Lung Tissue Classification using Graph Signal Processing

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
Tehran, Iran
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
Tehran, Iran
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
Tehran, Iran
email address or ORCID

***Abstract*—**

***Index Terms*—Graph, Visibility graph, Textue, ILD**

I. Introduction and Related Work

II. Method

A. Graph wavelet filter banks

Wavelet filter banks are among the most popular methods for extracting features from images or textures. One of their variations is graph wavelet filter banks that consider the relation of different pixels in the texture. In this work, we used the method proposed by Qiao et al. at [1]. We used two-level decomposition for graph wavelet filters; after calculating the filter response with Meyer kernel on the texture as discussed in [1], we moved a 4×4 window with the stride of 1 on the response of wavelet. For each window, we applied singular value decomposition. F_1, F_2, F_3 sets as discussed in [1] are

$$\begin{aligned} F_1 &= \{max(S_1), max(S_2), ..., \\ &\quad max(S_i), ..., max(S_N)\} \\ F_2 &= \{mean(S_1), mean(S_2), ..., \\ &\quad mean(S_i), ..., mean(S_N)\} \\ F_3 &= \{median(S_1), median(S_2), ..., \\ &\quad median(S_i), ..., median(S_N)\} \end{aligned} \tag{1}$$

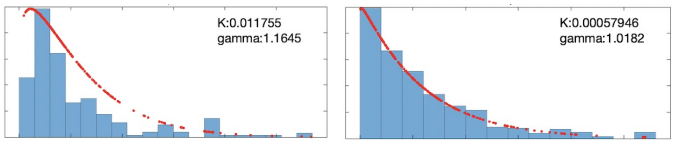
After fitting a Weibull distribution to each set, we will get two parameters, scale and shape parameters.

For each texture, we have two level graph wavelets, and every wavelet has four subbands, every subband results in three F_i sets, which will result in two parameters for Weibull distribution. Finally, we have a 48-dimensional feature vector for each texture.

Figure 1 shows an example of weibull distribution fitting to F_1 and F_2 sets.

B. Visibility Graphs

In recent yeast, increasing the popularity of complex systems and networks motivates researchers to develop methods for transforming classic signals into graph signals which help indicate the relation of signal values. one of these methods is called visibility graphs which are introduced in [2]. This method proposed in that paper is mainly applied to 1D signals. However, the concept is the same for higher dimension signals. As we know, images are 2D signals, and Iacovacci et al. in [3] proposed a new version of visibility graphs for images called image visibility graphs.



(a) (b)

Fig. 1: Example of Weibull fitting to F_i sets

1) Natural Visibility Graphs: We can connect the visibility line between two nodes with any slope in this definition. Instead, we can only have horizontal visibility lines in the horizontal visibility graph. for the time series signal $S(t) = \{x_1, x_2, x_3, ..., x_N\}$, we can define a N node natural visibility graph (N is the numbet of signal points) which x_i and x_j are connected only if we can find a line from x_i to x_j without intercepting other signal values.Equivalently x_i and x_j are connected if following constraint is hold,

$$x_k < x_i + \frac{k-i}{j-i}(x_j - x_i), \forall (i < k < j) \tag{2}$$

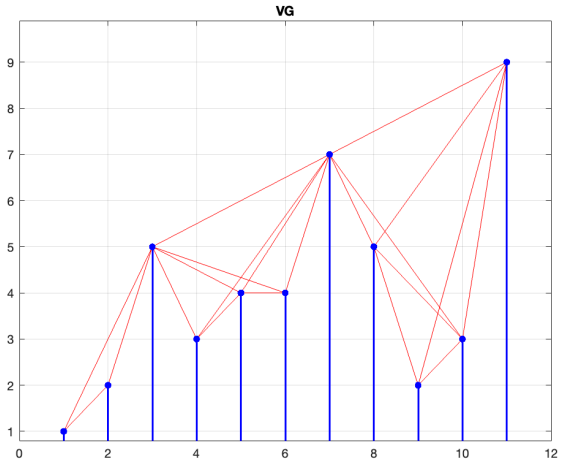
2) Horizontal Visibility Graphs: The definition of horizontal visibility graphs is the same as natural visibility graphs, but we are only allowed to have horizontal lines instead of any slope for visibility lines. The Following equation depicts this definition,

