Pyramidal Stochastic Graphlet Embedding for Document Pattern Classification

Anjan Dutta, Pau Riba, Josep Lladós and Alicia Fornés
Computer Vision Center, Computer Science Department,
Universitat Autònoma de Barcelona, Barcelona, Spain
Email: {adutta, priba, josep, afornes}@cvc.uab.es

***Abstract*—Document pattern classification methods using graphs have received a lot of attention because of its robust representation paradigm and rich theoretical background. However, the way of preserving and the process for delineating documents with graphs introduce noise in the rendition of underlying data, which creates instability in the graph representation. To deal with such unreliability in representation, in this paper, we propose Pyramidal Stochastic Graphlet Embedding (PSGE). Given a graph representing a document pattern, our method first computes a graph pyramid by successively reducing the base graph. Once the graph pyramid is computed, we apply Stochastic Graphlet Embedding (SGE) for each level of the pyramid and combine their embedded representation to obtain a global delineation of the original graph. The consideration of pyramid of graphs rather than just a base graph extends the representational power of the graph embedding, which reduces the instability caused due to noise and distortion. When plugged with support vector machine, our proposed PSGE has outperformed the state-of-the-art results in recognition of handwritten words as well as graphical symbols.**

***Index Terms*—graph embedding, hierarchical graph representation, graph clustering, stochastic graphlet embedding, graph classification**

I. Introduction

The problem of document pattern classification is stated as follows: given an image of a word or a graphical symbol, the goal is to predict the class (also referred as keyword) that best describes the visual content of the image. Most of the earlier image pattern classification methods were designed based on some statistical techniques such as BoVW [7], Shape Contexts [2] etc. These statistical methods usually ignore spatial information of the entire pattern. Extensions of this model that somehow integrate spatial information, such as spatial pyramid [16], quickly overtake the baselines. Indeed, it is true that different parts of an image pattern do not appear independently and spatial relationships between these parts are crucial in order to achieve effective image description and classification [19].

Among the existing image and pattern description and classification solutions, those based on graphs are particularly successful [23], this is because of their general principle that consists of modelling local visual features in images as well as their spatial interactions. In these models, images are represented as graphs where nodes correspond to local features and edges describe their spatial and geometric relationships. However, delineating real world documents with graphs involves certain preprocessing techniques which are prone to introduce noise in the rendition of underlying data, resulting in unstable graph representation that further leads to incorrect recognition. In case of documents the scenario is even more challenging, as documents are usually stored for ages before digitization and hence, often suffered by degradation, show through etc. Moreover, these documents also deteriorate due to noise resulting from scanning and superimposition of graphic and textual parts etc.

Graph-based methods have received a lot of interest for document pattern representation and recognition task. At the beginning, the community was more focused on the graphical documents because of the triviality in the graph-based representation of the underlying data. Within graphics recognition, the research is mostly focused on symbol (graphical part) *recognition* and *spotting*. Many researchers have formulated the symbol spotting problem into an inexact subgraph matching problem [3], [9], [17]. They used different graph representation for representing the underlying document and then used some kind of approximated graph matching to solve the problem. Some other authors used indexing based techniques for better efficiency in symbol spotting and retrieval problem [26]. Graph embedding has also caught the attention of the community [18], [21] for the same task. For symbol spotting, hierarchical graph representation has also been addressed to deal with document noise and distortion [4]. On the other hand, frequent subgraph discovery has received a lot of attention for isolated symbol recognition task within the community [1], [8]. They used common substructures, such as, graph paths for defining the similarity between two symbols.

For word recognition and spotting, one of the first methods based on graphs was proposed by Fischer *et al.* [11], where they used graph similarity features as a descriptor for handwriting recognition in historical documents using HMM. For isolated word retrieval problem, Wang *et al.* proposed a skeleton graph representation and an approximate graph edit distance formulation for handwritten word retrieval task [28]. More recently, some other researchers also proposed graph based keyword spotting methods, where they have used variant of approximate graph edit distance based approach to solve the problem [22], [27]. They have also proposed different graph representation to handle noise and distortion as the methods mostly deal with handwritten documents. Despite having a plenty of works that use graph representation for document recognition task, only a very few of them explored the hierarchical representation of graph for tolerating noise, which acts as a motivation in our case.

