



HAL
open science

Learning embeddings for cross-time geographic areas represented as graphs

Margarita Khokhlova, Nathalie Abadie, Valérie Gouet-Brunet, Liming Chen

► **To cite this version:**

Margarita Khokhlova, Nathalie Abadie, Valérie Gouet-Brunet, Liming Chen. Learning embeddings for cross-time geographic areas represented as graphs. SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Mar 2021, Virtual Event Republic of Korea, South Korea. pp.559-568, 10.1145/3412841.3441936 . emse-03548148

HAL Id: emse-03548148

<https://hal-emse.ccsd.cnrs.fr/emse-03548148>

Submitted on 29 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning embeddings for cross-time geographic areas represented as graphs

Full Research Paper

Margarita Khokhlova

LASTIG, Univ Gustave Eiffel, ENSG, IGN, France

LIRIS/Lyon Centrale

Saint-Mande, France

margarita.khokhlova@ign.fr

Nathalie Abadie

LASTIG, Univ Gustave Eiffel, ENSG, IGN

Saint-Mande, France

nathalie-f.abadie@ign.fr

Valérie Gouet-Brunet

LASTIG, Univ Gustave Eiffel, ENSG, IGN

Saint-Mande, France

valerie.gouet@ign.fr

Liming Chen

LIRIS/Lyon Centrale

Ecully, France

ABSTRACT

Geographic entities from the vertical aerial images can be viewed as discrete objects and represented as nodes in a graph, linked to each other by edges capturing their spatial relationships. Over time, the natural and man made landscape may evolve and thus also their graph representations.

This paper addresses the challenging problem of the retrieval and fuzzy matching of graphs to localize near-identical geographical areas across time. Several use-case scenarios are proposed for the end-to-end learning of a graph embedding using Graph Neural Networks (GNN), along with an effective baseline without learning.

The results demonstrate the efficiency of our approach, that enables efficient similarity reasoning for novel hand-engineered cross-time graph data. Code and data processing scripts are available online ¹.

CCS CONCEPTS

• **Computing methodologies** → **Feature selection; Image representations;** • **Information systems** → **Image search;**

KEYWORDS

Geospatial descriptors, research and retrieval of historical data, graph embedding, contrastive learning, GCN

ACM Reference Format:

Margarita Khokhlova, Nathalie Abadie, Valérie Gouet-Brunet, and Liming Chen. 2021. Learning embeddings for cross-time geographic areas represented as graphs: Full Research Paper. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of*

¹<http://alegoria.ign.fr/benchmarks>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8104-8/21/03...\$15.00

<https://doi.org/10.1145/3412841.3441936>

Korea. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3412841.3441936>

1 INTRODUCTION

The cross-time aerial image matching problem is very challenging as landscape and appearance changes through time can be dramatic. However, the structural and spatial information about the man-made and natural objects and their spatial relations derived from images can be more persistent than the appearance-based one. Topographic databases are an incredible source of information, from which semantic cues may be distilled. Recently with the progress of CNN architectures for segmentation the task of fully automated scene segmentation became possible [21, 40]. Semantic maps are an additional source of information that can potentially improve geo-localisation, cross-view and cross-time retrieval [22]. Among these, there are numerous detailed maps available for the French territory, from the earlier dates up until today. In this work, we follow the intuition that this rich source of semantic information can be leveraged to improve geographic areas matching and retrieval through time. In order to test this hypothesis, we propose to use already existing topographic databases as sources of semantic annotations. The aimed task requires many steps shown schematically in Figure 1, which can be considered as independent research questions, and in the current work we thus concentrate on the problem of graph matching and retrieval across time. We start from already existing topographical databases captured in the same year as the aerial photograph that we want to match, and that are geometrically consistent with their corresponding photographs, represent these data as graphs and propose a Graph Learning CNN model to match the areas across years.

By using mainly the structural and spatial information about the scene, we aim to propose a new approach which has a potential to be more robust than image-based similarity search. By explicitly modeling relationships between geographic entities of several categories from topographic databases as the graph entities, we demonstrate how the well-known image-based retrieval task can be addressed in a new manner, benefiting from the latest advances in graph learning. Graph-based data structures are commonly used in many application domains, as diverse as social

media, biology, natural language processing, etc. Common types of graphs in graph learning are social networks, knowledge graphs representing different types of data, graphs representing molecular data, graph-structured parses for natural language understanding, and finally 3D meshes. Recent works [24, 25, 30] have demonstrated the efficiency of graph learning on various tasks.