$$x_k < inf(x_i, x_j), \forall (i < k < j) \tag{3}$$

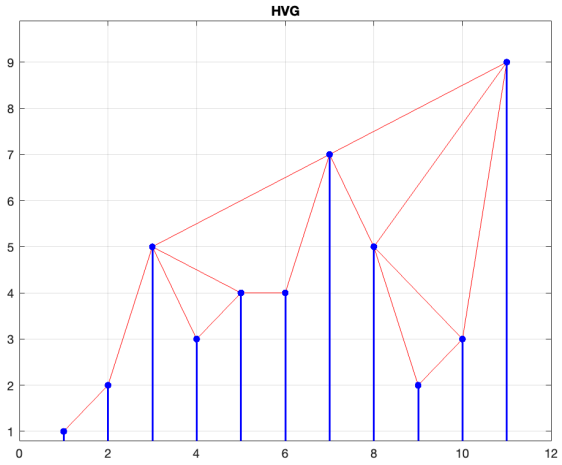
Figure 2 is an example of HVG and NVG for a time-series signal.

C. Image Visibility Graphs

Visibility graphs showed a great potential to define the relation in time series signals, and they easily apply to 2D signals, i.e., images. Iacovacci et al. at [3] developed the idea of visibility graphs for images. We had two definitions in time series signals for VG's, and we also have two definitions for



(a) Natural visibility graph



(b) Horizontal visibility graph

Fig. 2: Example of NVG and HVG for a time series signal

image visibility graphs derived from the time series version of VG's.

1) ***INVG*** *(Image natural visibility graph):* The natural visibility graph for an $N \times N$ image is a graph with N^2 nodes. Each node corresponds to a pixel, and the node's value is equal to the intensity of the corresponding image pixel. Each node is labeled as I_{ij} wich i and j indicate the position of the pixel in the image.

According the definition proposed in [3],nodes I_{ij} and node $I'_{i'j'}$ are connected if following conditions hold,

- $(i = i')\ OR\ (j = j')\ OR$
  $[\exists k \in \mathbb{Z} : \big((i = i' + k)\ AND\ (j = j' + k)\big)]$
- I_{ij} *and* $I'_{i'j'}$ *are connected in the sequence* $\{I_{ij}, I'_{i'j'}, ij, i'j'\}$ *with the definition of NVG for time-series signals.*

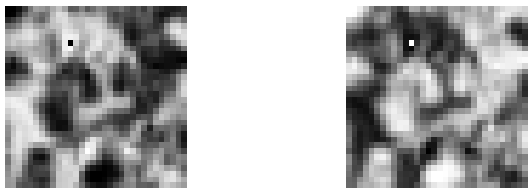
2) ***IHVG*** *(Image Horizontal visibility graph):* The definition of the IHVG is the same as the INVG, but in the second condition, we have to change NVG with the HVG.

3) ***Global and local features for IVG's:*** After defining the IVGs, we have to extract features to use them for classification; Iacovacci et al. at [3] introduced global features to comprise global properties of the image and local features for compacting the local characteristics of the image. For the global feature, we used degree distribution, $p(k)$, and visibility patch with the order of 3, VP_3 as defined in [3] for the local feature, even though the global feature did not use in the final feature selection process which is discussed in section III

III. Feature Selection

This section will discuss the various features that we can choose for our final feature vector; firstly, we discuss each component individually and then demonstrate the final feature vector.

1) ***Image, reversed-image:*** As we showed before, the nature of the IVG and HVG definition behaves differently about the local maxima, so taking into account this fact, we calculate features for the original image and reverse image, which is obtained by replacing each pixel value with the maximum pixels value subtracted by the current pixel value. Figure 3 shows an example of a texture and its reverse.



