ℓ_1 norm (Manhattan distance):

$$d_1(\boldsymbol{a},\boldsymbol{b}) = \|\boldsymbol{a}-\boldsymbol{b}\|_1 = \sum_{i=1} |\boldsymbol{a}_i - \boldsymbol{b}_i| \tag{6}$$

- ℓ_2 norm (Eucleadian distance):

$$d_2(\boldsymbol{a},\boldsymbol{b}) = \|\boldsymbol{a}-\boldsymbol{b}\|_2 = \sqrt{\sum_{i=1}^{d}(\boldsymbol{a}_i-\boldsymbol{b}_i)^2} \tag{7}$$

- ℓ_∞ norm, which in some cases is replaced with ℓ_1 norm, because it is the rotated version of ℓ_1 norm (Fig. 4):

$$d_\infty(\boldsymbol{a},\boldsymbol{b}) = \|\boldsymbol{a}-\boldsymbol{b}\|_\infty = \max_{i=1}^{d}\|\boldsymbol{a}_i-\boldsymbol{b}_i\| \tag{8}$$

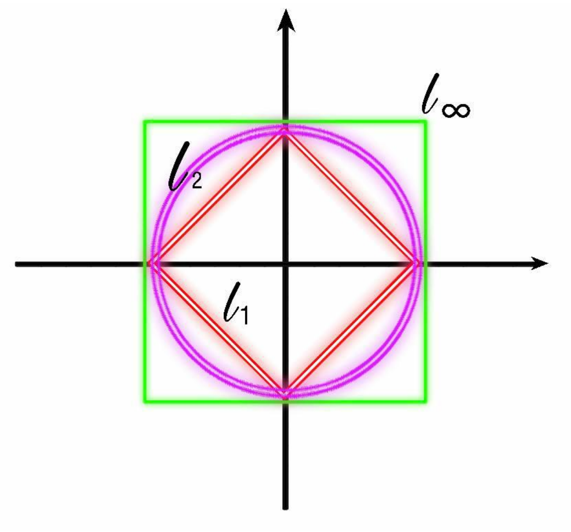

Fig. 4. Unit balls in $\mathbb{R}^2$ for the different norms.

There are two main methods to find the sparse representation: greedy methods [42] and relaxation methods [43], which are described in detail below.

1) Greedy method: An iterative algorithms, which searches for the sparsest solution in each iteration, known as greedy method. In each iteration, it searches for a new solution, while keeping the previous solution. Matching Pursuit (MP) [25], Orthogonal Matching Pursuit (OMP), Least Squares Orthogonal Matching Pursuit (LS-OMP), Stagewise Orthogonal Matching Pursuit (StOMP), generalized OMP

(gOMP), and Compressive Sampling Matching Pursuit (CoSaMP) are some examples of greedy algorithms [44].

2) Relaxation method: Relaxation methods are based on convex optimization methods, in which the smallest l_1 or l_2 norm of coefficients among all decompositions are searched, that these algorithms relax the sparsity constraint. The most applicable relaxation algorithms in sparse coding are basis pursuit [28], Least Absolute Shrinkage and Selection Operator (LASSO) [29], Least Angle Regression (LARS) [45], Focal Underdetermined System (FOCUSS) [46], and Iteratively Reweighted Least Squares (IRLS) [47].

Figure 5 summarizes the sparse coding, mentioning some of the popular algorithms.

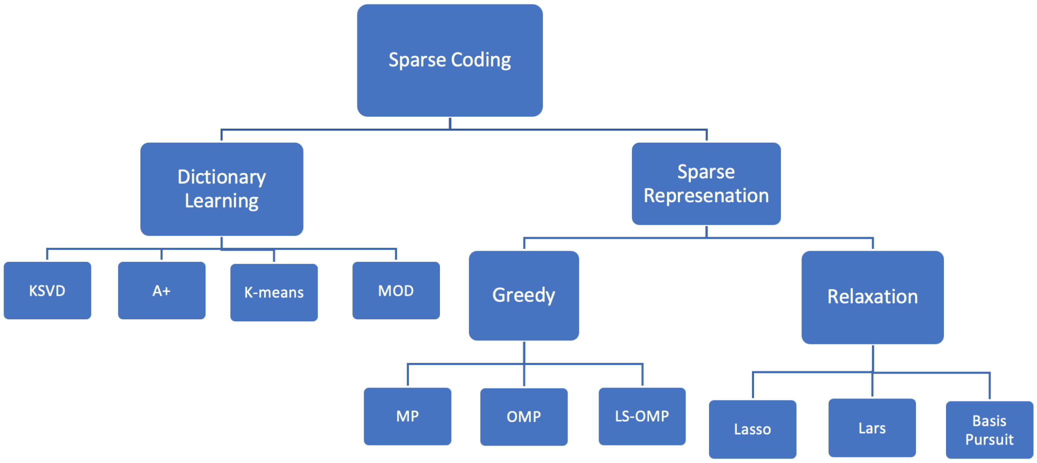

Fig. 5. Sparse coding map

2.1.1 Matching Pursuit

Matching pursuit [25] is a greedy algorithm that decomposes the data by using a redundant dictionary. The notable point about this algorithm is its simplicity and using the residual error in each step for attaining the best atom. In each iteration, MP selects one atom as the initial column of the dictionary $\boldsymbol{d}_i$ and then optimizes the approximation problem in two steps. First, it finds the most correlated column of the dictionary using the inner product of the residual error and $\boldsymbol{d}_i$, and then computes the new column of the dictionary $\boldsymbol{d}_n$ in each iteration n:

$$\boldsymbol{d}_n = arg \max_{\boldsymbol{d}_i} |\langle \boldsymbol{\epsilon}_{n-1}, \boldsymbol{d}_i, \rangle|, 1 \leq i \leq K \tag{9}$$

where n is the number of iterations, and $\boldsymbol{\epsilon}_{n-1}$ is the residual error at $(n-1)$ iteration. The term ”matching” refers to the mentioned correlation. The next step is computing the corresponding weight of the $\boldsymbol{d}_n$ in the sparse representation vector $\boldsymbol{\alpha}_n$. And then update the residual error:

$$\boldsymbol{\alpha}_n = |\langle \boldsymbol{\epsilon}_{n-1}, \boldsymbol{d}_i, \rangle| \tag{10}$$

$$\boldsymbol{\epsilon}_n = \boldsymbol{\epsilon}_{n-1} - \boldsymbol{\alpha}_n \boldsymbol{d}_n \tag{11}$$

Finally, after n iterations, y_n, which is the nth approximation of the data, is:

$$y_n = \sum_{j=1}^{n} \boldsymbol{\alpha}_n \boldsymbol{d}_n \tag{12}$$

Although matching pursuit is a simple approximation method, there are more precise methods to apply such as OMP, LS-OMP and BP, which will be discussed in next sections.

2.1.2 Orthogonal Matching Pursuit

Orthogonal matching pursuit was suggested by Tropp and Gilbert [26] as an improved version of MP so it applies a similar procedure to achieve the representation. OMP is also a greedy method, which approximates the optimal solution of the problem by adding one non-zero value at each iteration. In the first step, the sparse representation is initialized to zero, and the residual error is equal to the input image:

$$\epsilon_0 = \boldsymbol{x} - \boldsymbol{D\alpha} \tag{13}$$

To search for the first atom, the $\boldsymbol{d}_i$ is multiplied with a scaler c, which is obtained by:

$$c_{opt} = \boldsymbol{d}_i^T \epsilon_{n-1} \tag{14}$$

The obtained scalar value is multiplied with the chosen atom $\boldsymbol{d}_i$ in the dictionary and then the residual error is subtracted from the computed value. Using the ℓ_2 norm, we achieve a minimum error value $\boldsymbol{E}_i$

$$\boldsymbol{E}_i = min_c \|c\boldsymbol{d}_i - \epsilon_{n-1}\|_2^2 \tag{15}$$

After comparing $\boldsymbol{E}_i$ values, the new atom with the smallest $\boldsymbol{E}_i$ is chosen and the corresponding coefficient in the sparse representation will be computed:

$$\boldsymbol{\alpha}_n = min_{\boldsymbol{\alpha}} \|\boldsymbol{D\alpha} - \boldsymbol{x}\|_2^2, 1 \leq i \leq K \tag{16}$$

And the representation vector is updated. The significant point about OMP is the orthogonality of the chosen atoms D_s and using the residual error which results in the uniqueness of atom selection. In other words, the selected atom will never be chosen again.

2.1.3 Least Square Orthogonal Matching Pursuit

LS-OMP is another greedy method, which computes and uses the actual error to update the terms, while OMP uses the residual error. The algorithm steps are similar to OMP, but in the first step,$\boldsymbol{E}_i$ is computed as:

$$\boldsymbol{E}_i = min_{\boldsymbol{\alpha}} \|\boldsymbol{D\alpha} - \boldsymbol{x}\|_2^2 \tag{17}$$

As it can be seen, there is no chosen atom, so the error is computed for all the columns of the dictionary to find the atom with minimum error, which makes the LS-OMP slower than the other two mentioned methods (section 2.1.1 and 2.1.2). After comparing the error, the support set and the sparse representation vector are updated, and then to control the stopping criterion the residual error is computed.

As mentioned above the LS-OMP is slower but more precise than MP and OMP. Figure 6 compares these three greedy methods in terms of accuracy and speed.