In this work, we introduce *Pyramidal Stochastic Graphlet Embedding* (PSGE) which tolerates the instability in the graph representation by considering a graph pyramid which

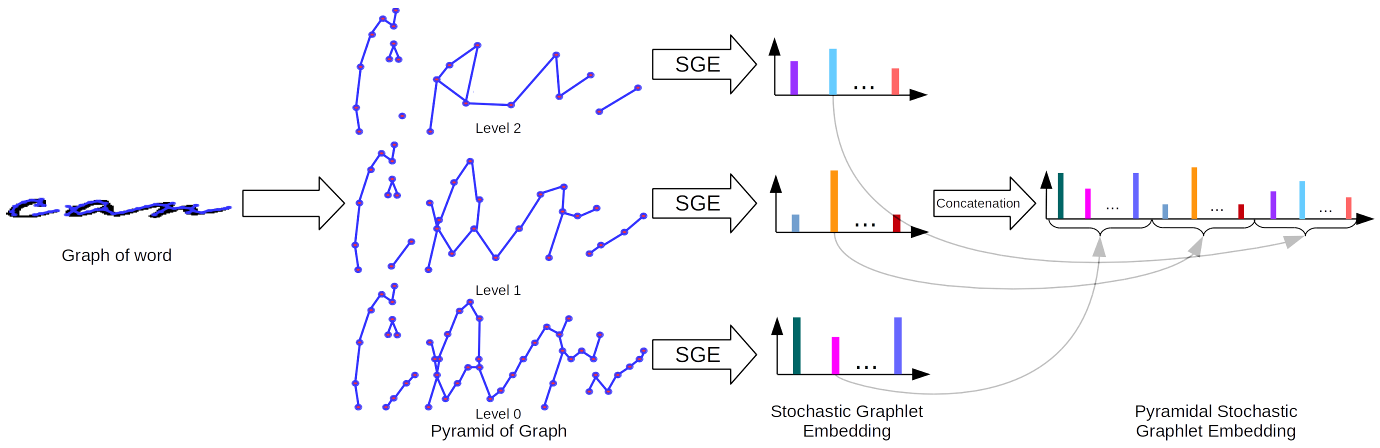

Fig. 1. Overview of our Pyramidal Stochastic Graphlet Embedding framework. Given any graph of a pattern, the method constructs the a pyramid of graph by reducing the size of the base graph successively. We have used two different graph clustering algorithms for the base graph reduction. Stochastic Graphlet Embedding is applied to embed each of the graphs in the pyramid to embed into real vector space. These embeddings of different levels are concatenated to obtain the Pyramidal Stochastic Graphlet Embedding which is used for underlying pattern classification task.

is primarily a collection of graphs all deduced from an original base graph (see Figure 1). Given an attributed graph $G = (V, E, L_V, L_E)$[1], the method first creates a graph pyramid by successively downsampling the original one until some desired point is reached. This is done by clustering the nodes of the base graph by graph clustering algorithms to be described in Section II. Once the graph pyramid is computed, we apply the Stochastic Graphlet Embedding (SGE) [10], described in Section III, to each level of the graph pyramid and concatenate the embedded representations to obtain a global delineation of the graph G. The consideration of graph pyramid instead of a single base graph empowers the representation ability and hence cope with noise, distortions in documents. Thereafter, the embedded representations are plugged into a support vector machine for achieving effective document pattern classification. The main contribution of our work is the consideration of graph pyramid rather than a single original graph for embedding in order to extend the representation power of the embedded graph and tolerate the instability caused due to noise and distortion in case of document patterns. Our proposal is absolutely robust because, on the one hand, it considers the relation between object parts and their complex interactions, and on the other hand, it organizes the structural information in hierarchical abstraction. Additionally, the proposed method is absolutely generic and can adapt any other graph embedding algorithm in the framework, as well as the PSGE can be applied to different types of graphs from diverse application domain.

The rest of this paper is organised as follows: In Section II we present the pyramidal graph representation. Section III describes the *Stochastic Graphlet Embedding* used to embed a graph to high dimensional vector space. Afterwards, in Section IV, we present our experimental validation and compares the proposed method with available state-of-the-art algorithms. Finally in Section V, we conclude the work and future direction of the work is defined.

