Gland Segmentation in Hyperspectral Images using Unsupervised methods for Cancer detection in Colon

A B. Tech Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

by

Vinay Chandragiri
(120101018)

under the guidance of

Dr. Amit Sethi & Dr. Saswata Shannigrahi

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Gland Segmentation in Hyperspectral Images using Unsupervised methods for Cancer detection in Colon** " is a bonafide work of **Vinay Chandragiri** (**Roll No. 120101018**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Guide: Dr. Amit Sethi,
Department of Electronics & Electrical Engineering,
Supervisor: Dr. Saswata Shannigrahi,
Department of Computer Science & Engineering,
Indian Institute of Technology Guwahati

Date: 12 November 2015

Acknowledgements

I would like to take this moment to express my deep and sincere gratitude to Dr. Saswata Shannigrahi, from the Department of Computer Science for permitting me to work with Dr. Amit Sethi from the Department of Electronics and Electrical Engineering.

I acknowledge with thanks for the kind of support, inspiration and constructive timely guidance which I received from Dr. Amit. I believe this experience will help me through out my career and also motivated me in pursing graduate studies in the field of Machine Learning precisely Deep Learning.

Vinay Chandragiri

Abstract

Hierarchical clustering is particularly important in medical imaging as it helps in visualization of levels of dissimilarity among different cells and tissues. Semi supervised clustering uses the knowledge constraints to cluster the data, most of the existing research in semi supervised clustering use instance level must-link and cannot-link constraints, which cannot be used for hierarchical clustering since data samples are linked over different levels of hierarchy, we propose a 2-stage semi supervised clustering model which gives the necessary hierarchical structure in the data but alleviates large memory requirement problem of hierarchical clustering, hierarchical clustering is done 2nd stage using ultra metric transformation of dissimilarity matrix which is subject to triple wise relative constraints[3]. The efficiency and effectiveness of our proposed method is shown by experimental results.

The features corresponding to the local correlation are extracted effectively by CNNs because of their weight sharing architecture,since spectral features in our HSI data have shown local correlation we used CNN for their extraction, two architectures are proposed, each containing 6 layers, input layer, convolution layer, max-pooling layer, fully connected layer, dropout layer, output layer[5]. Experiments are performed on the HSI cell data set(4 classes), AVIRIS data set, results achieved outperformed support vector machines and 2 layer fully connected neural networks.

Index terms: Hierarchical clustering, Agglomerative clustering, Semi supervised clustering, Non negative matrix factorization, Hyper-spectral image, Convolutional neural network, dropout, Max-pooling, Support vector machine, Spectral feature.

Contents

List of Figures

1. Introduction

1.1 Non negative matrix factorization

Non negative matrix factorization is used to find an approximation of a non negative matrix of low rank, this is used to extract features from the data matrices.
Given a data matrix **D** NMF constructs a 2-factor decomposition of **D** with an aim to minimize the cost function.

$$\mathbf{L} = ||\mathbf{D} - \mathbf{UV}||^2$$

where **U** is the basis matrix, **V** is the feature matrix and norm is the frobenius norm. This minimization can be achieved by using multiplicative updates iteratively[1].

$$\mathbf{U} = \mathbf{U} * (\mathbf{DV^T})/([\mathbf{DV}]\mathbf{V^T})$$

$$\mathbf{S} = \mathbf{V} * (\mathbf{U^T D})/(\mathbf{U^T}[\mathbf{UV}])$$

1.2 Convolutional neural networks(CNN)

CNNs are biologically inspired from the human visual cortex, CNNs have relative location based sparse connections, each filter(hidden layer weights) are spanned across the whole input space, these tied weights not only decreases no of learning parameters but also extract robust high level features.

A pooling layer usually follows the convolution layer, maximum of the input is returned by the max-pooling layer, it produces translational invariance and reduces computational complexity[5].

Traditionally CNNs are used in many visual related classification problems to achieve high classification accuracies, In the proposed model CNNs are used to extract both spectral and spacial features through training.

1.3 Dropout

Dropout[1] is a recent advancement in the field of Artificial Neural Networks(ANNs), it randomly zeros out a small fraction of weights and corresponding connections for each iteration during training[4].

Dropout alleviates the over fitting on the training set, reduces the co-adaptation of the features thus extracting the independent and robust features thus improving the performance on test set.

1.4 Hyperspectral images(HSI)

Hyperspeectral images are characterised by high spectral resolution over a range of the frequencies in the electromagnetic spectrum, large dimension of HSI data is a trade off for large amount of information it captures in the image which is vital in medical imagery, high dimensionality of HSI data also imposes restriction on few preprocessing techniques and enforces a requirement of smart features, both spectral and spatial information embedded in HSI data needs to exploited for maximum classification accuracy.

1.4.1 Datasets

AVIRIS

AVIRIS dataset contains one remote sensing image with 8 classes(fully annotated).

Its is collected by NASA using a airborne drone called Airborne Visible and Infrared Image Spectrometer, used as trade mark dataset to test training models for HSI datasets.

HSI cell

HSI cell dataset contains 10 medical images with same 4 classes in each image(partially annotated), with each image pixel having 226 spectral bands corresponding to 900 wave numbers to 1800 wave numbers(which are shown in figure 1.1.

- Class0 : background + other cell structures

- Class1 : non-goblet cell epithelium (probably enterocytes)
- Class2 : stroma
- Class3 : goblet cell