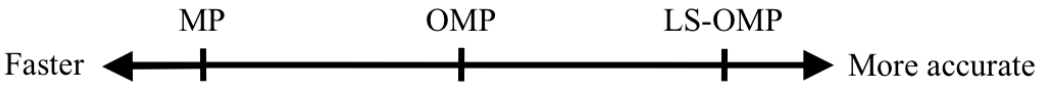

Fig. 6. Pursuit algorithms comparison

2.1.4 *Basis Pursuit*

Basis pursuit is a relaxation-based algorithm which aims to achieve the most precise and sparse approximation of data by relaxing the optimization constraints. So, it becomes a convex optimization problem that can be solved by linear programming (interior point) algorithms. In Equation 2 BP searches for a coefficient with minimum ℓ_1 norm.

$$\min_{\alpha} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \tag{18}$$

Notice that under certain conditions of $\boldsymbol{D}$ and L the ℓ_1 and ℓ_0 norm of $\boldsymbol{\alpha}$ can be equal, and the BP algorithm uses this method to converge to the global optimum point.

2.1.5 *Least Absolute Shrinkage and Selection Operator*

In 1994, Tibshirani [29] proposed a new method known as LASSO which searches for an optimized sparse solution and meanwhile controls the sparsity (Equation 3). LASSO shrinks the residual sum of squares to zero and its performance is more efficient than repetitive greedy algorithms, but the issue with this method is the ℓ_0 norm. To tackle this problem, LASSO transforms the sparse approximation to a convex problem by changing the ℓ_0 to ℓ_1 norm (relaxing the constraints):

$$\min \|\boldsymbol{\alpha}\|_1^2 \quad s.t. \quad \boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha} \tag{19}$$

So, the objective function becomes:

$$arg\min_{\boldsymbol{x}} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 \quad s.t \quad \|\boldsymbol{\alpha}\|_1 \leq L \tag{20}$$

where the term $\|\boldsymbol{\alpha}\|_1 \leq L$ controls the sparsity. There are various methods to solve convex optimization problems such as directional derivatives, but ℓ_1 norm cannot be directly derived. The solution to this problem is deriving along the eigenvector $\boldsymbol{u}$:

$$\nabla f(\boldsymbol{\alpha}, \boldsymbol{u}) = \lim_{t \to 0^+} \frac{f(\boldsymbol{\alpha} + t\boldsymbol{u}) - f(\boldsymbol{\alpha})}{t} \tag{21}$$

Using this equation, the optimized $\boldsymbol{\alpha}$ is found.

2.2 Dictionary Learning

As discussed in the sparse representation in Section 2.1, signals can be modeled as a linear combination of a sparse approximation and a given dictionary (Equation 3). We studied five methods to approximate the sparse representation, and in this section, we will study algorithms to achieve the accurate dictionary. To obtain the dictionary there are mainly two paths: using fixed dictionaries, and applying dictionary learning algorithms.

Fourier bases [48], wavelets [49], discrete cosine bases [50], and contourlet [51] are some of fixed dictionaries which are known as transforms and have been used to solve sparse coding problems. There are another group of fixed dictionaries that can be tunned with respect to input image, such as curvelets [52] , bandlets [53], and wavelet packets [54]. To apply the transform T on input signal x with L non-zero elements, the steps are mentioned below:

1) Apply the chosen transform method and select L coefficients, then make the rest of the elements zero.
2) Obtain the approximated signal $\hat{\boldsymbol{x}}_L$, using the inverse transform
3) Compute the difference between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}_L$, which is a function of L

$$e^2(L) = E\|\boldsymbol{x} - \hat{\boldsymbol{x}}_L\|_2^2 \tag{22}$$

Our goal is to see the drop of the error $e^2(L)$ with maximum speed. To escalate the error drop, researchers have used a mathematical description of the data. They have replaced real images with a piecewise smooth region model, separated with piecewise smooth edges. Using the mathematical description of the data resulted in more accurate transforms such as contourlet [51] and curvelet transforms [52]. However, real images are more complicated than the analyzed signals, so fixed dictionaries lead to unsatisfactory results. To solve this issue, dictionary learning algorithms were implemented. In 1996, Olshausen and Field [41] published a paper which studied simple cells in mammalian visual cortex and showed that the brain applies a sparse algorithm for processing scenes. However, the suggested algorithm in the paper was not sufficiently effective, it proved dictionary learning is more advantageous than PCA. The study of Olshausen and Field was the beginning of many dictionary learning based researches and algorithms, such as MOD [30], K-SVD [32] and A+ [33].

There are various parameters in a dictionary learning including number of non-zero elements L, size of dictionary $K \times N$, and the desired residual error ϵ. The format of the objective function of the problem depends on known parameters. If L is known, the objective function is:

$$\min_{\boldsymbol{D},\boldsymbol{\alpha}_j} \sum_{j=1}^{M} \left\|\boldsymbol{x}_j - \boldsymbol{D}\boldsymbol{\alpha}_j\right\|_F^2 \quad s.t. \quad \forall j, \left\|\boldsymbol{\alpha}_j\right\|_0 \leq L \tag{23}$$

$$\min_{\boldsymbol{D},\boldsymbol{\alpha}_j} \sum_{j=1}^{M} \left\|\boldsymbol{\alpha}_j\right\|_0 \quad s.t. \quad \forall j, \left\|\boldsymbol{x}_j - \boldsymbol{D}\boldsymbol{\alpha}_j\right\|_F^2 \leq \epsilon \tag{24}$$

where M is number input signals. In Equation 23, we aim to minimize the representation error with a fixed number of non-zero elements. Hence, the$\boldsymbol{D}$ and $\boldsymbol{\alpha}$ and their multiplication are undefined, there is a scale ambiguity between them, therefore there is no unique solution for this problem. To fix this issue, the atoms must be forced to have ℓ_0 norm as a constraint of the objective function (Equation 23). In general, if $N \leq K$ the dictionary is overcomplete and the sparse representation is unique, but if $K < N$, it becomes an ℓ_0 minimization problem [32]. In this case, $\boldsymbol{D}$ must have $2L$ columns and every column of it must be linearly independent.

2.2.1 Method of Optimal Directions

In 1999, Engan et al. proposed a simple and fast dictionary learning method know as Method of Optimal Directions (MOD) [30]. This algorithm applies Frobenius norm to compute the distance between the input signal $\boldsymbol{x}$ and approximated signal$\boldsymbol{D}\boldsymbol{\alpha}$.

$$\min_{\boldsymbol{D},\boldsymbol{\alpha}} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_F^2 \quad s.t. \quad \forall j, \|\boldsymbol{\alpha}\|_0 \leq L \tag{25}$$

MOD is performed in three steps:

1) Initialize dictionary:The initialized dictionary can be chosen from the predefined dictionaries, like wavelet, or selecting M random elements from the training data.
2) Fix dictionary and update sparse representation: The sparse representation can be obtained using any matching pursuit algorithm such as MP, OMP or LS-OMP.
3) Update dictionary using the updated sparse representation: When the sparse representation is updated and fixed, we can minimize the quadratic expression at Equation 25 with respect to the $\boldsymbol{D}$, as a least squares problem. For minimization, the

gradient of the quadratic expression must be nulled, which results in the dictionary update formula.

$$\nabla_D \|\boldsymbol{x} - \boldsymbol{D\alpha}\|_F^2 = (\boldsymbol{x} - \boldsymbol{D\alpha})\boldsymbol{\alpha}^T \tag{26}$$

This optimization problem is done by Moore-Penrose pseudo inverse.

MOD is suggested in low-dimensional problems because it converges fast. But it is not efficient in high-dimensional problems due to the difficulty of computing the pseudo-inverse of High-dimension matrices. Hence, MOD is not widely applicable.

2.2.2 K-Means

K-means is an unsupervised iterative algorithm for partitioning data using a pre-defined number of clusters. K-means is applicable in image segmentation, image compression, market segmentation and many other signal processing areas. The algorithm starts with random selection of K cluster centers known as centroids. Then the input samples are grouped with the nearest centroid. The distance $\boldsymbol{d}_j$ of the centroid $\boldsymbol{c}_j$, which is in the cluster j, and the *i*th data point in the same cluster $\boldsymbol{x}_i^j$, is minimized using the following objective function:

$$min \sum_{j=1}^{k} \sum_{i=1}^{m} \left\| \boldsymbol{x}_i^j - \boldsymbol{c}_j \right\|^2 \tag{27}$$

Where *m* is the number of data points in each cluster. K-means clustering is able to solve a specific type of dictionary learning problem. If there is only one non-zero value or atom, this algorithm is helpful and obtains an extremely sparse representation. To optimize Equation 21, after labeling nearest datapoints to the corresponding centroid $\boldsymbol{d}_j$, as mentioned above, the mean of each cluster is computed. Then the initialized centroid is replaced with the new mean of the cluster (Equation 28).

$$\boldsymbol{c}_j = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{x}^j \tag{28}$$

The centroid value keeps being updated until it reaches a stopping criterion, for example:

- the distance $\boldsymbol{d}_j$ is satisfying.
- the cluster stops changing.
- the algorithm has reached the maximum iteration number.

K-means is not a practical method because there is no guarantee to reach the convergence, but it is advantageous in other dictionary learning methods.