[1]An *attributed graph* is a 4-tuple $G = (V, E, L_V, L_E)$ comprising a set V of *nodes* together with a set $E \subseteq V \times V$ of *edges* and two *mappings* $L_V : V \rightarrow \mathbb{R}^m$ and $L_E : E \rightarrow \mathbb{R}^n$ which respectively assign attributes to the nodes and edges.

II. Pyramidal Graph Representation

Multi-scale graph representation provides information about structures at different resolutions. Depending on the reduction ratio, the graph becomes more abstract providing new useful information. A *hierarchical graph* H is defined as a 6-tuple $H = (V, E_N, E_H, L_V, L_{E_N}, L_{E_H})$ where V is the set of nodes; $E_N \subseteq V \times V$ are the neighborhood edges; $E_H \subseteq V \times V$ are the hierarchical edges; $\mathrm{L_V}$, $\mathrm{L_{E_N}}$ and $\mathrm{L_{E_H}}$ are three labeling functions defined as $\mathrm{L_V} : V \rightarrow \Sigma_V \times A_V^k$, $\mathrm{L_{E_N}} : E_N \rightarrow \Sigma_{E_N} \times A_{E_N}^l$ and $\mathrm{L_{E_H}} : E_H \rightarrow \Sigma_{E_H} \times A_{E_H}^m$, where Σ_V, Σ_{E_N} and Σ_{E_H} are three sets of symbolic labels for vertices and edges, A_V, A_{E_N} and A_{E_H} are three sets of attributes for vertices and edges, respectively, and $k, l, m \in \mathbb{N}$.

The pyramidal construction of the graph is based on graph clustering techniques [15]. Usually, this methodology has been used in *Social Media Analysis* for *community detection* problems. Two classical clustering frameworks have been compared in the experiments. The first one, uses the idea of the shortest paths between nodes to find and remove the connections between clusters and the second one makes use of the spectral graph theory.

The *Girvan-Newman algorithm* for community detection [14], removes iteratively the edge with highest *betweenness centrality*[2]. At each step of the algorithm, the connected components of the resulting graph are considered as the communities. This algorithm ends up with a dendrogram where at the end, each node is considered as independent cluster. If the desired number of clusters is known, the algorithm can be stopped when the needed number of connected components is reached.

The *grPartition algorithm* presented in [15] is based on the *spectral graph theory*. This methodology, uses a doubly stochastic weighted adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$ which codifies the cost associated to edge removal. Let us assume $k \in \mathbb{N}$ is the number of desired clusters, $U \in \mathbb{R}^{|V|\times k}$ is the matrix defined from the eigenvectors corresponding to the k largest eigenvalues of A in its columns. Finally, *k-means algorithm* is applied to the rows of A assigning a cluster to each node.

Once the clusters have been created, the pyramid is generated by creating a node as representative for each cluster. At this step, hierarchical edges are created linking the clustered nodes to their representative. Moreover, neighbourhood edges must be created for the new nodes. An edge is created if the connection ratio between pairs of clusters is greater than a threshold. Figure 2 shows the resulting pyramid given an input graph using the Girvan-Newman clustering function. Each level of the pyramid reduces by a factor if 1.5 the number of nodes of the previous level. Therefore, each new level provides more abstract information. For visualization issues, only some hierarchical edges (in red) are shown. Graphlet examples of size 5 (in terms of edges, $T = 5$) are drawn on the graph (in green). In more abstract levels, same sized graphlets covers more extension of the graph. Once the graph pyramid is computed, graph from each level of the pyramid is embedded using the Stochastic Graphlet Embedding (SGE) to be explained in the following section.