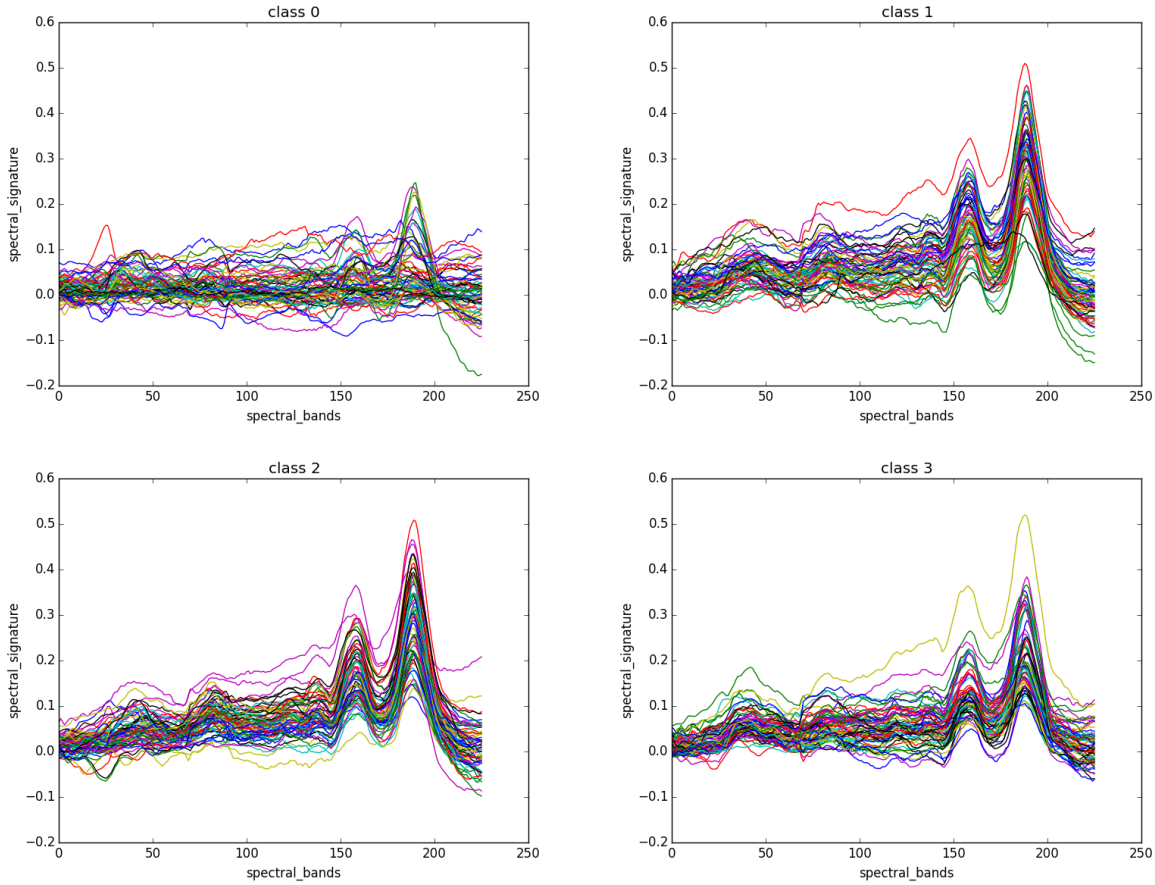


Figure 1.1: Spectral signature vs bands(0 225) for the 4 classes in HSI cell dataset.

1.5 Challenges

- High Dimensionality: By their very definition HSI are high dimensional thus increasing computational complexity.

- Imaging Variability: Difference in any or all of illumination conditions, backgrounds, imaging spectrometers produce variability in the data, which can be observed from spectral signatures in figure 1.1.

- Small Training Dataset: HSI cannot be visualized, making the annotation harder and costly,HSI cell dataset is only partially annotated, all the training and test samples must be selected from this partially annotated subset.

2. Semi-supervised Hierarchical Clustering model

2.1 Feature extraction by using SSNMF

Semi-Supervised Non negative Matrix Factorization is used to extract the features from a data matrix $\mathbf{D} = [d_1, d_2, ..., d_n]$ each column is a m-dimensional data sample which belongs to one of the k classes, $\mathbf{L} = [l_1, l_2, ..., l_n]$ is the label matrix, **V** is the feature matrix, **U** and **X** are the basis matrices for **D** and **L** respectively, **W** is the weight matrix to handle missing labels, λ is the trade-off parameter for the supervised term.

$$W_{ij} = \begin{cases} 0.001, & \text{if } L_{ij} = 1 \\ 1, & \text{if } L_{ij} = 0 \\ 0 & \text{if } L_{ij} \text{ is unknown} \end{cases}$$

Weightage for $L_{ij} = 1$ is relaxed so that the corresponding estimated value is non-zero but not necessarily close to one[1].

The cost function **J** is given by

$$\mathbf{\Phi} = ||\mathbf{D} - \mathbf{UV}||^{\mathbf{2}} + \lambda||\mathbf{W} * (\mathbf{L} - \mathbf{XV})||^{\mathbf{2}}$$

Multiplicative updates are used iteratively for **U**,**X** and **V** to minimize the cost function.

$$\mathbf{U} = \mathbf{U} * (\mathbf{DV^T})/([\mathbf{UV}]\,\mathbf{V^T})$$

$$\mathbf{X} = \mathbf{X} * ([\mathbf{W} * \mathbf{L}]\,\mathbf{V^T})/([\mathbf{W} * \mathbf{XV}]\,\mathbf{V^T})$$

$$\mathbf{V} = \mathbf{V} * (\mathbf{U^T D} + \lambda \mathbf{X^T}\,[\mathbf{W} * \mathbf{L}])/(\mathbf{U^T}\,[\mathbf{UV}] + \lambda \mathbf{X^T}\,[\mathbf{W} * \mathbf{XV}])$$

NMF computational complexity is $\Theta(mnr)$ and that of SSNMF is $\Theta((m+k)nr)$, since $k << m$ SSNMF extracts better discriminative features with approximately same complexity as NMF.

2.2 Stage 1 of semi supervised hierarchical clustering

The feature vectors extracted are first over clustered using a partitional clustering algorithm, here we used k-means algorithm to form the clusters, number of cluster is chosen such that no cluster has the labeled samples of two or more different classes.

Initially start clustering with number of clusters same as the number of classes in the dataset, if any two labeled samples of different classes are present in the cluster then double the number of clusters and repeat until no two labeled samples of different classes are present in the cluster.

These initially formed clusters means are used as the data samples in the hierarchical clustering process.

2.3 Stage 2 of semi supervised hierarchical clustering

Given the set of feature vectors $\mathbf{V} = \{v_1, v_2, ..., v_n\}$, their pairwise dissimilarities $\mathbf{D} = \{d(v_i, v_j)|v_i, v_j$ and a set of constraints $\mathbf{C} = \{(v_i, v_j, v_k)|d(v_i, v_j) \leq (v_i, v_k), v_i, v_j, v_k \in V\}$, semi supervised hierarchical clustering tries to maintain the merge order in dissimilarities while trying to satisfy as many knowledge constraints as possible, thus giving a hierarchy of clusters or a dendogram.

2.3.1 Ultra Metric Distance

In agglomerative clustering two clusters are merged at each step to form a new one such that

$$d(M_i, M_J) \leq \min(d(M_i, M_k), d(M_j, M_k))$$

This property is called **reducibility property**,dissimilarities satisfy the ultrametric inequality, if the reducibility property holds.

$$d(v_i, v_j) < max(d(v_i, v_k), d(v_j, v_k)), \forall v_i, v_j, v_k \in V$$

Thus the original dissimilarity matrix subject to ultrametric transformation gives a hierarchical clustering, this transformation can be modeled mathematically as finding the optimal ultrametric

distance matrix[3].

$$arg_{D^*} min \sum_{x_i, x_j \in X} (D_{ij} - D^*_{ij})^2$$

This least squares optimization is NP-hard and three approximation approaches- transitive dissimilarity approach, iterative projection approach and SUMT approach are discussed.

2.4 Transitive Dissimilarity

By treating D as the transition matrix on a graph in which each column represents node in a graph a transitive dissimilarity matrix is constructed.