2.2.3 K-SVD

In 2006, Aharon et al. introduced a new dictionary learning algorithm, which is based on K-means and singular value decomposition (SVD) [32]. Generally, the method is similar to MOD with the same objective function (Equation 19), but the atoms of the overcomplete dictionary D are updated one by one, and it updates the sparse representation matrix $\boldsymbol{\alpha}$ row by row, by a pursuit algorithm. The update is done in two main steps:

1) Approximating sparse representation: Using the initialized dictionary, the sparse representation matrix can be approximated by any of the pursuit algorithms, discussed in Section 2.1. The paper suggests using orthogonal matching pursuit because of its fast performance and high accuracy.

$$arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_F^2 \quad s.t. \quad \forall j, \|\boldsymbol{\alpha}\|_0 \leq L \tag{29}$$

2) Updating dictionary: In this step, the approximated $\boldsymbol{\alpha}$ is used to update the dictionary (Equation 29). As mentioned above, K-SVD updates the dictionary atoms one by one, by fixing all the atoms except one $\boldsymbol{d}_k$, and then updates it. In Equation

29, the $\boldsymbol{D\alpha}$ is broken into n rank-1 elements, each being one atom multiplied with its corresponding coefficient in the sparse representation matrix $\boldsymbol{\alpha}_k$.

$$\|\boldsymbol{x}-\boldsymbol{D\alpha}\|_F^2 = \left\| \boldsymbol{x} - \sum_{j=1}^{n} \boldsymbol{d}_j \boldsymbol{\alpha}_j^T \right\|_F^2 \tag{30}$$

Assuming that all other atoms are fixed, Equation 30 can also be written like:

$$\left\| \boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{\alpha}_k^T \right\|_F^2 = \left\| (\boldsymbol{x} - \sum_{j \neq k}^{n} \boldsymbol{d}_j \boldsymbol{\alpha}_j^T) - \boldsymbol{d}_k \boldsymbol{\alpha}_k^T \right\|_F^2 \tag{31}$$

where $\boldsymbol{E}_k$ is the residual error while removing the atom k. Then, by applying the SVD decomposition:

$$\boldsymbol{E}_k = \boldsymbol{U} \boldsymbol{A} \boldsymbol{V}^T \tag{32}$$

As we know, the first eigenvector is the largest and most important column of matrix $\boldsymbol{U}$, and we use the value of, $\boldsymbol{u}_1$, as the new $\boldsymbol{d}_k$. After updating $\boldsymbol{d}_k$, the corresponding element in the sparse representation must be updated, too. The updated $\boldsymbol{\alpha}_k$ is:

$$\boldsymbol{\alpha}_k = \boldsymbol{v}_1 \boldsymbol{A}_{1,1} \tag{33}$$

where $\boldsymbol{v}_1$ is the first column of matrix $\boldsymbol{V}$ and $\boldsymbol{A}_{1,1}$ is the first element of matrix $\boldsymbol{A}$. K-SVD is an accurate dictionary learning method, but because of computing the inverse, its speed drops by increasing the dimension. There is a faster method, which will be discussed in the next sections.

2.2.4 Adjusted Anchored Neighborhood Regression

Adjusted anchored neighborhood regression, known as A+, is a method for single image super-resolution proposed by Timofte et al. [55]. In 2017, Aeschbacher et al. modified the A+ algorithm for spectral super resolution based on the K-SVD [33]. In this method, the dictionary and the sparse representation are obtained by K-SVD and OMP.

Then, using the CIE 1964 color matching function, the input data and the overcomplete dictionary are projected to a lower spectral resolution (LSR) and each atom of the dictionary is shown by I_i. The objective function is:

$$arg \min_{\boldsymbol{\alpha}} \|\boldsymbol{x}_L - \boldsymbol{N}_L \boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_2^2 \tag{34}$$

where $\boldsymbol{x}_L$ is the input data projected the LSR and $\boldsymbol{N}_L$ is matrix of I_i nearest neighbors in LSR. The goal of Equation 34 is to optimize the least squares error of the LSR data and the multiplication of the $\boldsymbol{N}_L$ and its corresponding sparse representation. Also, λ is the regularization term for sparsity. The closed-form solution of Equation 34 is:

$$\boldsymbol{\alpha} = (\boldsymbol{N}_L^T \boldsymbol{N}_L + \lambda I)^{-1} \boldsymbol{N}_L^T . \boldsymbol{x}_L \tag{35}$$

Using the obtained $\boldsymbol{\alpha}$, the nearest neighbor's matrix of the higher spectral resolution $\boldsymbol{N}_H$ is computed (Equation 36).

$$\boldsymbol{x}_H = \boldsymbol{N}_H \boldsymbol{\alpha} \tag{36}$$

In order to have a projection from LRS to HSR, we define a projection matrix $\boldsymbol{P}_i$:

$$\boldsymbol{P}_i = \boldsymbol{N}_H . (\boldsymbol{N}_L^T \boldsymbol{N}_L + \lambda I)^{-1} \boldsymbol{N}_L^T \tag{37}$$

$$\boldsymbol{x}_H = \boldsymbol{P}_i \boldsymbol{x}_L \tag{38}$$

Finally, the LSR data can be embedded to the HSR by multiplying with the computed projection matrix. In comparison with the K-SVD, A+ has faster speed and the reconstruction phase is more accurate.

3 CANCER DATASET

In this chapter, we study the hyperspectral colon cancer dataset, which we have used in our research. First, we discuss how the dataset is collected and its numerical and spectral characteristics. Then, we describe the cancer detection method proposed in [24] using the same dataset. Next, we explain the pre-processing and data augmentation methods used for this dataset, and finally, we study the architecture and implementation of the neural network and the detection performance.

3.1 Colon Cancer Dataset

The colon cancer dataset is collected by researchers at the University of South Alabama Medical Center Department of Surgery using Q-Imaging Corporation's Rolera EM-C2 camera with 14-bit digital output [23]. The dataset contains 175 hyperspectral images of tissue samples in two classes: lesional and non-lesional (Fig.7).

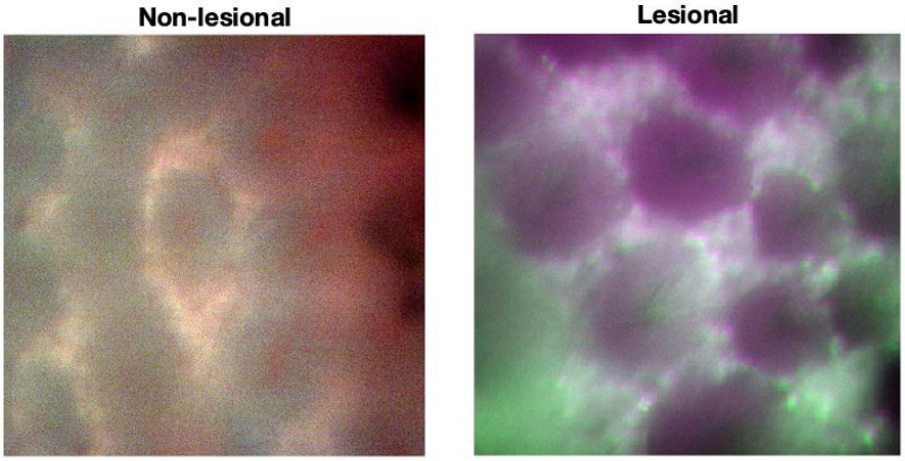

Fig. 7. Hyperspectral image samples of lesional and non-lesional tissues of patient number 8

Each class includes hyperspectral images of 13 patients from at least two different points of view (Table 1). The tissue samples were removed from the patients bodies during surgery (Fig.8). And then, the sample is cut into two parts known as lesional and non-lesional, homogeneously. Using the samples, 88 and 87 images have been taken as lesional and non-lesional, respectively.

Table 1
Number of Tissue Sample HS Images by Patients in Each Class

Patient ID	Number of Lesional Images)	Number of Non-lesional Images	Total
1	3	2	5
2	5	4	9
3	3	5	8
4	8	6	14
5	9	9	18
6	4	6	10
7	15	6	21
8	9	6	15
9	7	8	15
10	8	12	20
12	10	9	19
13	5	6	11
14	2	8	10
Total	88	87	175

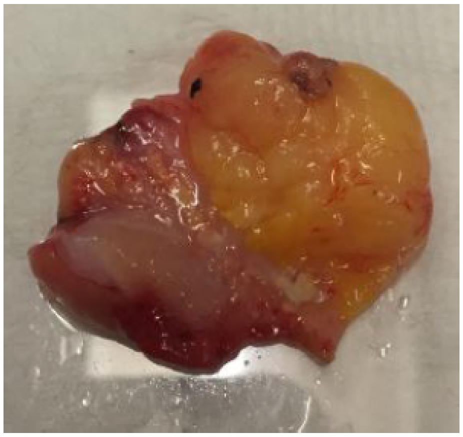

Fig. 8. A tissue sample of the colon cancer dataset

The size of each image is 501×502, and there are 38 layers in the spectral dimension, which results in 9,557,076 pixels, and each pixel is 8 μm × 8 μm. The spectral intensity of each pixel is between 0 to 16383, and it is unitless. The spectral range is 190 nm, from 360 to 550 nm, and the spectral resolution is 5 nm. As shown in Figure 9 and Figure 10, the spectral distributions in lesional and non-lesional images are different. The spectral range of the blue, green, and red lights are shown with the dotted lines.