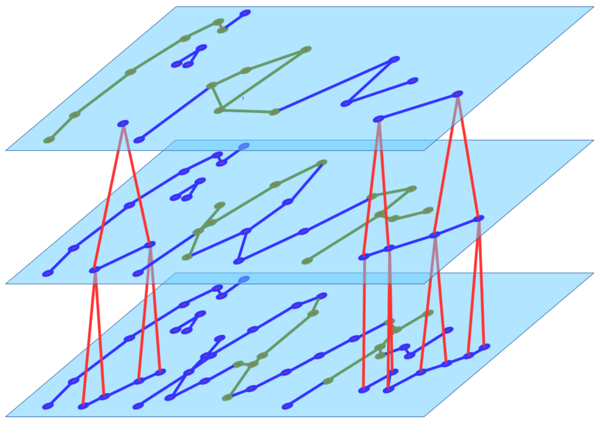

Fig. 2. $3-$level hierarchical graph from the handwritten word "*can*". Each level contracts nodes following the hierarchical edges (in red) to obtain a smaller graph. Graphlet examples of size 5 (in terms of edges, $T = 5$) are drawn on the graph (in green).

III. Stochastic Graphlet Embedding

Stochastic Graphlet Embedding (SGE) can be defined as a mapping $f : \mathbb{G} \rightarrow \mathbb{R}^n$ that explicitly embeds a graph $G \in \mathbb{G}$ to a high dimensional vector space $\mathbb{R}^n$ [10]. The entire procedure of SGE can be described in two stages, where in the first step, the method samples graphlets from G in a stochastic way and in the second step, it counts the frequency of each isomorphic graphlet from the extracted ones in an accurate approximated manner. The entire procedure fetches a precise distribution of connected graphlets with increasing number of edges in G with a controlled complexity, which fetches the relation among information represented as nodes and their complex interaction.

A. Stochastic Graphlets Sampling

Considering a graph $G = (V, E, L_V, L_E)$ that corresponds to a given pattern $\mathcal{P}$, the goal of the graphlet extraction procedure is to obtain a statistics of constituting stochastic graphlets with increasing number of edges in G. The way of extracting graphlets is stochastic and it samples graphlets with unlimitedly increasing number of edges without constraining their topology or structural properties such as maximum degree, maximum number of nodes, etc. The graphlet sampling procedure is recurrent and the number of recurrence is controlled by a parameter M that indicates the number of distinct graphlets to be sampled. Also, each of these M recurrent processes is regulated by another parameter T that denotes the maximum number of iterations a single recurrent process should have. Since each of these iterations adds an edge to the presently constructing graphlet, T indirectly specifies maximum number of distinct edges each graphlet should contain. Considering U_t and A_t respectively as the aggregated sets of visited nodes and edges till iteration t, they are initialized at the beginning of each recurrent step as $A_0 = \emptyset$ and $U_0 = \{u\}$ with a randomly selected node u which is uniformly sampled from V. Thereafter, at t^{th} iteration (with $t \geq 1$), the sampling procedure randomly selects an edge $(u, v) \in E \backslash A_{t-1}$ that is connected from any node $u \in U_{t-1}$. Accordingly, the process updates $U_t \leftarrow U_{t-1} \cup \{v\}$ and $A_t \leftarrow A_{t-1} \cup \{(u, v)\}$. All these processes within a recurrent step are repeated T times to sample a graphlet with maximum T edges. M is set to relatively large values in order to make graphlet generation statistically meaningful. Theoretically, the values of M follows the theorem of *sample complexity* [29], however, the discussion and proof of that is out of scope of the current paper. Intuitively, the graphlet sampling procedure explained in this section follows a random walk process with restart that efficiently parses G and extracts the desired number of connected graphlets with an increasing number of edges. This algorithm allows to sample connected graphlets from a given graph but avoids expensive way of extracting them in an exact manner. Here the hypothesis is that if a sufficient number of graphlets are sampled, then the empirical distribution will be close to the actual distribution of graphlets in the graph. Furthermore, it is important to be noted that from the above process, one can extract, in total, $M \times T$ graphlets each with number of edges varying from 1 to T.