Consider a path $Path_{ij}$ between v_i and v_j, transitive dissimilarity on the path $Path_{ij}$ is defines as

$$\mathbf{T(Path_{ij})} = max(d_{i,k_1}, d_{k_1,k_2}, ..., d_{k_{n-1},k_n}, d_{k_n,j})$$

Minimal transitive dissimilarity among all the path between v_i and v_j is given by

$$m_{ij} = min(T(P_{ij}))\text{for given vertices } v_i \text{ and } v_j$$

The minimal transitive dissimilarity satisfies the ultrametric inequality for any weighted dissimilarity graph.

$$m_{ij} \leq max(m_{ik}, m_{kj}) \forall v_i, v_j, v_k.$$

Proof: Let v_i and v_j are two nodes and $Path_{ij}$ is a set of all paths between them, (P_{ik}, P_{kj}) is a path from v_i to v_j via v_k, in all possible paths P_{ij} edge weights of any directly connected vertices is defined as $W(P_{ij})$.

$$\begin{aligned} m_{ij} &= min_{Path_{ij}} max\left[W(Path_{ij})\right] \\ &\leq min(p_{ik}, Path_{kj}) max(W(Path_{ik}, Path_{kj})) \\ &= min(p_{ik}, Path_{kj}) max(max(W(Path_{ik}), max(Path_{kj}))) \\ &= max\left[min_{Path_{ik}}(max\left[W_{Path_{ik}}\right]), min_{p_{kj}}(max\left[W_{Path_{kj}}\right])\right] \\ &= max(m_{ik}, m_{kj}) \end{aligned} \tag{2.1}$$

Modified Floyd-Warshall algorithm is used to compute the minimum transitive dissimilarity[3].

2.5 Optimization using SUMT

In this method we find a least squares approximation of the dissimilarity matrix **D** subject to ultra metricity and the set of triple wise relative constraints, since for a given ultrametric matrix there is a unique cluster hierarchy or dendogram, it sufficies to

$$\text{minimize } \mathbf{L}(\mathbf{D})$$

$$\text{subject to } d_{ij} \leq max(d_{ik}, d_{kj}) \; \forall \text{ i,j,k}$$

$$\text{and } d_{ij} - d_{ik} \leq 0 \; \forall \; (x_i, x_j, x_k) \in \mathbf{C}$$

Here $\mathbf{L}(\mathbf{D}) = \sum \sum_{i<j} (\delta_{ij} - d_{ij})^2$, $\delta_{ij} = \delta(i,j)$ euclidean distance between the samples i and j, $d_{ij} = d(i,j)$ is $(i,j)^{th}$ element of ultrametric matrix.

This constrained minimization problem is transformed into a sequence of unconstrained minimization problems using SUMT, ie., this is solved by sequentially minimizing the augmented function

$$\Phi(\mathbf{D}, q, r) = \mathbf{L}(\mathbf{D}) + qP(D) + rT(D)$$

for a incrementing valus of q and r. $\phi(D, q, r)$ is a linear weighted combination of loss function, penalty function P(D) [2]which enforces ultrametricity on D and penalty function T(D) which enforces triple wise relative constraints.

$$P(D) = \sum_{\Omega} (d_{ik} - d_{jk})^2$$

$$\text{with } \Omega = \{(i,j,k) | d_{ij} \leq min(d_{ik}, d_{kj}) \text{ and } d_{ik} \neq d_{jk}\}$$

ie., Ω is set of all 3-tuples that violate the ultra metric inequality

$$T(D) = \sum_{\Omega} (d_{ik} - d_{jk})^2$$

$$\text{with } \Omega = \{(i,j,k) | d_{ij} > d_{kj} \text{ and } (x_i, x_j, x_k) \in C\}$$

ie., Ω is set of all 3-tuples that violate the knowledge constraints.

The initial estimate of **D**, $D^{(0)}$ is determined by adding a zero mean random noise with varience

$$\frac{2}{3n(n-1)} \sum \sum_{i<j} (\delta_{ij} - \delta^*_{ij})^2$$

$$d_{ij}^{(0)} = \delta_{ij} + \epsilon_{ij}$$

.

(1) Initialize t with 1, determine D^0 and define q = $L(D^{(0)})/P(D^{(0)})$ and r = $L(D^{(0)})/T(D^{(0)})$

(2) Minimize $\Phi(\mathbf{D}, q^t, r^t)$ initalizing with $D^{(t-1)}$ to get $D^{(t)}$

(3) Convergence test- if $\sum \sum_{i<j}(d_{ij}^t - d_{ij}^{t-1})^2$ is less than a small predefined constant stop, else continue.

(4) Increment $q^{t+1} = 10 * q^t$, $r^{t+1} = 10 * r^t$ and repeat from step (2)

Initial values of q and r are chosen such that equal weightage is given to all terms in cost function at t=1, Powell's conjugate gradient procedure with automatic restarts is used for optimization in step (2).

3. Convolutional Neural Network based model

3.1 Background labelling

The HSI cell dataset has only three class labels and the class1 is annotated by using the filtering techniques, we use 3 filters Max, Min ,abs and then map the pixel values to 0 or 1 to get binary image which correspond to labels of class 0.

img corresponds to input image of size (a x b x 226)

3.1.1 Max filter:

Filtering window is along spectral dimension with size (1 x 1 x 226), it spans the input image(img) pixel by pixel to produce output of shape (a x b x 1).

3.1.2 Mix filter:

Similar to Max Filter window size is (1 x 1 x 226), it spans the input image(img) pixel by pixel to produce output of shape (a x b x 1)

3.1.3 Abs filter:

Abs Filter has window size is (1 x 1 x 1), it spans the input image of shape (a x b x 1) pixel by pixel and returns absolute value of each pixel to produce output of shape (a x b x 1)

Resultant of $Max(\mathbf{img}) - Abs(Min(\mathbf{img}))$ is mapped to 0 or 1 to get the labels of class 1

$$\text{pixel map} = \begin{cases} 1, & \text{if pixel value} < 0.17 \\ 0, & \text{otherwise} \end{cases}$$

labels of class $1 = Max(\mathbf{img}) - Abs(Min(\mathbf{img})) < 0.17$ here $<$ is the Relational operator

3.2 Spectral feature extraction and segmentation

Traditionally CNNs performs best in extracting local shapes as features due to their weight sharing model,as it can be seen in Figure 1(spectral signature vs bands) spectral signature exhibits local correlation among the adjacent bands, thus justifying our choice of CNN for spectral feature extraction[6].