(a) Original image (b) Reversed image

Fig. 3: original and reversed image of a Fibrosis texture

2) ***IHVG, INVG:*** As we discussed before, we had to choose to define our graphs on images,i.e., natural or horizontal visibility graphs. Our investigation indicates that both of these definitions are useful for describing the textures, and we finally chose features from both of these graph types. Figure 4 is an artifial 3×3 image and INVG is drawn for with Lattice and with No-lattice condition.

(a) With lattice (b) Without lattice

Fig. 4: An example of INVG with and without lattice

3) ***Lattice, without lattice:*** Based on the definition of a visibility graph, each pixel is connected to all of the neighbor pixels, so for simplicity, we can choose to consider these links or not. If we decided not to consider these links, we would

have a visibility graph with no-lattice and vice versa. No-lattice graphs result in a more simple graph representation and a more sparse adjacency matrix.

It is good to mention that we can choose both local and global features for each visibility graph configuration; we implemented local feature extraction with 3×3 windows and stride one. Also, for global features, we calculate the degree distribution for the graph and plot the histogram of it, then fit a Weibull distribution to that, which will give us two elements for the feature vector, so each of the visibility graph feature vectors is a 258-dimensional vector which the first 256 element is for local feature extraction, and the rest is for global features.

*4) **wavelet**:* The method discussed in [1] is a very innovative approach to defining a graph-based feature vector for a texture. We implement that method with a 2-level bipartite graph, and two levels of image,i.e., only one subsampling from 32×32 texture resulting in a 16×16 texture, and with Meyer kernel for graph wavelet filter with a filter length of 24. This method for each texture results in a 48-dimensional feature vector, and we can calculate it for both original textures and reversed textures, as discussed before.

***Final choice**:* Suppose we concatenate all of the possible choices for the final feature vector. In that case, it will give us a 2160-dimensional feature vector for a 32×32 texture image which is not desired and can cause overfitting of the model and harder convergence for the classifier. So we sort each feature vector component based on their part in the total variance of data; this job is done by the same strategy of primary component analysis. Before that, we have to make the feature vectors zero-mean and unit length (due to zero-mean, this has the same effect of making the data unit variance) to make all data have the same scale. Otherwise, we do not get the correct result. Also, an essential consideration in machine learning theory is to only perform PCA and data-preprocess (zero-mean and unit length) on Train data, which will be discussed later and not on the full dataset. Figure 5 shows a pie chart that indicates part of the total variance of data for each possible feature vector component. We should mention that sum of the variance of all global feature extraction methods for visibility graph components was only 1.3% of all variance. Hence, we decided to remove them from the final feature vector for training our classifier. We concatenate all possible feature vector components except three Image-IVG-Nolattice, Reversed-Image-HVG-Nolattice, and Reversed-Image-IVG-Nolattice, which will give us a 1376-dimensional feature vector for each texture.

IV. Experiments and Results

A. Dataset

We used a dataset of interstitial lung disease (ILD) [4] cases containing HRCT images of 128 patients with the inter-slide distance of 10mm, slice thickness of 1mm, and in-plane resolution of 512×512 pixels. In each 2D-single image of a patient, regions of interest (ROIs) are hand-drawn and labeled by two expert radiologists with 20 and 15 years of experience used as ground truth.

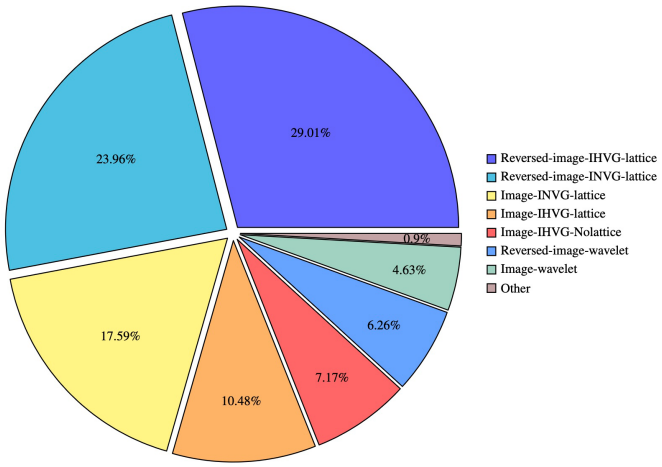


Fig. 5: Importance of each feature vector component based on the variance of all data

The dataset consists of 17 types of lung textures, but in this work, we only used the five most frequent ones, which are Healthy (H), Ground glass (GG), Micronodules (M), Emphysema (E), and Fibrosis (F). This choice only left us 108 patients from a total patient number of 128.