B. Hashed Graphlets Distribution

In order to obtain a distribution of the extracted graphlets G, it is needed to identify sets of isomorphic graphlets from the sampled ones and then count the cardinality of each isomorphic set. A trivial way of doing that certainly involves applying graph isomorphism to all possible pairs of graphlets

[2]The betweenness centrality on an edge $e \in E$ is defined as the number of shortest walks between any pair of nodes that cross e. For further information [12].

for detecting possible partitions exist among them. Nevertheless, graph isomorphism being an NP-complete problem for general graphs, that procedure is extremely costly as the method samples huge number of graphlets with many edges. An alternative, efficient and approximate way of partitioning isomorphic graphlets is *graph hashing*. A graph hash function can be defined as a mapping $h : \mathbb{G} \to \mathbb{R}^m$ that maps a graph into a hash code (a sequence of real numbers) based on the local as well as holistic topological characteristic of graphs. An ideal graph hash function should map two isomorphic graphs to the same hash code as well as two non-isomorphic graphs to two different hash codes. For obtaining a distribution of graphlets, the main aim of graph hashing is to assign extracted graphlets from G to corresponding subsets of isomorphic graphlets (a.k.a. partition index or histogram bins) in order to count and quantify their distributions. For that purpose, a global hash table $\mathbf{H}$ is maintained, whose single entry corresponds to a hash code of a graphlet g produced by the function h. $\mathbf{H}$ grows incrementally as the algorithm confronts new graph hash code and maintains all the unique hash codes encountered by the system. It is to be noted that the position of each unique hash code is kept fixed, because each position corresponds to a partition index or histogram bin. Now to allocate a given graphlet g to its corresponding histogram bin, its hash code $h(g)$ is mapped to the index of the hash table $\mathbf{H}$, whose corresponding graph hash code gives a hit with $h(g)$. If $h(g)$ does not exist in $\mathbf{H}$ at some point, it is considered as a new hash code (and hence g as a new graphlet) encountered by the system and appended $h(g)$ at the end of $\mathbf{H}$.

While it is easy to design hash functions that provide identical hash codes for isomorphic graphlets, it is very challenging to guarantee that non-isomorphic graphlets could never be mapped to the same hash code. The likelihood of mapping two non-isomorphic graphlets to the same hash code is termed as *probability of collision*. Denoting H_0 as the set of all pairs of non-isomorphic graphs, the probability of collision can be expressed as an energy function as below:

$$E(f) = P((g, g') \in H_0 | h(g) = h(g')) \tag{1}$$

So, in terms of collision probability, the hash functions that produce comparatively lower $E(f)$ values in Eqn. (1) are considered to be reliable for checking graph isomorphism. It has been shown that sorted *degree of nodes* has 0 collision probability for all graphs with number of edges less or equal to 4 [10]. Furthermore, it is also a well known fact that two graphs with the same *betweenness centrality* would indeed be isomorphic with high probability [6], [20]. Considering the above facts, in practice, we consider *degree of nodes* for graphlets with $t \leq 4$ and the *betweenness centrality* for graphlets with $t \geq 5$. It is to be noted that these hash functions only consider the topology of the underlying graphlets and the attributes of the graphlets are ignored. However, stochastic graphlet embedding allows to incorporate small set of node and edge attributes to be considered by adding the attribute signatures to the hash code. In our case, the sorted discrete node and edge labels are used as the attribute signatures and combined with the hash code. For any further details on SGE, interested readers are referred to [10]. In summary, it is to be noted that the PSGE is the concatenation of the SGEs of the graphs from different levels of the graph pyramid, which is plugged into SVM for subsequent document pattern classification.

IV. Experimental Validation

For evaluating the impact of our proposed Pyramidal Stochastic Graphlet Embedding algorithm on document patterns, we consider two benchmark datasets as described in the following section.

A. Datasets

GREC: The GREC dataset is a part of the IAM graph repository[3] [24] and consists of 1100 graphs representing 22 different architectural and electronic symbols each having 50 instances with different levels of noise. Therefore the task of the classifier is to predict the class of the graphs representing different symbols, which will be one of the 22 architectural or electronic symbols. These graphs are node and edge labelled, where node position and type (whether junction, intersection etc.) are used as node attributes and the connection type (whether line, arc etc.) between two nodes are used as edge label. The entire dataset is split into train, validation and test set, where the train and validation sets contain 286 graphs of symbols, and the test set contains 528 graphs.