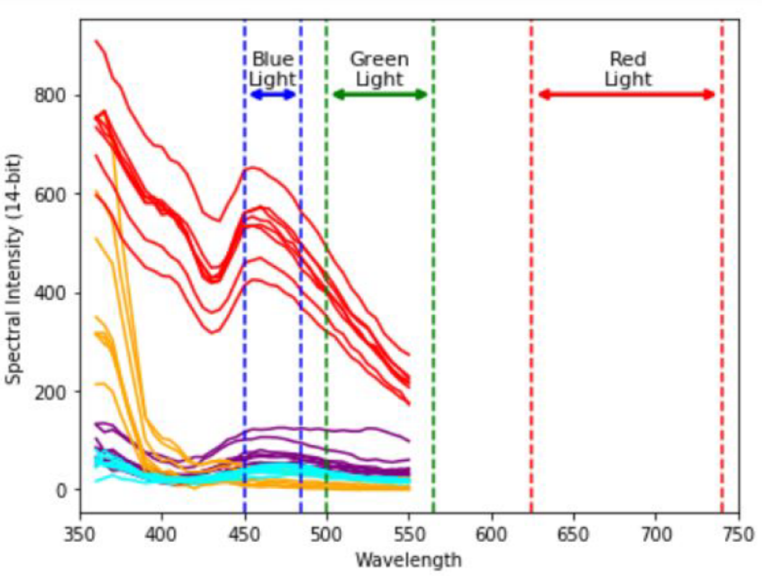

Fig. 9. Spectral distribution of lesional images

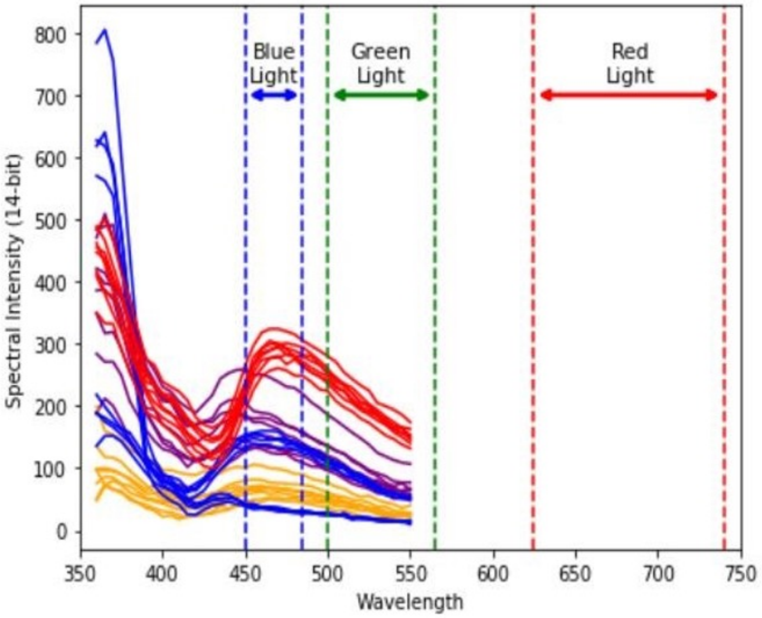

Fig. 10. Spectral distribution of non-lesional images

3.2 Pre-processing Data

Mobilia et al. have developed three pre-processing schemes before applying CNNs as classifiers on the colon cancer hyperspectral image dataset [24]. The schemes are:

1) Panchromatic (PC): All 38 spectral layers are merged into one single channel, so there are 175 grayscale images.
2) Individual Band (IB): The hyperspectral bands are divided and treated as an individual image. There are 175 images with 38 layers which results in 6650 grayscale images for training and testing.
3) Hypercube (Hyper): Original hyperspectral images with all 38 layers are used.

As mentioned in Section 1, high accuracy in training neural networks needs a large dataset. Considering the small size of our dataset for training the CNN, images have been divided into smaller segments as a data augmentation method. Based on the homogeneity of images, the Hyper and PC images have been segmented. They have been presented in two other styles, one with 100×100 and the other one with 50×50 pixel images. The 100×100 segmented Hyper and PC images have been named as segmented hypercubes (SH) and segmented panchromatic (SPC), respectively, and in total adding up to 4375 images. The smaller segmented hypercubes (SSH) and smaller segmented panchromatic are the Hyper and PC images with the size of 50×50, which in total are 17500 samples. Table 2 is a summary of different schemes and corresponding features or pixels.

Table 2
Summary of Schemes and their Corresponding Features in Mobilia et. al

Scheme	Total Number of Images	Image Dimensions	Number of Features
IB	6650	501 × 502 × 1	251502
PC	175	501 × 502 × 1	251502
Hyper	175	501 × 502 × 38	9557076
SPC	4375	100 × 100 × 1	10000
SH	4375	100 × 100 × 38	380000
SSPC	17500	50 × 50 × 1	2500
SSH	17500	50 × 50 × 38	95000

After pre-processing, the data must be processed further to be proper as an input to the CNN model. The required pre-processing actions for this CNN are normalizing, flattening, and storing in hdf5 files. To normalize the dataset, the pixels are divided by 16383, which is the maximum output value of the camera. The next step is flattening and then storing images (or image segments) and corresponding labels (0 for lesional and 1 for non-lesinonal). Mobilia et. al has used shuffling and one-hot-encoding method for the validation set [24]. As mentioned, there are 13 patients, and the number of images for each patient is known. The network is trained by images of 12 patients, and images of one patient are used as test data.

3.3 CNN Architecture and Implementation

The convolutional neural network is made by three 2-dimensional convolution layers with tanh as the activation function and the dropout regularization method. The convolutional layers apply zero padding on the inputs. After each convolution layer, there is a max-pooling layer which last one is followed by a fully connected layer. Table 3 shows a summary of the CNN architecture.

Table 3
CNN Architecture

CNN Layer	Kernel Window Size	Stride	Feature Maps
Conv+tanh+Drop	5 X 5 X d	3	24
Max Pool	2 X 2 X d	2	24
Conv+tanh+Drop	3 X 3 X d	1	48
Max Pool	2 X 2 X d	2	48
Conv+tanh+Drop	2 X 2 X d	1	64
Max Pool	2 X 2 X d	2	64
FC Layer	24	-	-

The network is implemented in Python using TensorFlow and the model uses the Leave One Patient Out (LOPO) method for validation. In other words, images of 12 patients are used for training. Each scheme has a different number of epochs, which is because of the convergence of the model with different datasets. Table 4 provides the test performance of the classification model with different pre-processing methods. As can be seen, smaller segmented hypercubes dataset has the best performance in comparison with other schemes.

Table 4
Test Performance of Mobilia et al.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Pre-processing**
IB	0.527	0.615	0.463	0.520	divided
PC	0.529	0.612	0.516	0.557	Merged
SPC	0.530	0.740	0.289	0.414	Segmented
SSPC	0.542	0.815	0.269	0.403	Segmented
Hyper	0.564	0.748	0.416	0.498	Original
SH	0.674	0.771	0.629	0.690	Segmented
SSH	0.741	0.853	0.666	0.747	Segmented

4 PROPOSED METHOD AND PERFORMANCE COMPARISON

In this chapter, we review the limited data size problem of medical hyperspectral datasets and discuss the methods we applied to solve this issue. Then, we provide results of our methods. Finally, we compare the result of our data augmentation method with other methods for classification improvement via data augmentation.

Hyperspectral image datasets provide valuable information about the target object, but gathering them is expensive and complicated, especially in medical cases. We know that training neural networks requires large datasets to achieve high accuracy. In this research, we augment the colon cancer HSI dataset via sparse coding, and then we feed the new dataset to the CNN model (Table 3) to classify the images as lesional and non-lesional.

4.1 Data Augmentation via Sparse Coding

To increase the number of images in the dataset, image reconstruction can be applied. There are numerous reconstruction methods such as using adversarial networks [56]. We have applied K-SVD, A+ with no patching and A+ with 3×3 patching for augmenting colon cancer hyperspectral images.

The data reconstruction is done in MATLAB and the reconstruction error was evaluated by the root-mean-square-error (RMSE) and relative RMSE (rRMSE). The RMSE and rRMSE are computed by:

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(l_R^i - l_O^I)^2} \tag{39}$$

$$rRMSE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{(l_R^i - l_O^I)^2}{l_O^i}} \tag{40}$$

where l_O^i and l_R^i are the ith element of the original and reconstructed image, respectively, and n is the number of pixels.

4.1.1 K-SVD

In this section, we use K-SVD method (Section 2.2.3) to reconstruct the hyperspectral images [35]. The dictionary size is set to 64 with support $L = 4$. In this study, the dictionary size, and the number of non-zero values in the sparse representation L are hyperparameters that we need to optimize. Hence, we have used grid search for optimizing our hyperparameters. We defined a set of numbers for each one of them. The set for L contains 4, 5, 6 and 7, and the set for dictionary size contains 32, 64, 128, and 256. After comparing RMSE and rRMSE of these different dictionaries, we chose the one with lowest error. The RMSE and rRMSE of reconstructing lesional hyperspectral images with K-SVD method are $7.89x10^{-1}$ and $9.13x10^{-2}$, respectively. And, for non-lesional images these numbers are $7.48x10^{-1}$ and $9.9x10^{-2}$. Reconstructing each image takes 194 seconds in average. The reconstructed and original hyperspectral images are shown in Figure 11 and the difference between these two images is shown in Figure 12.