For detecting and extracting textures from HRCT image series, in each 2D image with defined ROIs, we moved a window with the size of 32×32 pixels on the lung mask and if the percentage of window area that covers the specific ROI with a defined texture label was greater than a chosen threshold (the chosen threshold is 0.8).

For extracting textures, we mainly used the implementation that Anthimopoulos et al. [5] provided in a GitHub repository [1]. However, we modified and adjusted the code for our work.

Table I shows the distribution of textures among various classes.

TABLE I: Confusion Matrix for texture classification

Tissue Category	# patients	# patches
H	15	7828
GG	33	2986
M	18	11629
E	9	1981
F	39	5251

B. Data preprocessing & Classifier

Due to unbalanced data distribution between various classes, we used five-fold stratified cross-validation. For each fold, we make train data zero-mean with a unit length. Then, with the aim of the primary component analysis, we only chose features that makeup 94.85% of all data variation. Before PCA feature vector length was 1376, and after PCA was 616. We must use the same transform applied to train data for data to preprocess

[1] https://github.com/ChelovekHe/ild-cnn

and PCA for test data in each fold. After this transformation, the mean of the test data is not necessarily zero.

For the classifier, we used support vector machine, SVM, with a gaussian kernel and one-vs-other strategy for our multiple class problem. Also, to address the imbalanced distribution of data between classes, we used weighted SVM to solve this issue,i.e., for the class with the least data, the SVM weight is the largest.

C. *Performance evaluation*

After extracting features and preprocessing the data, all data are randomly partitioned into five parts. One of the parts is used for test data and the others for train. This process is repeated five times, and each time a different partition is used for test data, so in the end, all of the data is used for testing. This method is called five-fold cross-validation. After training an SVM classifier in each fold, we concatenate the result for the test partition, and a final evaluation is performed on this concatenated set. The following metrics measure our method's performance, and for each class, we used the one-vs-rest (i.e., OVR) policy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$TN - rate = \frac{TN}{TN + FP} \tag{7}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \tag{8}$$

$$AUC = the\,area\,under\,the\,ROC\,curve \tag{9}$$

Table II shows the confusion matrix for the final classifier. In table III, we summarize the performance of our method for each class and also overall classification performance.

TABLE II: Confusion Matrix for texture classification

	Predicted label				
	H	GG	M	E	F
H	7688	51	73	11	5
GG	80	2851	21	0	34
M	286	43	11214	4	82
E	42	6	11	1917	5
F	20	57	42	3	5129

TABLE III: Results

	Accuracy	Recall	Precision	TN-rate	AUC	F1-score
H	98.09%	98.21%	94.73%	98.04%	98.13%	96.44%
GG	99.02%	95.48%	94.78%	99.41%	97.45%	95.13%
M	98.11%	96.43%	98.71%	99.19%	97.81%	97.56%
E	99.72%	96.77%	99.07%	99.94%	98.35%	97.91%
F	99.16%	97.68%	97.6%	99.48%	98.58%	97.64%
Overal	97.05%	96.98%	96.91%	99.21%	-	96.93%

D. *Comparison to previous work*

In this subsection, we compare our result with some of the recent performances on the ILD texture classification; table IV shows this comparison.

TABLE IV: Comparison of ILD texture classification with the state-of-the-art performances.