HistoGraph: The HistoGraph dataset[4] [27] consists of graphs representing words from the communicating letters written by the first US president, George Washington. It consists of 293 graphs generated from 30 distinct words. Therefore, given a word, the task of the classifier is to predict its class which should be among the 30 words. Nodes are only labelled with their position in the image. Furthermore, this dataset used 6 different graph representation paradigms for delineating a single word into graph, which results in 6 different subsets of graphs. The entire dataset is divided into 90, 60 and 143 graphs respectively for train, validation and test purposes. Summary of both the datasets is presented in Table I and an example graph from both the datasets are shown in Figure 3.

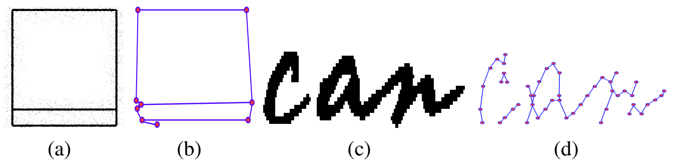

Fig. 3. (a) a graphical symbol from GREC dataset, (b) graph representation of the graphical symbol in (a), (c) a handwritten word from HistoGraph dataset, (d) graph representation of the handwritten word in (b). The spurious nodes on the corresponding graph representation are to be noted, which gives an intuitive idea about the instability in representation when a pattern is transformed to graph.

[3]Available at http://www.fki.inf.unibe.ch/databases/iam-graph-database
[4]Available at http://www.histograph.ch/

TABLE I
DETAILS ON GREC AND HISTOGRAPH DATASETS

Datasets	Subset	#Graphs	#Classes	Avg.$\|V\|$	Avg.$\|E\|$	Node labels	Edge labels
GREC	-	1100	22 (50 each)	11.5	11.9	Type of join and (x,y) position	Type of edge
HistoGraph	Keypoint	293 (90, 60, 143 for train, validation and test)	30	73	67	(x,y) position	-
	Grid-NNA			39	55		
	Grid-MST			46	44		
	Grid-DEL			52	138		
	Projection			44	41		
	Split			51	48		

B. Evaluation

Table II shows the parameters involved in our proposed Pyramidal Stochastic Graphlet Embedding. The construction of graph pyramid involves four parameters: (1) the graph clustering algorithm, (2) reduction rate, (3) pyramidal levels, and (4) edge threshold. We have experimented with two different graph clustering algorithms: (1) *Girvan-Newman* and (2) *grPartition*, and we have noticed that both the algorithms obtained similar results on both the datasets. For example, on GREC dataset, the highest accuracy obtained with two levels of graph pyramid has been obtained both with Girvan-Newman as well as grPartition. This fact is equally true for the HistoGraph dataset. Similarly, we have also observed that the reduction rate and the edge threshold neither controls the classification accuracy of the method, whereas, the number of pyramidal levels regulates the performance of the system quite prominently. It can be noted from the experimental results (shown in Table III and Table IV) that consideration of more pyramidal levels often improves the performance of the system on both the datasets. Equivalently, the stochastic graphlet embedding employs two parameters: (1) M that indicates the number of distinct graphlets to be sampled and (2) T maximum number of edges a single graphlet should possess. In general, very low values of M result in inadequate graphlet distribution, hence, also reduce classification accuracy. Therefore, M should necessarily be large enough, however, increasing M after a certain limit does not improve the performance. To be specific, all the experiments reported in this paper are done by setting $M = 46,000$. Further on, high values of T involves bigger graphlets that usually capture richer structural information. Therefore, commonly, increasing the value of T increases the performance but considering large T increase the number of iteration, and hence raise the computational complexity of the method. In graph pyramid, since the graphs in higher level reduce sizes, only for the base graph we use $T = 7$ and for all others in the graph pyramid, we use $T = 5$.

Table III shows the results of our proposed PSGE and comparison with the state-of-the-art methods on the GREC dataset. Both for SGE and PSGE, we have computed the accuracies both on the labelled as well as on the unlabelled GREC dataset. It is to be noted that the performance of both methods has improved on the labelled graphs compared to the unlabelled ones, which is well justified because attributes always provide richer information about the data. Both on labelled and unlabelled data, the PSGE have improved the accuracies with respect to the SGE, which shows the effectiveness of our proposal. Adding a third level, also improves the accuracy obtained.