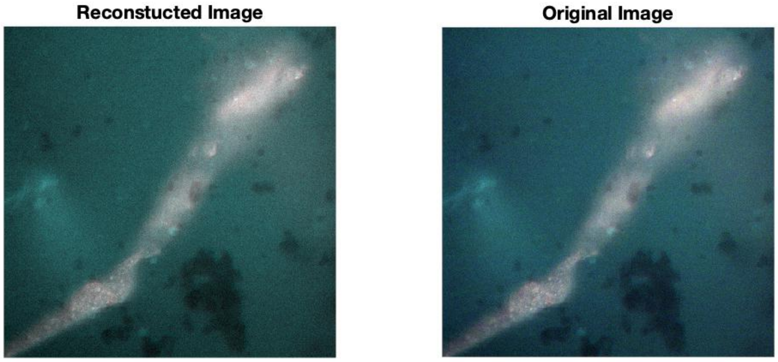

Fig. 11. The original and reconstructed image of lesional tissue via K-SVD

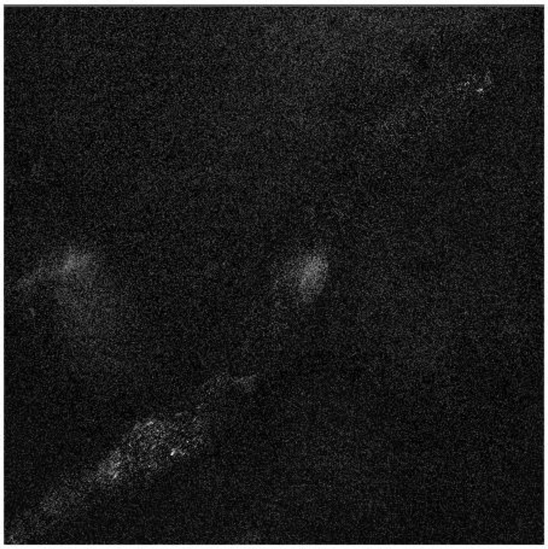

Fig. 12. The difference between original and the reconstructed image lesional tissue via K-SVD

4.1.2 A+ without patching

In general, the main goal of adjusted anchored neighborhood regression (A+) method is improving single image super resolution [55], but Aeschbacher et al. [33] have changed the algorithm for image recovery and reconstruction. The modified A+ has been proposed for reconstructing hyperspectral images with the visible spectrum, which is 400 nm to 700 nm. However, the spectrum of the colon cancer dataset ranges from 360 nm to 550nm, so we modify the color matching function using CIE 1964.

The sparsity regularization parameter λ is set to 0.1 and dictionary size is 512. As discussed in last section (4.1.1), λ and dictionary size in A+ methods are also hyperparameters and they must be optimized. Hence, we apply the same method with a set of numbers for each hyperparameter. The set for λ contains 0.01, 0.1, 1 and 10, and the set for dictionary size in A+ contains 128, 256, 512, 1024. Notice that increasing the size of dictionary requires more computing power and time.

Based on the discussed algorithms in section 2.2.4, we will attain two dictionaries: Lower Spectral Resolution (LSR) and Higher Spectral Resolution (HSR). In our case, we are working with RGB (3 spectral layers) and hyperspectral images with 38 spectral layers, so we set the LSR and HSR to 3 and 38, respectively. A+ is six times faster than K-SVD method and in average, each image reconstruction takes 33 seconds.

The RMSE and rRMSE of reconstructing lesional hyperspectral images (calculated by Equation 39 and 40) is $4.98x10^{-1}$ and $5.76x10^{-2}$, respectively. Also, the RMSE and rRMSE of non-lesional hyperspectral images are $4.77x10^{-1}$ and $5.67x10^{-2}$, respectively. There is a positive correlation between the dictionary size and rRMSE. By increasing the dictionary size, rRMSE drops, but there is a saturation point [33]. As mentioned above, we try different numbers for the dictionary size. We start by 128, increase it to 256, 512 and finally 1024. The smallest reconstruction error drops by increasing the dictionary size from 64 to 512, and then for 1024 the error increases, so we stop enlarging the dictionary size.

Figure 13 shows the reconstructed and original colon cancer hyperspectral images of patient number 8 in second field of view, and as can be seen the reconstructed image and spectral plot contains bluer spectrum than the original image, which means the A+ without patching method has used lower spectrum for reconstruction. Each band in the spectral plot presents one color spectrum (RGB). The difference between images is shown in Figure 14.

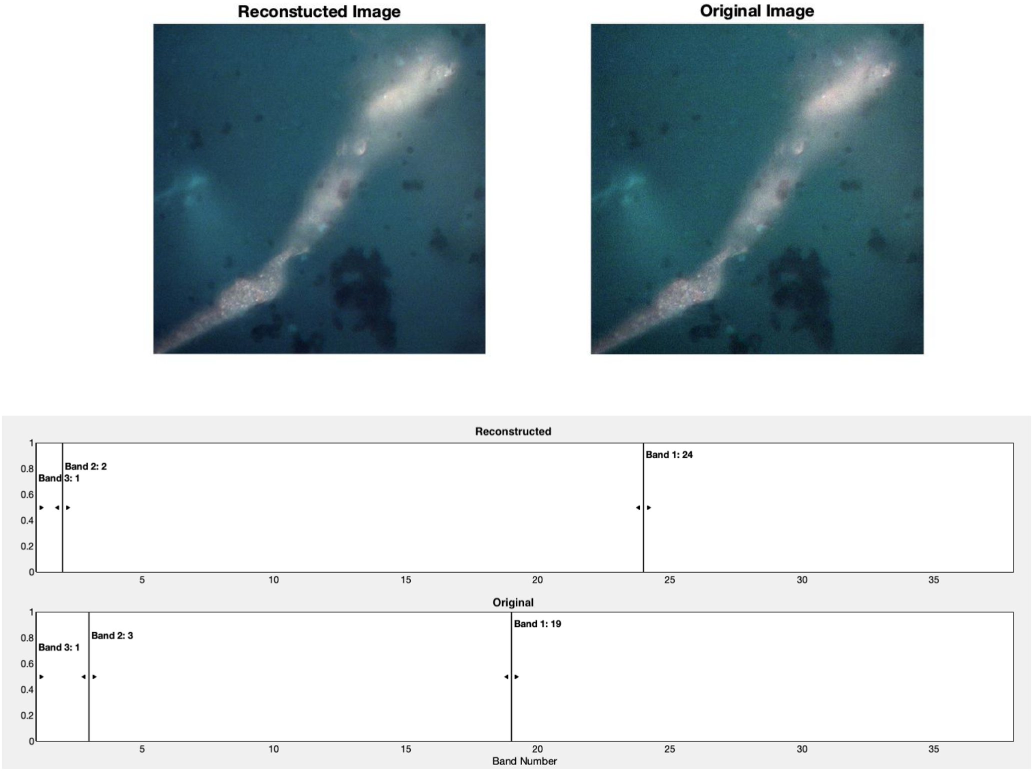

Fig. 13. The original and reconstructed lesional image using A+ and their spectral plot

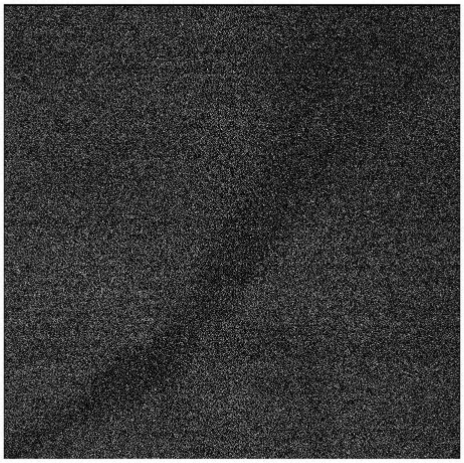

Fig. 14. The difference between original and the reconstructed lesional image using A+. The reconstruction error is for this images is 4.21×10^{-1}

4.1.3 A+ with 3x3 Patching

A+ with 3×3 patching applies the same algorithm as A+ without patching, but before using the images for selecting dictionary atoms, the images are patched 3×3 which results in overlapping patches with 9 pixels. As discussed in section 2.2.4, the A+ algorithm uses a matrix of nearest neighbors of a pixel for atom selection and in this case the nearest neighbor contains 9 closest pixels in the image. The difference is in the pre-processing section, which helps to remove noise of the image by averaging pixel values. The result of patching can be seen at the decrease of the error in comparison of previous methods.

Table 5 and 6 shows the RMSE and rRMSE of discussed algorithms. A+ with patching size 3 has the lowest error. Although, increasing the number of patches, increases the calculation and consequently the reconstruction time. In this study, we tried to increase the number of patches, but MATLAB has 10000 matrix array limit and cannot process A+ with patch sizes larger than 3.

Table 5
The RMSE and rRMSE of reconstruction methods for lesional set

Reconstruction Method	**RMSE**	**rRMSE**
K-SVD	7.89×10^{-1}	9.13×10^{-2}
A+	4.98×10^{-1}	$5.76 \times 10^{-}2$
A+ (Patch Size = 3)	3.56×10^{-1}	4.17×10^{-2}

Table 6
The RMSE and rRMSE of reconstruction methods for non-lesional set

Reconstruction Method	**RMSE**	**rRMSE**
K-SVD	$7.48 \times 10^{-}1$	$9.9 \times 10^{-}2$
A+	$4.77 \times 10^{-}1$	$5.69 \times 10^{-}2$
A+ (Patch Size = 3)	$3.39 \times 10^{-}1$	$4.07 \times 10^{-}2$

In Table 5 and 6 K-SVD has the highest and A+ with patch size 3 has the lowest RMSE and rRMSE. It is notable that one of the most important factors in reconstruction error is the luminance. The pixels with lower luminance cause higher RMSE, so we compute rRMSE which is not biased based on the luminance. The original and the A+ (3x3) reconstructed image are shown in Figure 15, and the difference between these two images can be seen in Figure 16.