The architecture contains input layer **I1**, convolutional layer **C2**, max pooling layer **P3**, dropout layer **D4**, fully connected layer **F5**, output layer **O6**, as shown in the Figure,

- **I1** - Each pixel vector is taken in the input layer of size (n_1,1) where n_1 is no of bands.
- **C2** - n_1 x 1 input data are taken, 20 filters each of size k_1 x 1 are applied to produce an output of size n_2 x 1 x 20, where $n_2 = n_1 - k_1 + 1$. Between input layer and **C2** there are 20 x (k_1 + 1) parameters to be trained.
- **P3** - Max pooling layer applies kernel of size (k_2, 1) and has n_3 x 1 x 20 nodes, where $n_3 = n_2/k_2$, there are no trainable parameters between **C2** and this layer.
- **F4** - Fuly connected layer has n_4 no of nodes and there are (20 x n_3 + 1) x n_4 parameters between **P3** and **F4**.
- **D5** - Dropout layer has n_4 no of nodes same as previous layer taking values either 0 or 1, there are no trainable parameters, one hyperparameter p(probability of a node taking the value 0).
- **O6** Output layer has n_5 nodes where n_5 is no of classes, there are (n_4 +1) x n_5 trainable parameters.

3.3 Training Strategies

3.3.1 Parameter Selection

Weight Initialization

The trainable parameters are initialised with random values uniformly sampled from

$$\left[-\sqrt{\frac{6}{f_{in}+f_{out}}}, \sqrt{\frac{6}{f_{in}+f_{out}}}\right]$$

for the parameters in the layer i f_{in}, f_{out} corresponds to number of units in previous layer and next layer respectively, *tanh* is used as the activation function in **C2** and **F4** layers while *Max()* is used in **M3** layer

Trainable parameters are initialized in a interval close to zero where the activation function has maximum derivative, this makes training and back propagation efficient.

Hyperparameter selection

A calculated guess is made on all the hyper parameters initially based on model sufficiency, and then they are modified through experimentation.//

size of input vector n_1 = 226

size of filter in $\mathbf{C2}$ layer k_1 = 27

number of convolutional filters n_2 = 20

Max pool filter size k_2 = 5

number of nodes in $\mathbf{F5}$ layer n_4 = 100

3.3.2 Forward Propagation

The CNN has 6 layers, let $\mathbf{x_i}$*for i = 1,2...,6* denote the input to the i^{th} layer

$$\mathbf{x_{i+1}} = f_i(\mathbf{u_i})$$

$$\mathbf{u_i} = \mathbf{W_i^T x_i} + \mathbf{b_i}$$

where $f_i, \mathbf{W_i}, \mathbf{b_i}$ are the activation function, wight matrix and bias vector for i_{th} layer.

Output of the **D5** layer is given to the *softmax* function which produces the distribution over the output class labels

$$y = \frac{1}{\sum_{k=1}^{n_5} e^{\mathbf{W_{i,k}^T x_i + b_{i,k}}}} \begin{bmatrix} e^{\mathbf{W_{i,1}^T x_i + b_{i,1}}} \\ e^{\mathbf{W_{i,2}^T x_i + b_{i,2}}} \\ \vdots \\ e^{\mathbf{W_{i,n_5}^T x_i + b_{i,n_5}}} \end{bmatrix} \tag{3.1}$$

3.3.3 Backward Propagation

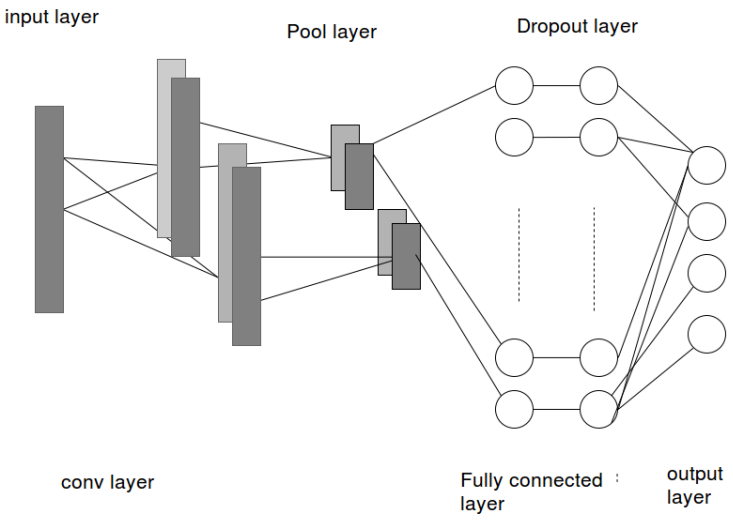
Negative log likelihood function is used as the cost function by comparing the class labels with those predicted using the current parameters, the cost function is given as

$$L(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n_5} I\left[j = \mathbf{Y^{(i)}}\right] log\left(\mathbf{y_j^{(i)}}\right)$$

where $\mathbf{y_j^{(i)}}$ is the j_{th} value of the predicted label $\mathbf{y^{(i)}}$ of the i_{th} sample in training set, $\mathbf{Y}$ is actual output label, m is number of training sample, $\mathbf{I}[]$ is the indicator function.

The gradient of the cost function is taken with respect to each of the parameters and the trainable parameters are updated using RMS propagation algorithm.

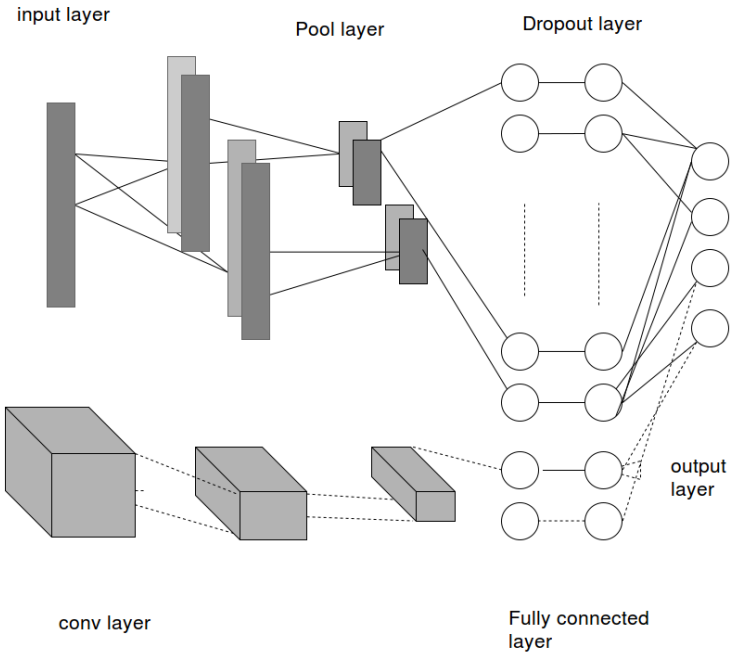
In RMS propagation the the gradient is scaled down by the running average of the gradient magnitude, running average is taken with expotential decay to ensure that recent gradient have maximum effect.



Training is done in mini batches with batch size of 500 so that each mini-batch still has the samples from different classes for effective training.

3.4 Segmentation using concatenated spectral and spatial features

Spectral features are extracted as mention in above section.



(8 x 8 x 226)Neighbourhood of the pixel vector is selected for spatial feature extraction and 20 convolutional filters each of size(5 x 5 x 226) are applied on this neighbourhood to get (4 x 4x 20) output, which is passed to a max pooling filter of size(2 x 2 x 1).