One of the main concerns in this work is to develop an accurate, robust to noise, and time-efficient learning process for the embedding of geographic information. We start from already existing topographical databases captured in the same year as the aerial photograph that we want to match, and that are geometrically consistent with their corresponding photographs. Then the geographic features are engineered manually as detailed in Section 4. The geometric features are used in both the unsupervised baseline methods and our proposed Graph Convolutional Network (GCN) based algorithm. The first unsupervised method is based on an approximate nearest neighbour similarity search tool [20]. In the second step, the commonly used graph kernels [39] were evaluated on our new dataset to complete the baseline study. We further demonstrate how to use GCN architectures to embed geographic entities graphs into a vector space, and learn this embedding model to make same geographic area graphs close to each other in the vector space, and different geographic area graphs far apart. One important property of our model is that it maps each graph independently to a global embedding vector, and then all the similarity computation happens in the vector space, which can reduce the search time.

To summarize, the contributions of this paper are: (1) we propose a new hand-engineered cross-temporal dataset for geographic areas matching; (2) we demonstrate how using the geometrical attributes of the scene landmarks we can retrieve the same geographic areas across time; (3) we propose a siamese-like GCN model to learn graph embeddings for two cross-time retrieval scenarios and established several hand-engineered baselines for future works; (4) we do an extensive evaluation of the parameters and model through an ablation study and perform a statistical analysis of the dataset.

2 RELATED WORK

2.1 Structured Data Learning

The similarity between a pair of graphs representing a geographic area can be calculated in multiple ways: graph kernels, graph embeddings, graph distances [8], and lately by graph neural networks trained for this task.

Graph Kernels. Similarity search based on graph kernels is a well-known research subject with a great number of various kernels proposed for the various specific cases and data types, many are available in Grakel library [39]. The similarity is typically defined by either exact matches (full-graph or sub-graph isomorphism) [26], random walks or paths on graphs [44], propagation of the information in the graph structure [34] or others. A recent survey on graph kernels can be found in [27]. It should be noticed that the kernels themselves are hand-designed and motivated by graph theory, and only some of them are designed to handle continuous attributes on edges and nodes of a graph. Graph kernels can be formulated as first computing the feature vectors for each graph, and then taking the inner product between these vectors to compute the kernel value, no learning involved with an exception of [49].

Graph kernels have shown themselves as very efficient tools for graph comparison but often take a significant time to compute.

Graph Distances. Learning a distance metric between graph nodes is the key focus of the area of metric learning. Common choices include spectral distances and distances based on node affinities. [45] compares commonly used graph metrics and distance measures, and demonstrate their ability to discern between common topological features found in both random graph models and real-world networks. Many of the classical graph kernels are also based on the graph distances [5, 39].

Graph Embeddings. Only relatively recently different Machine Learning and Deep learning algorithms were proposed for graph data. The data mining community has a strong interest in (knowledge) graph summarization because graph structure is ubiquitous: all kinds of data from social networks and up to research publications can be represented as graphs. A popular idea is to learn the embeddings for nodes [16] or even the whole graphs [2, 33] based on their features and structure. However, all these algorithms are based on the models coming from the text processing, so they were designed to produce the embeddings such that nodes with similar network neighbourhoods are embedded close together: the nodes are handled as words taken from a pre-defined vocabulary. Moreover, these methods can handle structure or label info but not both in the same time, which limits their application for our scenario.

Geographic Data. Graph-based representations of the geographic territory are traditionally used for networks representation and their associated computations such as routing applications. But graph based representation of places and landscapes can also reveal important insights in certain scenarios such as scene geolocalization, or geographic information retrieval.

Recently, geographic knowledge graphs have gained attention as they link geographic entities with entities from life sciences, linguistic domain, media, social networks, and various user-generated contents on the Web. [35] proposes a set of metrically refined approximate topological relations to enrich a geographic knowledge graph and improve its questions answering capabilities. Geographic knowledge graphs summarization is largely covered in [47]. The work covers topics such as understanding, representing, and reasoning about Points Of Interest (POI) but does not propose the ways to learn geographic area descriptors. Yan et al. [48] learn Points of Interests (places) semantic similarity from a topographic database. However, the authors try to capture the similarity between general pre-defined object types, when we aim to distinguish them to geolocalize the area. They also use the neighbourhood of the POIs based on the distance and not specifically the graph structure. Finally, Trisedya et al. [41] propose an entity alignment model for knowledge graphs based on the earlier Trans-E approach [4] which models relationships by interpreting them as translations operating on the low-dimensional embeddings of the entities. However, the method requires common unique geographical objects between two graph databases and a fixed set of relations between the nodes, so it cannot be used in our scenario.