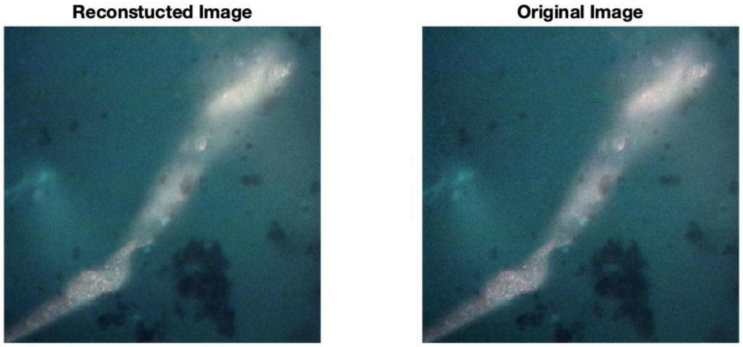

Fig. 15. The original and reconstructed image via A+ (3x3)

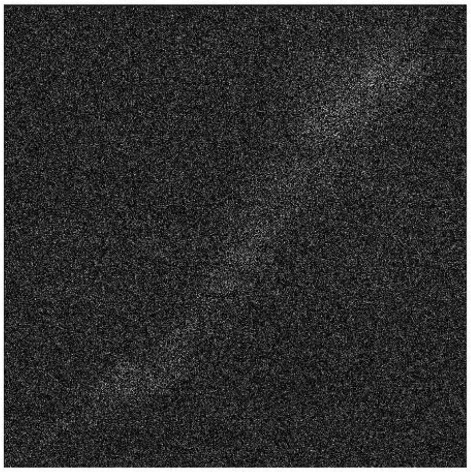

Fig. 16. The difference between original and the reconstructed image via A+ (3x3)

4.2 Image Classification by CNN

After reconstructing the colon cancer hyperspectral images, we use the original and reconstructed images to improve the performance of the CNN model, as discussed in section 3.1. Images of patients 3, 6, 9, and 12 have been separated as the test set, so we have not used them in the reconstruction part. We have reconstructed 123 images with three reconstruction methods which results in 297 images, and 246 of these images are used in the training set. Notice that the test images must not be augmented, and we have used the original images of the mentioned patients for testing, which in total resulted in 52 test images. To compare the effects of reconstruction methods, we have trained the model with four different datasets:

- OKSVD: Original with images generated using K-SVD
- OA+: Original with images generated using A+
- OA+3by3: Original with images generated using A+ and patching of size 3 patching
- OMix: Original with images generated using

As mentioned in section 3.2, the data can be presented in three ways: Panchromatic, Individual Band, and Hypercube. In this study, we have used the simple Hypercube images, without any image segmentation. The datasets have been normalized and flattened to be proper for feeding to the convolutional neural network. We have used the CNN architecture discussed in the previous Chapter (Table3).

4.2.1 Classification Results

To study the result of increasing number of images in the dataset, we have increased number of reconstructed images gradually. In each dataset, we have trained the model with 123 (64 lesional + 59 nonlesional), 147(76 lesional + 71 nonlesional), 171(88 lesional + 83 nonlesional), 195(100 lesional + 95 nonlesional), 219(112 lesional + 107 nonlesional), and 246 (128 lesional + 118 nonlesional) training images. The OMix dataset contains reconstructed images of all three reconstruction methods, and we have more than

246 training images. Hence, we increased the number of images in the OMix dataset to 266 training images. This section has been implemented in Google Colab Pro with 25 GB RAM. Unfortunately, because of lack of computing RAM, we could not use more data for training. Following figures show the result of increasing training data with different dictionary learning methods in testing and training accuracy of the colon cancer classification problem.

We started with OKSVD dataset. The classification test accuracy with the 123 original training images was 56.4%. In each step, 24 reconstructed images have been added to the training set and number of iterations in each epoch is proper to the number of training images. Notice that in the last step 27 images have been added. The reconstructed images have some residual error, which makes them different from the original image. Hence, adding reconstructed images increases the diversity of the training set and results in more robust convolutional neural network. As anticipated, by increasing the number of training images, the test accuracy increases, and it is to a value of 86.53%. Also, the training accuracy raises from 75.24% to 90.79%. By looking at the last steps of Figure 17, there appears to be a sharp upward pattern in the test accuracy. To follow this pattern, there was no more K-SVD reconstructed images to add to the training set for classification, so we have used the OMix dataset, which will be discussed further.

$$Precision = \frac{True\,Positive}{True\,Positive + False\,Positive} \tag{41}$$

$$Recall = \frac{True\,Positive}{True\,Positive + False\,Negative} \tag{42}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{43}$$

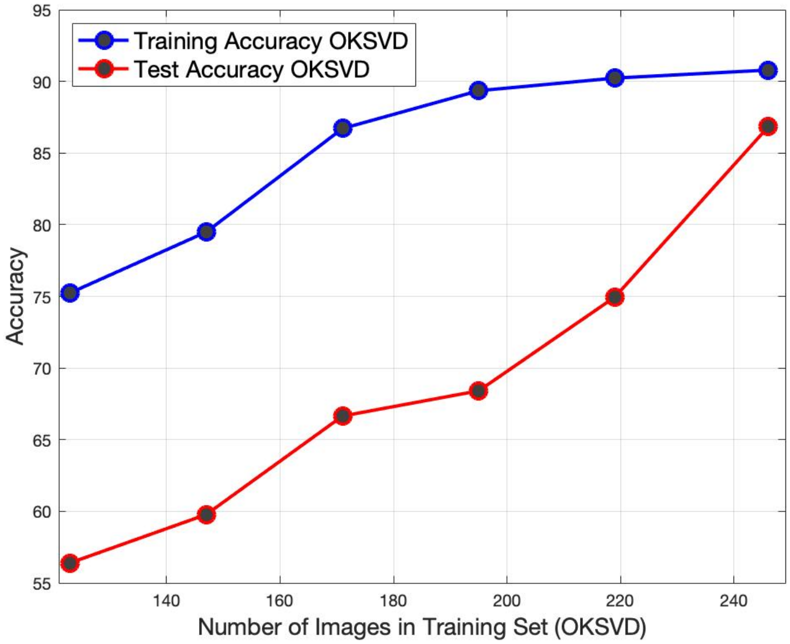

Fig. 17. The classification accuracy of the OKSVD dataset

The data augmentation technique -using the OKSVD training set, also increases precision, recall, and F1-score by 23.09%, 28.44%, and 27.47%, respectively. In other words, there is a decrease in false classification of both cancerous and non-cancerous tissues. The positive and negative term in Equations 41, 42 and 43, means the lesional and non-lesional images, hence, true positive refers to the lesional image which has been correctly classified.

Figure 18 demonstrates the result of data augmentation by A+ method without patching. There is a consistent rise in the test accuracy, while the growth of training accuracy in last three steps becomes slower. The highest achieved test accuracy is 82.67% which is 26.27% higher than the original image set, and training accuracy reaches to 91.28%. As it is shown in Figure 18, the difference between the test and training accuracy is dropping as the number of images in the training set increases. The precision and recall

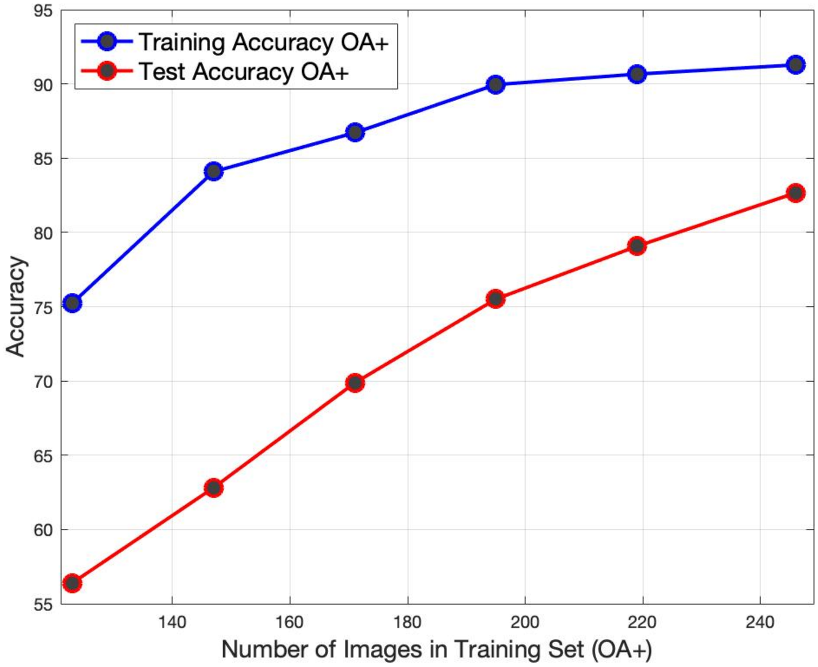

Fig. 18. The classification accuracy of the Original + A+ dataset

of this training set has risen to 94.91% and 68.29%. The F1-score has climbed from 49.8% to 76.07%.

The test and training accuracy of the OA+3by3 dataset are shown in Figure 19. There is a steady rise in the test accuracy from 123 to 195 training images, then the trend becomes slower and saturates around at 80.76%. The training accuracy has surged from 75.24% toward 91.25%, and the difference between test and training accuracy has decreased from 18.84% to 10.94%, which shows augmenting the data has helped solving the overfitting problem.