Max pooling is done with overlapping to get an output of size(3 x 3 x 20) which is then flattened and concatenated with spectral features, this new feature vector is given to the fully connected layer and classified[5].

4. Results

4.1 Classification Results

4.1.1 Classification results for 2-stage clustering model.

Clustering of each image is done separately, each time 80% of labeled samples are used in the semi supervised clustering by formulating triple wise relative constraints exhaustively, rest 20% are retained for calculating the clustering accuracy.

Two stages in the clustering model restrict us from using F score as a metric for clustering accuracy.

Accuracies over different images are weighted with ratio of number of test samples of that image to the total number of test samples and then mean value is calculated to get the average clustering accuracy of 97.1%.

4.1.2 Classification results for CNN model.

The proposed CNN model is tested on AVIRIS datset, it gives an accuracy of 91.82 percent on test set.

The HSI cell data contains 10 images, all of them partially annotated, dataset is extracted from this partially annotated subset of the original data and used for both training and testing.

Half of the labeled data from each class is used for training including the validation, remaining half is used for testing.

An accuracy of 96.13 percentage is attained on this test set

With the same amount of training samples SVMs achieved 87.31 percent and 2 layer NN

achieved 90.31 percent classification accuracies.

4.2 Comparision of classification results

The CNN classifier and semi supervised hierarchical clustering classifier are used on the original image data both initially annotated and non-annotated sets and the results are displayed in the figure 4.1- 4.10.

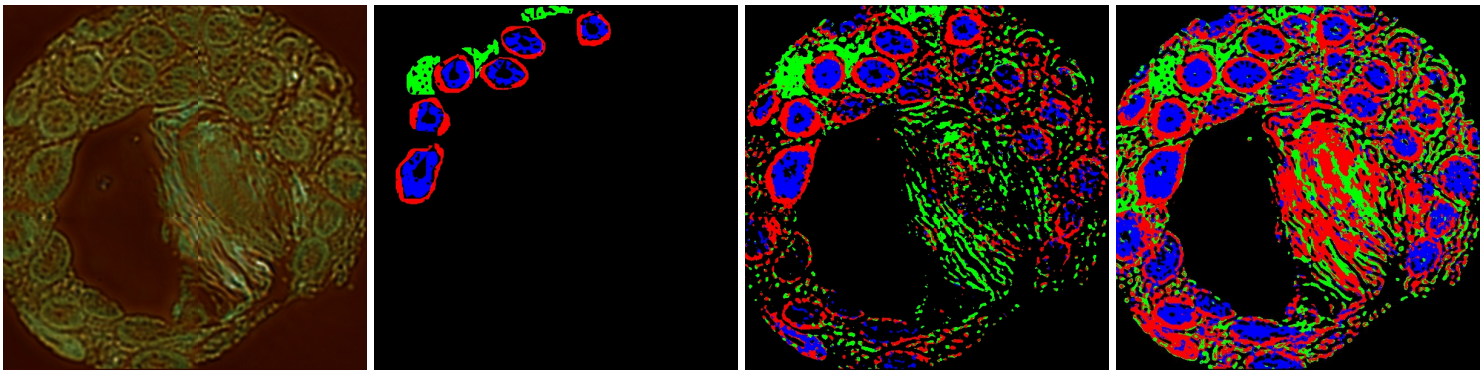


Figure 4.1: Original HSI, Partial labels, CNN results, 2-stage clustering results for img1.

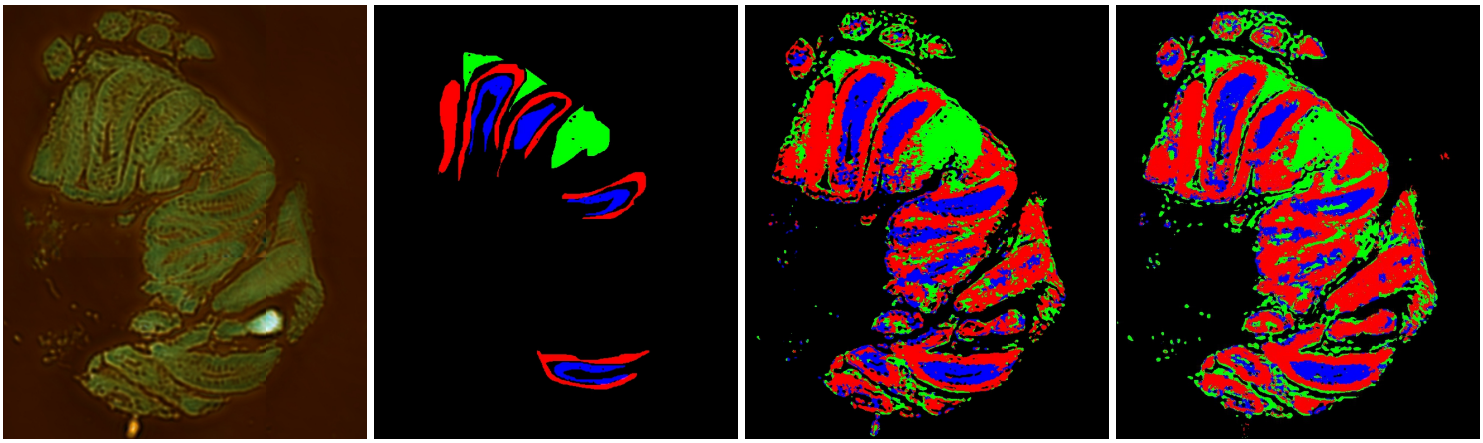


Figure 4.2: Original HSI, Partial labels, CNN results, 2-stage clustering results for img2.

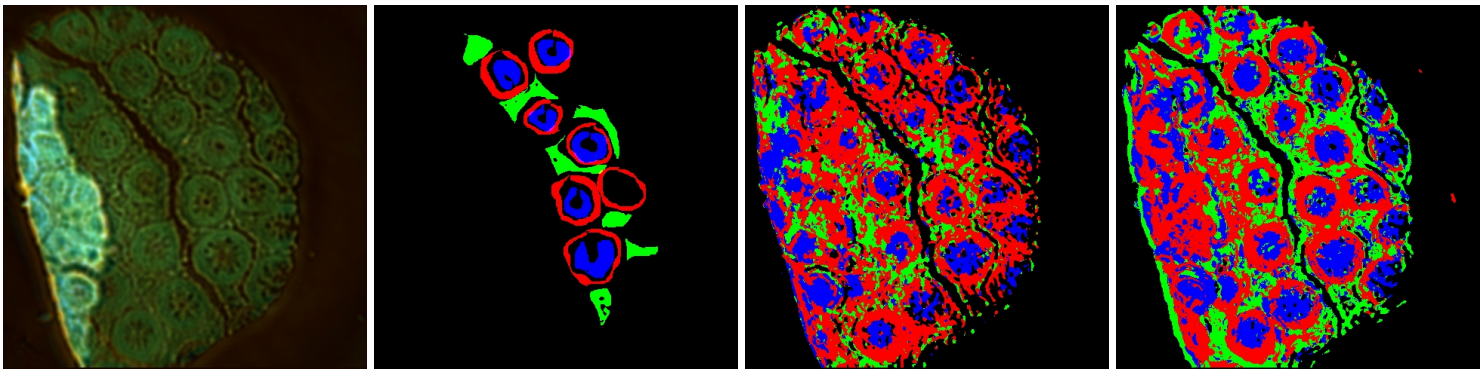


Figure 4.3: Original HSI, Partial labels, CNN results, 2-stage clustering results for img3.