TABLE II
SUMMARY OF PARAMETERS USED IN THE PROPOSED METHOD.

	Parameter	Values
Pyramid	Clustering Algorithm	Girvan-Newman grPartition
	Reduction Rate	1.5, 2
	Pyramidal Levels	1, 2, 3
	Edge Threshold	0, 0.25
SGE	M	$46,000$
	T	7

TABLE III
CLASSIFICATION ACCURACY FOR GREC DATASET. GAINS ARE SHOWN WITH RESPECT TO SGE.

Method	Unlabelled	Labelled	
Dissimilarity Embedding [5]	-	95.10	
Node Attribute Statistics [13]	-	99.20	
Fuzzy Graph Embedding [18]	-	97.30	
SGE [10]	92.80	99.62	
		Level 2	Level 3
PSGE	93.18 (+0.38)	99.62 (+0.00)	**99.81 (+0.19)**

Table IV shows the results of our proposed PSGE on all the subsets of the HistoGraph dataset. Here we have compared our results with the graph edit distance (GED) [25] and the SGE [10]. On this experiment as well, we utilized two levels of the graph pyramid. Similar to the previous experiment, here also, it is to be noted that the proposed PSGE improves the performance accuracies in most of the cases, which justifies our proposal. Furthermore, we have observed that successive increment in the number of levels successively increases the performance for most of the subsets. This proves that the increase of levels in the graph pyramid adds further discrimination power in the graphs. However, redundant amount of abstraction could cause noisy embedding.

V. CONCLUSIONS

In this paper we have proposed Pyramidal Stochastic Graphlet Embedding and used it for document pattern classification problem. The method constructs a graph pyramid by successively reducing the base graph, which is done by clustering the nodes of the graph at a certain level. After having the graph pyramid, Stochastic Graphlet Embedding is applied to each graph of the pyramid and combined to give a global

TABLE IV
CLASSIFICATION ACCURACY FOR HISTOGRAPH DATASET. GAINS ARE SHOWN WITH RESPECT TO SGE.

Subset	**Acc. GED**	**Acc. SGE**	**Acc. PSGE**	
			Level 2	Level 3
Keypoint	77.62	78.32	**80.42 (+2.10)**	78.32 (+0.00)
Grid-NNA	65.03	72.73	72.73 (+0.00)	**74.13 (+1.40)**
Grid-MST	74.13	**76.92**	75.52 (-1.40)	74.83 (-2.09)
Grid-DEL	62.94	74.83	**79.02 (+4.19)**	**79.02 (+4.19)**
Projection	**81.82**	79.02	79.72 (+0.70)	80.42 (+1.40)
Split	80.42	77.62	**80.42 (+2.80)**	77.62 (+0.00)

representation which is used for classification. Our proposed PSGE considers a set of hierarchical graphs rather than just the original graph to empower the embedded representation. We have shown the performance of our proposed PSGE on two different benchmarks and have surpassed the available state-of-the-art results on most of them, which clearly showed the effectiveness of the proposal.

Hierarchical edges in the graph pyramid certainly contain structural information of the original graph. Moreover, studying the propagation of node attributes through the hierarchical levels of graph pyramid will also be interesting. All these are the promising lines to be investigated in future.

ACKNOWLEDGMENTS

This work has been partially supported by the European Union's research and innovation program under the Marie Skłodowska-Curie grant agreement No. 665919, Spanish project TIN2015-70924-C2-2-R, a FPU fellowship FPU15/06264 from the Spanish Ministerio de Educación, Cultura y Deporte, the Ramon y Cajal Fellowship RYC-2014-16831, and the CERCA Program/Generalitat de Catalunya.

REFERENCES

[1] E. Barbu, P. Héroux, S. Adam, and E. Trupin, "Frequent graph discovery: Application to line drawing document images," *ELCVIA*, vol. 5, no. 2, pp. 47–54, 2005.

[2] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE TPAMI*, vol. 24, no. 4, pp. 509–522, 2002.

[3] P. L. Bodic, P. Héroux, S. Adam, and Y. Lecourtier, "An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol spotting in technical drawings," *PR*, vol. 45, no. 12, pp. 4214–4224, 2012.