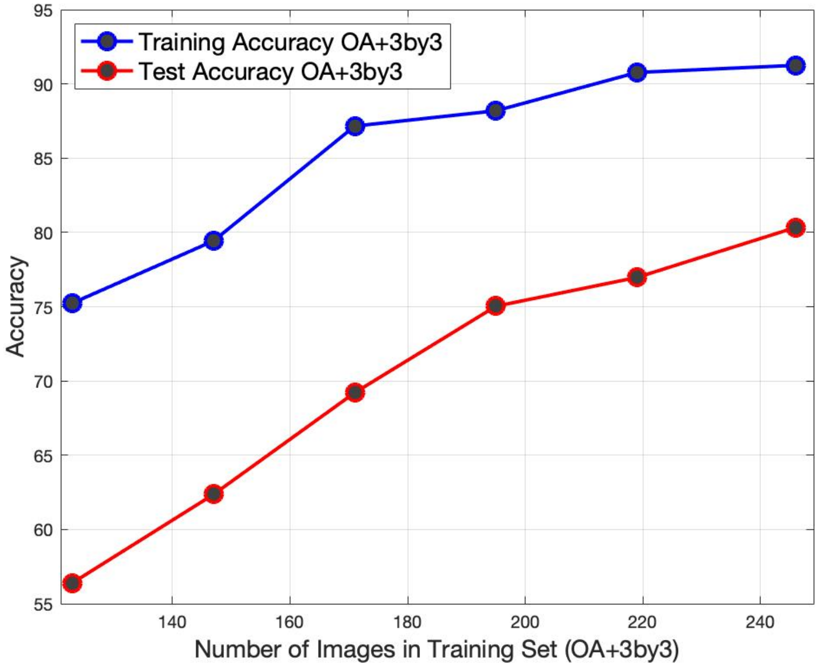

Fig. 19. The classification accuracy of the Original + A+ (3x3) dataset

Improving the CNN performance by sparse coding is proven. In every three cases we tried the maximum number of reconstructed images, but the figures are still moving upward. So, we made a mixed dataset of the three reconstruction methods to add more data for training. . In each step we added 24 images, except the last two steps, that we added 27 and 20 images, respectively. The images were added randomly and fairly, in other words, from each reconstruction method (K-SVD, A+ without patching, A+ with 3×3 patching) we have randomly selected 8 reconstructed images, which results in 24 images. Then these 24 images are added to the training set in each step.

By reaching to 266 training images, we reached to maximum RAM in Google Colab and could not add more training data. Figure 20 demonstrates the accuracy improvement by adding more data. In previous figures, the maximum number of training images was 246, but in the OMix dataset, this number is increased to 266. Also, the maximum accuracy with 246 images is 86.53% but then with 266 images ascents to 94.23%. Also, the precision rises to 1 which means there is no non-lesional image predicted as lesional. The recall and F1-score have reached 87.5% and 93.33%. Table 7 demonstrates the confusion matrix of OMix dataset.

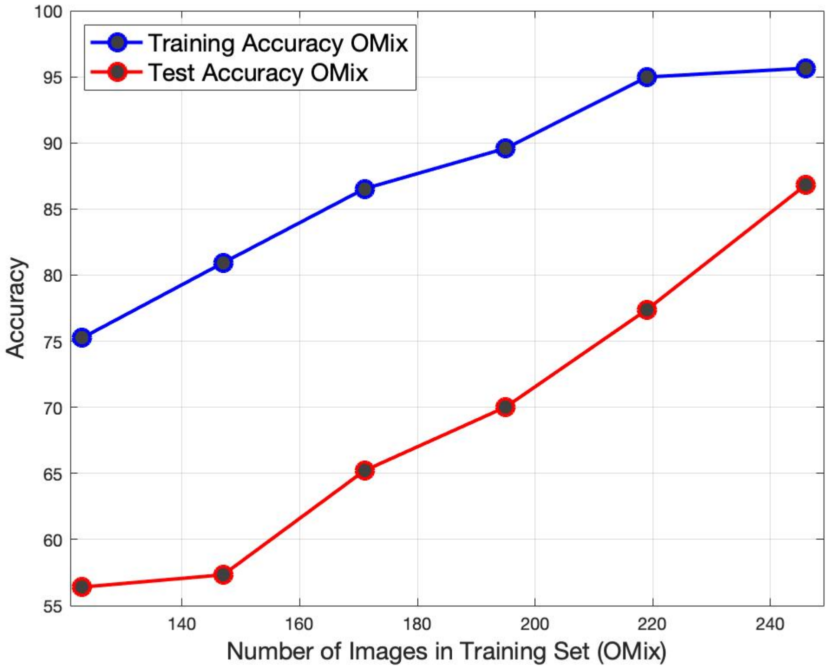

Fig. 20. The classification accuracy of the Original + Mix dataset

Table 7
The confusion matrix of OMix dataset (Test Set)

	Predicted Positive (Lesional)	**Predicted Negative** (Non-lesional)
Actual Positive	21	3
Actual Negative	0	28

4.3 Performance Comparison

In this section, performances of different augmentation techniques are compared. In [24], authors apply image segmentation method to improve the accuracy of the classifier, and its results are summarized in Table 4. The SSH scheme has the best performance by 74.1% accuracy, which is less than the performance obtained using any of the reconstruction methods. Furthermore, precision, recall, and F1-score of methods in [24] are also less in comparison with our methods. We have also applied Simple Data Augmentation (SDA) techniques such as rotating, flipping and cropping, but it did not a have significant impact. Performance of the model trained by SDA is weaker than SH, SSH, and any model trained by reconstructed data (the classification metric values are presented in Table 8 and 9). Table 9 contains the results of original test data which does not include any reconstructed or generated image.

Table 8
Accuracy of original and reconstructed datasets (%)

Metric	**Original**	**OKSVD**	**OA+**	**OA+3by3**	**OMix (266)**	**SDA**
Accuracy	75.24	90.79	91.28	91.25	97.65	83.71

Table 9
Classification metrics of original and augmented datasets (%)

Training Set	**Test Accuracy**	**Test Precision**	**Test Recall**	**Test F1-score**
Original	56.4	74.8	41.6	49.8
OKSVD	86.53	97.89	70.04	77.27
OA+	82.67	94.91	68.29	76.07
OA+3by3	80.76	92.14	67.81	75.36
OMix	85.69	96.83	70.91	76.5
SDA	63.71	80.83	61.4	65.98
IB	52.7	61.5	46.3	52
PC	52.9	61.2	51.6	55.7
SPC	53	74	28.9	41.4
SSPC	54.2	81.5	26.9	40.3
SH	67.4	77.1	62.9	69
SSH	74.1	85.3	66.66	74.7

As seen in Table 9, the OKSVD outperforms each of other augmented datasets with 86.53% accuracy, 97.89% precision and 77.27% F1-score, and the OMix dataset stands in the second rank. In Table 5 and 6, we show the RMSE and rRMSE of the three reconstruction methods. OKSVD has the highest error in lesional and non-lesional reconstruction, which means the augmented data is more different, in comparison with other two methods. We believe the difference between the original and reconstructed images helps the model to be more robust, which results in lower classification error. Although, the reconstruction error increases the robustness of the CNN model, there is a limit for this performance gain. If the error exceeds, the reconstructed spectral layers will not be similar enough to the original set. Hence, there is a trade-off between the reconstruction error and the classification accuracy.

As shown in Figure 20, the accuracy of the OMix dataset with 266 training images reaches 94.23%, which is the highest achieved value between all the mentioned methods. Also, precision, recall and F1-score of this set are higher than previous methods. Considering the reconstruction time and number of augmented data, we suggest using the OMix dataset for the colon cancer hyperspectral image classification problem.

5 CONCLUSION AND FUTURE WORK

This thesis aimed to improve the classification of the hyperspectral colon cancer dataset by applying sparse coding as a data augmentation method. To classify this dataset, we have applied Convolutional Neural Network (CNN), but the small size of the training set caused overfitting and low accuracy in this classification problem. In most cases, medical datasets contain a small amount of data, and this problem becomes more challenging in hyperspectral datasets because obtaining these images is expensive and complicated. To tackle this issue, the dataset was reconstructed by three dictionary learning methods and then added to the original training set. We have used K-SVD, A+ without patching, and A+ with 3×3 patching. The new training sets were fed to the CNN model to classify the lesional and non-lesional hyperspectral images. Using the reconstruction methods, number of images in the training set were doubled in each new dataset.

Table 10 compares the classification metrics for the proposed schemes. OKSVD has the best performance in augmented datasets with 246 training images. Considering the reconstruction RMSE and rRMSE, higher reconstruction error causes more diverse training sets, which results in robustness of the CNN.

Table 10
Performance of classifier model with original and reconstructed datasets (%)

Training Set	Accuracy	Precision	Recall	F1-score
Original	56.4	74.8	41.6	49.8
OKSVD	86.53	97.89	70.04	77.27
OA+	82.67	94.91	68.29	76.07
OA+3by3	80.76	92.14	67.81	75.36
OMix (246 images)	85.69	96.83	70.91	76.5
OMix (266 images)	94.23	100	87.5	93.33

As shown in Table 10, OKSVD method had the highest and OA+3by3 has the lowest RMSE, and the effect of this error on training classification model is notable in Table 10. Although, high RMSE might help the CNN performance, there is a balance between reconstruction error and classification accuracy. If the RMSE exceeds a threshold, the images are not precisely reconstructed, and they are not advantageous for training the CNN model.

To study the increasing trend of test accuracy, a new dataset was provided, which is a combination of all three reconstruction methods, known as OMix dataset. Maximum number of images in other datasets was 246, but we have added 20 more reconstructed images to this dataset. Using the new dataset, the test accuracy has reached to 94.23%, and also the difference between the test and train accuracy has dropped from 18.84% to 3.42%, which proves an improvement in the overfitting problem. We suggest using OMix dataset, because the number of images can be two times more than OKSVD dataset, and also reconstruction time is improves since reconstructing images with A+ methods is six times faster than K-SVD method.