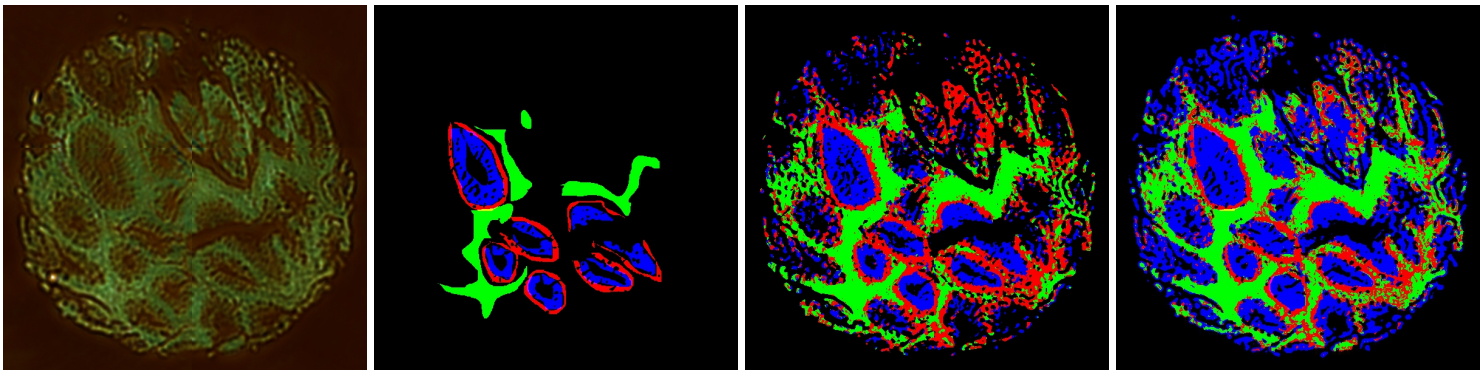


Figure 4.4: Original HSI, Partial labels, CNN results, 2-stage clustering results for img4.

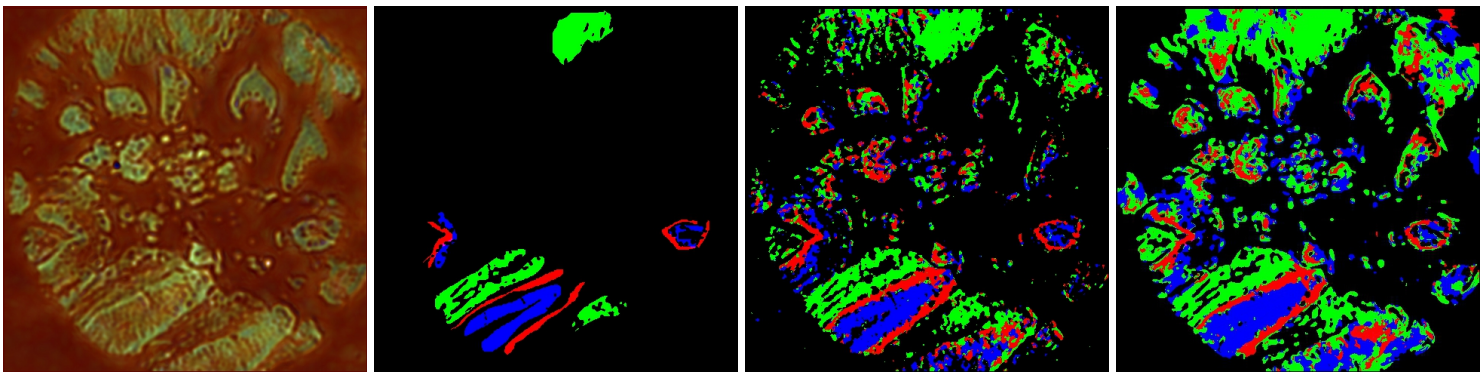


Figure 4.5: Original HSI, Partial labels, CNN results, 2-stage clustering results for img5.

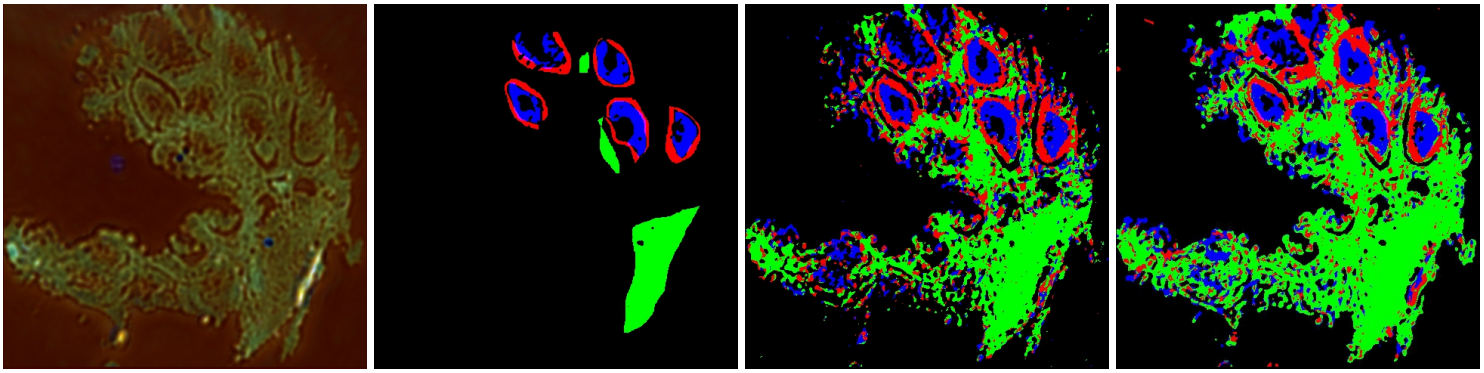


Figure 4.6: Original HSI, Partial labels, CNN results, 2-stage clustering results for img6.