The idea of using different distinct geographic objects to automatically geolocalize and match image is not novel. Early research in Computer Vision envisaged this long time ago [17], and there are recent works for image-based localisation [12, 19]. However,

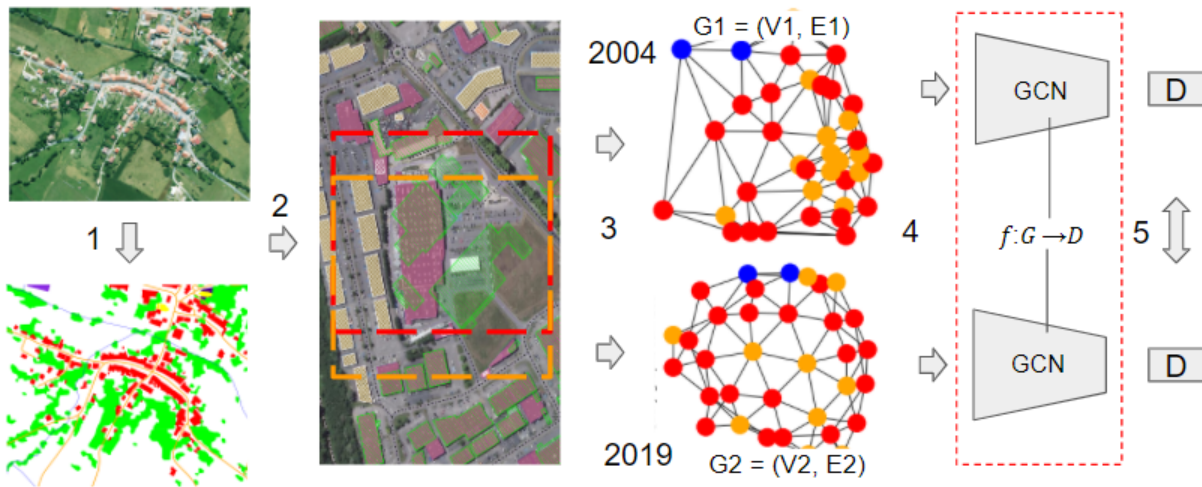


Figure 1: The proposed approach consists of several steps: 1) manual labeling of the aerial images before this work; 2) selection of the POI and surrounding areas for cross-time data; 3) representation of these geographic entities as connected graphs; 4) GCN-based embedding learning with shared weights; 5) research and retrieval of geographic areas across time.

these works, don't explicitly use the graph structure and do not aim for cross-time matching as we do in this research.

Convolutional Graph Networks. In the past few years graph neural networks (GNNs) have emerged as an effective class of models for learning representations of structured data and for solving various supervised prediction problems on graphs. Such models are invariant by design to permutations of graph elements and compute graph node representations through a propagation process which iteratively aggregates local structural information. Nodes on isomorphic graphs (with the same node and edge features) will have the same representations regardless of the ordering. GNNs have different architectures and can be roughly classified into several categories. Spectral methods [7] perform graph convolution by employing the eigen vectors of the graph Laplacian as the transformation matrix, methods that work in the spatial domain [24] and methods complementary to GNNs and agnostic to the choice of a GNN itself (i.e. pooling, attention) [42]. GNNs have been successfully used in many domains from drug discovery [43] to social network classifications [29]. Independently on the network nature, the common task accomplished by them is the supervised learning of the node embeddings. These node representations are then used directly for node classification, or pooled into a graph vector for graph classification. Problems beyond supervised classification or regression are relatively less well-studied for GNNs. Xu et al.[46] proved that with CNNs we can measure the graph's similarity similar to Weisfeiler-Lehman similarity test. However, as it is shown in the next section of our work, graph isomorphism problem is not very relevant for our type of data. More general graph similarity learning approaches were recently proposed by [3, 30]. These learned models can adapt to the desired metric and are potentially interesting for our target scenario. However, Li et al. demonstrate the performance of the method on graphs with no attributes and only minor structural changes. In this paper, we focus on representation and similarity metric learning for attributed graphs representing the near-identical geographic area across time.

