Texture Classification in Interstitial lung disease patients 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*—**

I. Introduction

II. Related work

III. Method

A. *Graph wavelet*

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 [1]. 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 [2] proposed a new version of visibility graphs for images called image visibility graphs.

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,

$$x_k < x_i + \frac{k-i}{j-i}(x_j - x_i), \forall(i < k < j) \tag{1}$$

2) Horizontal Visibility Graphs:

C. *Image Visibility Graphs*

1) INVG: Image natural visibility graph:

2) IHVG: Image Horizontal visibility graph:

3) Feature extraction : discuss about local and global features

IV. Feature selection

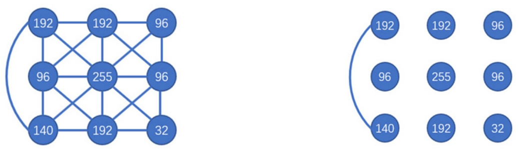
This section will discuss the various features that we can choose for our final feature vector; firstly, we discuss each component individually and then demonstrate the final feature vector.

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 1 shows an example of a texture and its reverse.

(a) Original image (b) Reversed image

Fig. 1: 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 2 is an artifial 3×3 image and in [3] INVG is drawn for with Lattice and with No-lattice condition.



(a) With lattice (b) Without lattice

Fig. 2: An example of INVG with and without lattice [3]

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 [4] 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 total varience of data,this job 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 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 3 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.

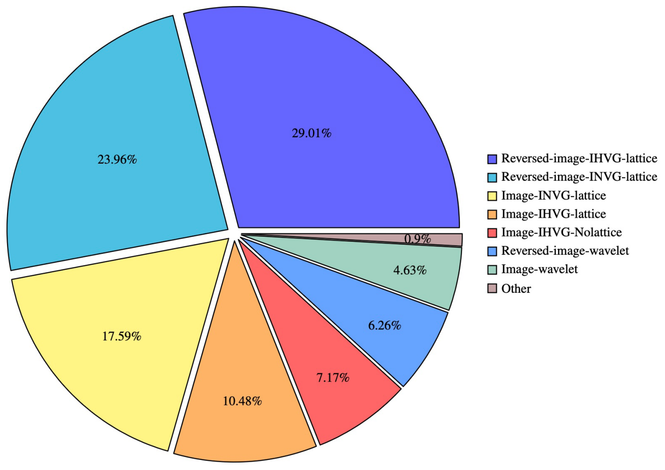


Fig. 3: Importance of each feature vector component based on the variance of all data

V. Experiment

A. *Dataset*

B. *preprocessing & Classifier*

Due to unbalanced data distribution between various classes, we used 10-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 make up 95% of all data variation. Before PCA feature vector length was 1376, and after PCA was 561. We have to use the same transform applied to train data for data preprocess 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. *Results and comparison*

VI. Conclusions

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.