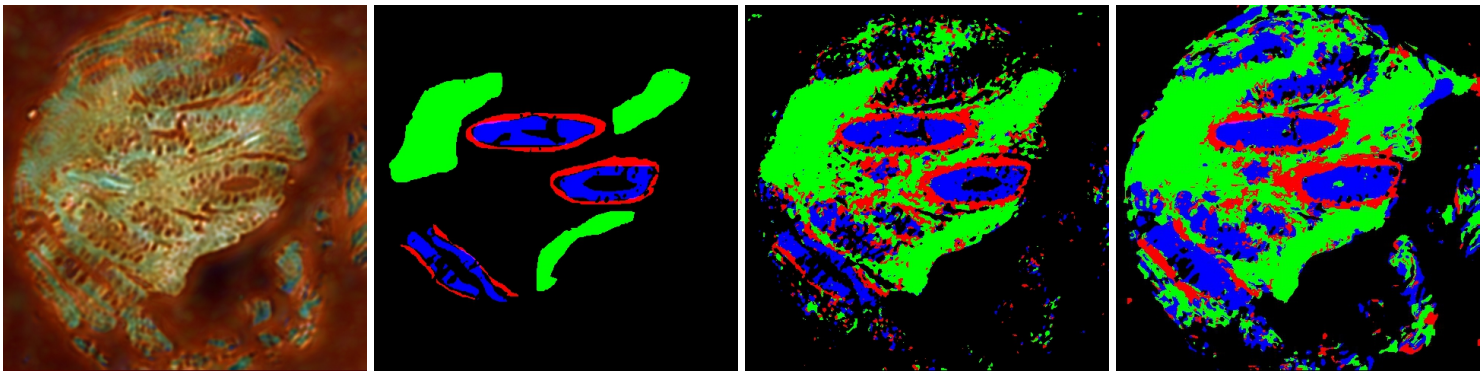


Figure 4.7: Original HSI, Partial labels, CNN results, 2-stage clustering results for img7.

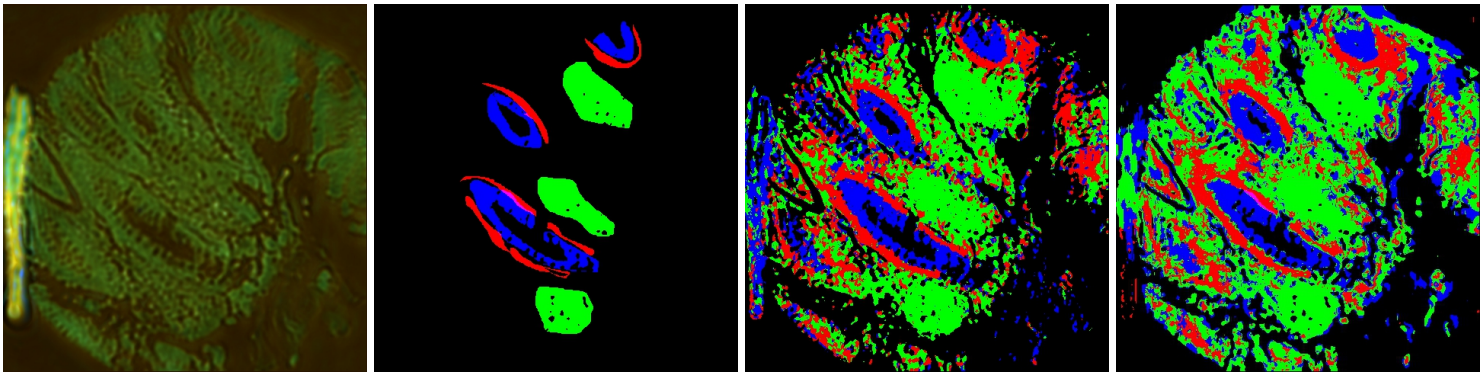


Figure 4.8: Original HSI, Partial labels, CNN results, 2-stage clustering results for img8.

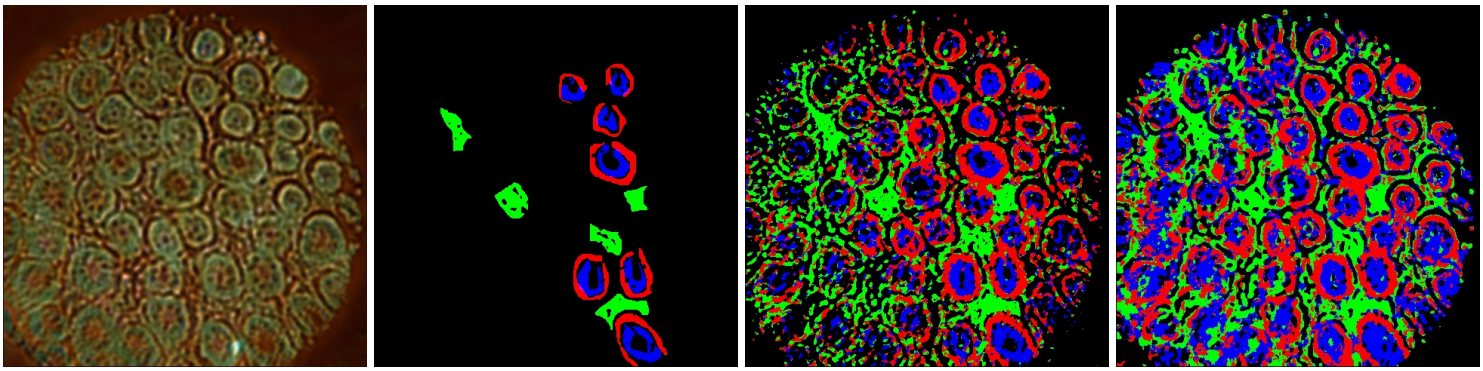


Figure 4.9: Original HSI, Partial labels, CNN results, 2-stage clustering results for img9.

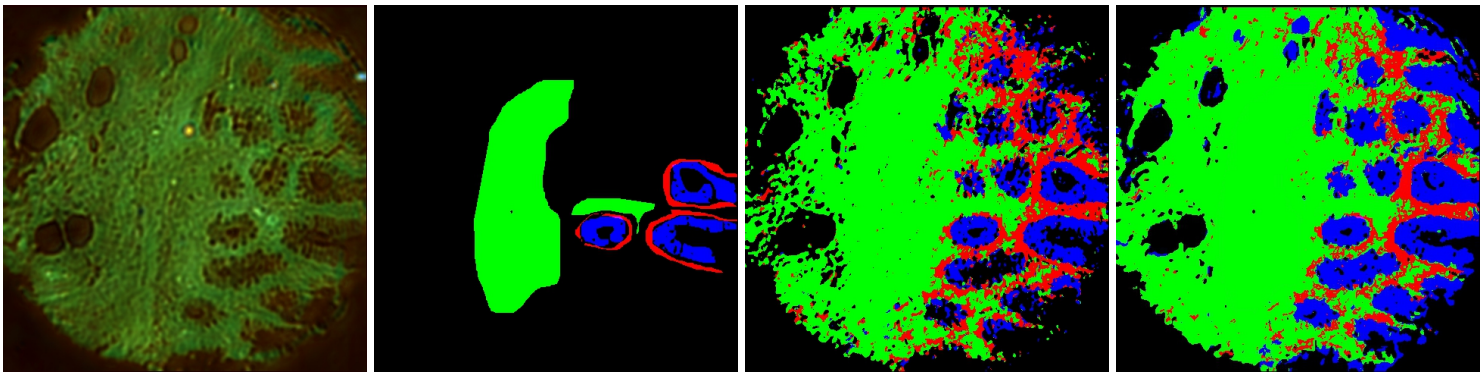


Figure 4.10: Original HSI, Partial labels, CNN results, 2-stage clustering results for img10.