Reference	**Accuracy**	**F1-score**	**Method**
Dudhane et al. [6]	93.41%	-	GLCM, GLRLM, hist. of LBP
Song et al. [7]	86.1%	-	LMLE
Depeursinge et al. [8]	86%	-	CBIR, 3D localization
Doddavarapu et al. [9]	94.67%	-	CNN model
Gao et al. [10]	92.8%	-	CNN model
Anthimopoulos et al. [11]	-	89%	DCT filter bank,gray-level hist.
Guo et al. [12]	-	98.4%	SK-DenseNet (CNN model)
Huang et al. [13]	-	96%	CNN model
Our method	**97.05%**	**96.93%**	**IVG, graph wavelet**

V. Conclusions

Our study evaluated the performance of graph signal processing (i.e., GSP) methods to classify the five most frequently appeared lung tissues in ILD patients. Some of the previous works that have been done in the literature are only using the deep learning method for feature extraction. Generally, they get higher accuracy on this dataset. Although we insist on not using the deep learning method, we achieved higher performances than most of these methods and the best performance in the classic methods. We inspired our work by the success of [14], and [1] using the image visibility graphs and graph wavelets for texture classification.

The nature of the visibility graph differentiates between local minima and maxima in the signal, so we calculated the IVG for both the original and reverse image. This action improved the overall accuracy by almost five percent.

As far as our knowledge, we performed the first graph-based techniques for classifications of ILD tissues. Compared to others, our result on this complex dataset indicates the high potential of GSP for various tasks like texture classification.

References

[1] Y.-L. Qiao, Y. Zhao, C.-Y. Song, K.-G. Zhang, and X.-Z. Xiang, "Graph wavelet transform for image texture classification," *IET Image Processing*, vol. 15, no. 10, pp. 2372–2383, 2021.

[2] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuno, "From time series to complex networks: The visibility graph," *Proceedings of the National Academy of Sciences*, vol. 105, no. 13, pp. 4972–4975, 2008.

[3] J. Iacovacci and L. Lacasa, "Visibility graphs for image processing," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 974–987, 2019.

[4] A. Depeursinge, A. Vargas, A. Platon, A. Geissbuhler, P.-A. Poletti, and H. Müller, "Building a reference multimedia database for interstitial lung diseases," *Computerized medical imaging and graphics*, vol. 36, no. 3, pp. 227–238, 2012.

[5] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1207–1216, 2016.

[6] A. Dudhane, G. Shingadkar, P. Sanghavi, B. Jankharia, and S. Talbar, "Interstitial lung disease classification using feed forward neural networks," in *International Conference on Communication and Signal Processing 2016 (ICCASP 2016)*. Atlantis Press, 2016, pp. 509–515.

[7] Y. Song, W. Cai, H. Huang, Y. Zhou, D. D. Feng, Y. Wang, M. J. Fulham, and M. Chen, "Large margin local estimate with applications to medical image classification," *IEEE transactions on medical imaging*, vol. 34, no. 6, pp. 1362–1377, 2015.

[8] A. Depeursinge, T. Zrimec, S. Busayarat, and H. Müller, "3d lung image retrieval using localized features," in *Medical Imaging 2011: Computer-Aided Diagnosis*, vol. 7963. International Society for Optics and Photonics, 2011, p. 79632E.

[9] V. Sukanya Doddavarapu, G. B. Kande, and B. Prabhakara Rao, "Differential diagnosis of interstitial lung diseases using deep learning networks," *The Imaging Science Journal*, vol. 68, no. 3, pp. 170–178, 2020.

[10] M. Gao, Z. Xu, L. Lu, A. P. Harrison, R. M. Summers, and D. J. Mollura, "Holistic interstitial lung disease detection using deep convolutional neural networks: Multi-label learning and unordered pooling," *arXiv preprint arXiv:1701.05616*, 2017.

[11] M. Anthimopoulos, S. Christodoulidis, A. Christe, and S. Mougiakakou, "Classification of interstitial lung disease patterns using local dct features and random forest," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 6040–6043.

[12] W. Guo, Z. Xu, and H. Zhang, "Interstitial lung disease classification using improved densenet," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 30 615–30 626, 2019.

[13] S. Huang, F. Lee, R. Miao, Q. Si, C. Lu, and Q. Chen, "A deep convolutional neural network architecture for interstitial lung disease pattern classification," *Medical & biological engineering & computing*, vol. 58, no. 4, pp. 725–737, 2020.

[14] L. Pei, Z. Li, and J. Liu, "Texture classification based on image (natural and horizontal) visibility graph constructing methods," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 1, p. 013128, 2021.