[4] K. Broelemann, A. Dutta, X. Jiang, and J. Lladós, "Hierarchical graph representation for symbol spotting in graphical document images," in *S+SSPR*, 2012, pp. 529–538.

[5] H. Bunke and K. Riesen, "Improving vector space embedding of graphs through feature selection algorithms," *PR*, vol. 44, no. 9, pp. 1928 – 1940, 2010.

[6] F. Comellas and J. Paz-Sánchez, "Reconstruction of networks from their betweenness centrality," in *AEC*, 2008, pp. 31–37.

[7] G. Csurka, C. Dance R., L. Fan, J. Williamowski, and C. Bray, "Visual categorization with bags of keypoints," in *SLCVW, ECCV*, 2004, pp. 1–22.

[8] A. Dutta, J. Lladós, and U. Pal, "Bag-of-graphpaths for symbol recognition and spotting in line drawings," in *GREC*, 2011, pp. 208–217.

[9] A. Dutta, J. Lladós, and U. Pal, "A symbol spotting approach in graphical documents by hashing serialized graphs," *PR*, vol. 46, no. 3, pp. 752–768, 2013.

[10] A. Dutta and H. Sahbi, "High order stochastic graphlet embedding for graph-based pattern recognition," *CoRR*, pp. 1 – 14, 2017.

[11] A. Fischer, K. Riesen, and H. Bunke, "Graph similarity features for hmm-based handwriting recognition in historical documents," in *ICFHR*, 2010, pp. 253–258.

[12] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[13] J. Gibert, E. Valveny, and H. Bunke, "Graph embedding in vector spaces by node attribute statistics," *PR*, vol. 45, no. 9, pp. 3072 – 3083, 2012.

[14] M. Girvan and M. Newman, "Community structure in social and biological networks," *NAS*, vol. 99, no. 12, pp. 7821–7826, 2002.

[15] J. P. Hespanha, "An efficient matlab algorithm for graph partitioning," *Santa Barbara, CA, USA: University of California*, 2004.

[16] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006, pp. 2169–2178.

[17] J. Lladós, E. Martí, and J. J. Villanueva, "Symbol recognition by error-tolerant subgraph matching between region adjacency graphs," *IEEE TPAMI*, vol. 23, no. 10, pp. 1137–1143, 2001.

[18] M. M. Luqman, J.-Y. Ramel, J. Lladós, and T. Brouard, "Fuzzy multilevel graph embedding," *PR*, vol. 46, no. 2, pp. 551–565, 2013.

[19] N. Morioka and S. Satoh, "Compact correlation coding for visual object categorization," in *ICCV*, 2011, pp. 1639–1646.

[20] M. J. Newman, "A measure of betweenness centrality based on random walks," *SN*, vol. 27, no. 1, pp. 39 – 54, 2005.

[21] R. Qureshi, J.-Y. Ramel, D. Barret, and H. Cardot, "Spotting symbols in line drawing images using graph representations," in *GREC*, 2008, pp. 91–103.

[22] P. Riba, J. Lladós, and A. Fornés, "Handwritten word spotting by inexact matching of grapheme graphs," in *ICDAR*, 2015, pp. 781–785.

[23] P. Riba, J. Lladós, A. Fornés, and A. Dutta, "Large-scale graph indexing using binary embeddings of node contexts for information spotting in document image databases," *PRL*, vol. 87, pp. 203 – 211, 2017.

[24] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," in *S+SSPR*, 2008, pp. 287–297.

[25] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *IVC*, vol. 27, no. 7, pp. 950 – 959, 2009.

[26] M. Rusiñol, A. Borràs, and J. Lladós, "Relational indexing of vectorial primitives for symbol spotting in line-drawing images," *PRL*, vol. 31, no. 3, pp. 188–201, 2010.

[27] M. Stauffer, A. Fischer, and K. Riesen, "A novel graph database for handwritten word images," in *S+SSPR*, 2016, pp. 553–563.

[28] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Lladós, and A. Fornes, "A novel learning-free word spotting approach based on graph representation," *DAS*, pp. 207–211, 2014.

[29] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, "Inequalities for the l1 deviation of the empirical distribution," HP Labs, Palo Alto, Tech. Rep., 2003.