5. Visualisation

5.1 Filter Visualisation

The CNN filters obtained after training are shown in the figure 5.1 and 5.2
They provide a great amount of information on the detailed analysis.

Fourth filter is selecting bands close to 200, as can be seen from the spectral signature figure bands around 200 have high magnitude and variability among the four classes.

Similarly various spectral characteristics can be studied from this filter data, once the spectral variability across class is known then classification can even be done in deterministic manner through thresholding.

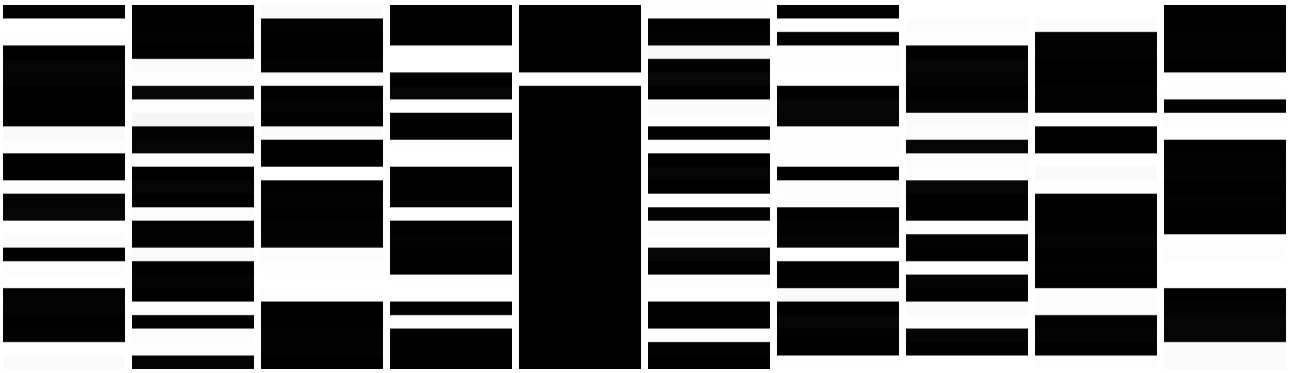


Figure 5.1: Weights of the classifier in the convolution layer for filters 0-9.

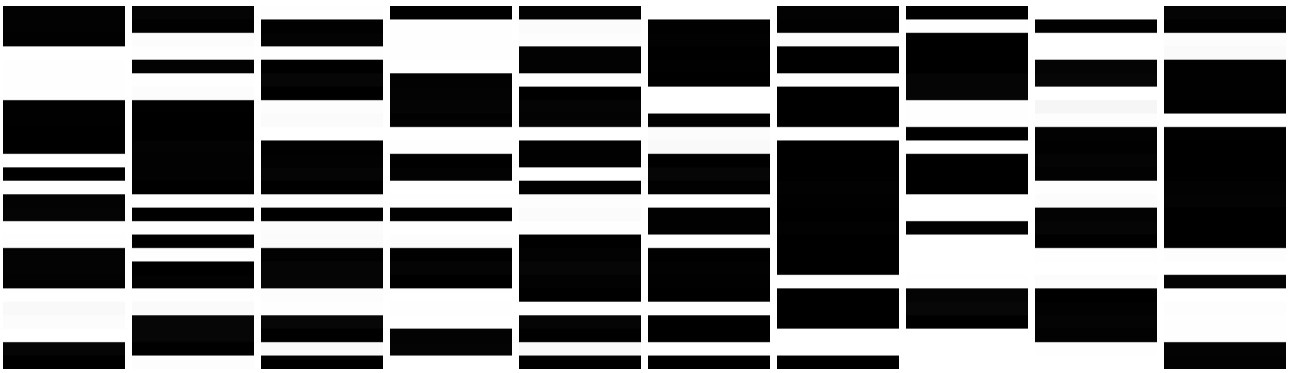


Figure 5.2: Weights of the classifier in the convolution layer for filters 10-19.

6. Conclusion

6.1 Training error vs Validation error

As can be seen from the figure 6.1 training error almost decreases monotonically, which helps us interpret that there is no over fitting despite the fact that dataset is high dimensional and very few samples are used for training.

From this we can conclude that dropout is working as expected.

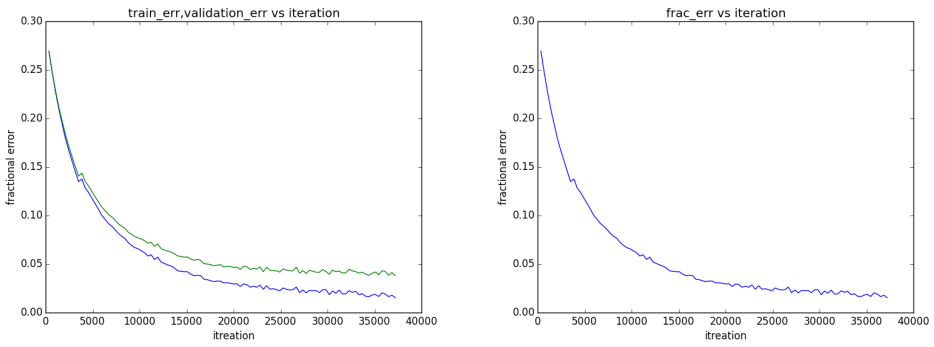


Figure 6.1: Training error and Validation error vs iterations, Training error vs iteration

Bibliography

[1] Lee, Hyekyoung, Jiho Yoo, and Seungjin Choi. ”Semi-supervised nonnegative matrix factorization.” Signal Processing Letters, IEEE 17.1 (2010): 4-7.

[2] De Soete, Geert. ”A least squares algorithm for fitting an ultrametric tree to a dissimilarity matrix.” Pattern Recognition Letters 2.3 (1984): 133-137.

[3] Zheng, Li, and Tao Li. ”Semi-supervised hierarchical clustering.” Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE, 2011.

[4] Srivastava, Nitish, et al. ”Dropout: A simple way to prevent neural networks from overfitting.” The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[5] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li, Deep Convolutional Neural Networks for Hyperspectral Image Classification, Journal of Sensors, vol. 2015, Article ID 258619, 12 pages, 2015. doi:10.1155/2015/258619

[6] Bernard, Kevin, et al. ”Spectralspatial classification of hyperspectral data based on a stochastic minimum spanning forest approach.” Image Processing, IEEE Transactions on 21.4 (2012): 2008-2021.