In the proposed work, we have applied convolutional neural network for classifying the hyperspectral images. For future work, we suggest using transfer learning method for image classification. Pretrained neural networks are popular in cases we are facing small and more complicated datasets. Using Resnet50, VGG-16, and EfficientNet would facilitate image classification task. In [24], author has utilized transfer learning but showed that the performance does not improve without data augmentation. We believe if transfer learning is combined with OMix data augmentation, the performance can be further improved.

Another suggestion for future work is using pairing samples as a data augmentation method. In this technique, two images are randomly selected and their average for each pixel is computed. The average image is added to training set, and it helps the classification model to achieve higher accuracy.

To conclude, we have applied three sparse coding methods and made a mixed training set of the original and reconstructed images to improve the CNN model performance. The experimental results show that augmenting hyperspectral colon cancer images increases the robustness of classification model and helps to tackle the overfitting issue.

References

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Processing Mag*, vol. 19, no. 1, pp. 17–28, 2002.

[2] P. G. et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 37–78, 2017.

[3] L. M. D. et al., "Hyperspectral imaging applications in agriculture and agro-food product quality and safety control: A review," *Applied Spectroscopy Reviews*, vol. 48, no. 2, pp. 142–159, 2013.

[4] R. L. Schuler, P. E. Kish, and C. A. Plese, "Preliminary observations on the ability of hyperspectral imaging to provide detection and visualization of bloodstain patterns on black fabrics," *Journal of Forensic Sciences*, vol. 57, no. 6, pp. 1562–1569, 2012.

[5] E. B. et al., "Mapping salt-marsh vegetation by multispectral and hyperspectral remote sensing," *Remote sensing of environment*, vol. 105, no. 1, pp. 54–67, 2006.

[6] J. Freeman, F. Downs, L. Marcucci, E. Lewis, B. Blume, and J. Rish, "Multispectral and hyperspectral imaging: applications for medical and surgical diagnostics," in *Proc. 19th Annu. Int. Conf.IEEE Eng. Med. Biol. Soc. 'Magnificent Milestones and Emerging Opportunities in Medical Engineering' (Cat. No.97CH36136)*, vol. 2, pp. 700–701, 1997.

[7] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.

[8] G. Lu and B. Fei, "Medical hyperspectral imaging: a review," *Journal of Biomedical Optics*, vol. 19, no. 1, pp. 1 – 24, 2014.

[9] B. J. Sumpio, G. Citoni, J. A. Chin, and B. E. Sumpio, "Use of hyperspectral imaging to assess endothelial dysfunction in peripheral arterial disease," *Journal of Vascular Surgery*, vol. 64, no. 4, pp. 1066–1073, 2016.

[10] R. G. et al., "Systemic effects of shock and resuscitation monitored by visible hyperspectral imaging," *Journal of biomedical optics*, vol. 5, no. 5, pp. 847–55, 2003.

[11] D. Cohen, M. Arnoldussen, G. Bearman, and W. Grundfest, "The use of spectral imaging for the diagnosis of retinal disease," in *LEOS'99. 12th Annual Meeting. IEEE Lasers and Electro-Optics Society 1999 Annual Meeting (Cat. No. 99CH37009)*, vol. 1, pp. 220–221, 1999.

[12] J. V. Frangioni, "New technologies for human cancer imaging," *Journal of Clinical Oncology : Official journal of the American Society of Clinical Oncology*, vol. 26, no. 24, pp. 4012–21, 2008.

[13] M. Halicek, H. Fabelo, S. Ortega, G. M. Callico, and B. Fei, "In-vivo and ex-vivo tissue analysis through hyperspectral imaging techniques: Revealing the invisible features of cancer," *Cancers*, vol. 11, no. 6, p. 756, 2019.

[14] M. A. Calin, S. V. Parasca, D. Savastru, and D. Manea, "Hyperspectral imaging in the medical field: Present and future," *Applied Spectroscopy Reviews*, vol. 49, no. 6, pp. 435–447, 2014.

[15] S. G. Kong, M. E. Martin, and T. Vo-Dinh, "Hyperspectral fluorescence imaging for mouse skin tumor detection," *ETRI Journal*, vol. 28, no. 6, pp. 770–776, 2006.

[16] C. E. DeSantis, J. Ma, M. M. Gaudet, L. A. Newman, K. D. Miller, A. Goding Sauer, A. Jemal, and R. L. Siegel, "Breast cancer statistics, 2019," *CA: A Cancer Journal for Clinicians*, vol. 69, no. 6, pp. 438–451, 2019.

[17] S. V. Panasyuk, S. Yang, D. V. Faller, D. Ngo, R. A. Lew, J. E. Freeman, and A. E. Rogers, "Medical hyperspectral imaging to facilitate residual tumor identification during surgery," *Cancer Biology & Therapy*, vol. 6, no. 3, pp. 439–446, 2007. PMID: 17374984.

[18] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. 71, no. 3, pp. 209–249, 2021.

[19] H. e. a. Akbari, "Hyperspectral imaging and quantitative analysis for prostate cancer detection.," *Journal of biomedical optics*, vol. 17, no. 7, p. 076005, 2012.

[20] R. Vega, P. Gorji, Z. Zhang, X. Qin, A. Hareendranathan, J. Kapur, J. Jaremko, and R. Greiner, "Sample efficient learning of image-based diagnostic classifiers using probabilistic labels," *ArXiv*, vol. abs/2102.06164, 2021.

[21] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “Synthetic data augmentation using gan for improved liver lesion classification,” in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 289–293, 2018.

[22] J. Nalepa, M. Myller, and M. Kawulok, “Training- and test-time data augmentation for hyperspectral image segmentation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 292–296, 2020.

[23] J. Deal, S. Mayes, C. Browning, S. Hill, P. Rider, C. Boudreaux, T. C. Rich, and S. J. Leavesley, “Identifying molecular contributors to autofluorescence of neoplastic and normal colon sections using excitation-scanning hyperspectral imaging,” *Journal of Biomedical Optics*, vol. 24, no. 2, pp. 1 – 11, 2018.

[24] S. Mobilia, B. Sirkeci-Mergen, J. Deal, T. C. Rich, and S. J. Leavesley, “Classification of hyperspectral colon cancer images using convolutional neural networks,” in *2019 IEEE Data Science Workshop (DSW)*, pp. 232–236, 2019.

[25] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[26] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[27] S. CHEN, S. A. BILLINGS, and W. LUO, “Orthogonal least squares methods and their application to non-linear system identification,” *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[28] S. Chen and D. Donoho., “Basis pursuit,” *Stanford University*, 1994.

[29] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[30] K. Engan, S. Aase, and J. Hakon Husoy, “Method of optimal directions for frame design,” in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 5, pp. 2443–2446 vol.5, 1999.

[31] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[32] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[33] J. Aeschbacher, J. Wu, and R. Timofte, "In defense of shallow learned spectral reconstruction from rgb images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[34] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[35] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural rgb images," in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 19–34, Springer International Publishing, 2016.

[36] J. K. Dutta and B. Banerjee, "Improved outlier detection using sparse coding-based methods," *Pattern Recognition Letters*, vol. 122, pp. 99–105, 2019.

[37] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2790–2797, 2009.

[38] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Computer Vision – ECCV 2008* (D. Forsyth, P. Torr, and A. Zisserman, eds.), (Berlin, Heidelberg), pp. 155–168, Springer Berlin Heidelberg, 2008.

[39] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[40] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[41] 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.

[42] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, “A survey of sparse representation: Algorithms and applications,” *IEEE Access*, vol. 3, pp. 490–530, 2015.

[43] H. Cheng, Z. Liu, L. Yang, and X. Chen, “Sparse representation and learning in visual recognition: Theory and applications,” *Signal Processing*, vol. 93, no. 6, pp. 1408–1425, 2013.

[44] N. Lee, “Map support detection for greedy sparse signal recovery algorithms in compressive sensing,” *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 4987–4999, 2016.

[45] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *The Annals of Statistics*, vol. 32, no. 2, pp. 407 – 499, 2004.

[46] I. Gorodnitsky and B. Rao, “Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm,” *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[47] L. Shuang, “Sparse representation of hardy function by iteratively reweighted least squares,” in *2020 International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*, pp. 57–60, 2020.

[48] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, “Simple and practical algorithm for sparse fourier transform,” in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1183–1194, SIAM, 2012.

[49] M. Stéphane, “Chapter 7 - wavelet bases,” in *A Wavelet Tour of Signal Processing (Third Edition)* (M. Stéphane, ed.), pp. 263–376, Boston: Academic Press, third edition ed., 2009.

[50] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. USA: Academic Press Professional, Inc., 1990.

[51] M. Do and M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[52] E. J. Candes and D. L. Donoho, “Curvelets: A surprisingly effective nonadaptive representation for objects with edges,” tech. rep., Stanford Univ Ca Dept of Statistics, 2000.

[53] E. Le Pennec and S. Mallat, “Bandelet image approximation and compression,” *Multiscale Modeling & Simulation*, vol. 4, no. 3, pp. 992–1039, 2005.

[54] R. R. Coifman and M. V. Wickerhauser, “Adapted waveform analysis as a tool for modeling, feature extraction, and denoising,” *Optical Engineering*, vol. 33, no. 7, pp. 2170 – 2174, 1994.

[55] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” vol. 9006, pp. 111–126, Cremers, D, Springer, 2015.

[56] A. Alvarez-Gila, J. van de Weijer, and E. Garrote, “Adversarial networks for spatial context-aware spectral image reconstruction from RGB,” *CoRR*, vol. abs/1709.00265, 2017.