2.2 Siamese Networks.

Siamese network architectures aim to construct an embedding, where two extracted features corresponding to the same identity are more likely to be closer than features from different identities [6]. They are a popular choice for scenarios dealing with so-called one-shot learning problems when a single training sample is available for each class. The efficiency of Siamese networks was previously demonstrated for visual object tracking [28], person identification [10], cross-view image matching [18] and other tasks. Siamese networks can also be used for graph similarity learning as it was demonstrated in [30]. The closest to ours is the recent work of [8] where the authors successfully use a Siamese Graph Convolutional architecture for research and retrieval of remote sensing (RS) images represented as region adjacency graphs (RAG).

We follow a similar idea to use the descriptive power of graph representation along with a Siamese-based GCN, however, the graph creation process differs from RAG approach, and the architecture we propose is conceived for our type of data and corresponding features. The final scenario also differs: we want to retrieve the exact location and not the similar classes, hence we deal with a more challenging problem with many classes and a few examples per class (mostly, a single correspondence). Moreover, our end goal is to make an image to vector data² correspondence.

3 PROBLEM DEFINITION.

The semantic information available in the topographic data and their geometries can be leveraged to create scene relational graphs $G(V, E)$, where V is the graph's set of nodes and E its set of edges. In the graph representing a given geographic area, each geographic entity can be represented as a node v with attributes X describing its properties (e.g. its geographic entity type, its name, its area, etc.). Edges represent spatial relationships between nodes; they may also have attributes, although we do not use this possibility in this work. For a given geographic area, the graph representing

²Structured geographic data representing the location and shape of geographic entities with vector geometries.

its spatial configuration can thus be summed up by the following equation:

$$G_e = (R_e, X, A) \quad (1)$$

where R_e is the reference area, X is the set of values associated with all nodes, so-called node features and $A \in \mathbb{R}^{N \times N}$ is the Adjacency matrix to encode the relational information between all nodes. The reference area R_e is a generic term that defines the geographic area covered by the graph.

Given the subsets of graphs representing parts of the territory through the years, we want to do the cross-time aerial image matching based on the structural and spatial information about the scene. Hence we want to learn an embedding of the graph structure, which can take into account the node attributes and structure relationships between the nodes, be robust to the noise and changes in the data, and be compact so it can be later used for the fast search and retrieval in a large database of thousands of graphs.

4 DATASET

The dataset originates from the French Mapping Agency (IGN) [1] and contains graphs derived from semantic annotations of vertical aerial images taken from three French regions (Moselle, Bas-Rhin, and Meurthe-and-Moselle) in four different years: 2004, 2010, 2014 and 2019. We provide the code designed for the graph creation from the vector data³ and the data⁴.

Graph formation. We selected the geographic areas for the database so that each graph contains at least one so-called point of interest (POI) - a building from the following categories: a building of religious nature, historical objects and monuments, castles or forts, local governmental buildings, buildings with the sport functions, railway stations, airports. In our geographic dataset, each central POI entity e has a geospatial context GC_e represented as a graph: Each node in V represents a single geographical entity. The relational information is represented by the edges E of the graph, which are determined by building a Delaunay triangulation [13] between the centroids of nodes geometry. This method was selected to guarantee the formation of connected graphs.

Graph nodes have geometric attributes X (normalized perimeter, x_1 and eccentricity, x_2) and a discreet label, l_n , representing the node's nature (fifteen in total: river, road, railroad, religious building, castle, fort, tower, arc, monument, cemetery, sports ground, normal building, cemetery, public building, airport). Many other geometrical attributes are commonly used in geo-spatial studies such as general orientation, mean axes of the geometric forms, surface descriptors, various shape descriptors [32]. However, we limited this research to the simplest ones which do not require any orientation information nor a high level of details to compensate for the different levels of details in the annotations across time and to not make any assumptions on the orientation of the scene. Eccentricity in our case is simply $E = \frac{L}{W}$, where L and W are the length and width of the minimal geometry's bounding box. Normalized perimeter is simply $P_n = \frac{P}{H \times W}$. Note that we do not use the edge attributes because we wanted our graphs to be rotation and scale-invariant: hence, the angles between objects cannot be

year	# nodes	# edges	node degree
	av, st, min, max	av, st, min, max	av, st, min, max
2004	29.51, 15.16, 3, 91	76.78, 43.40, 3, 256	5.20, 1.25, 2, 12
2010	30.18, 15.39, 3, 92	78.73, 44.09, 3, 260	5.22, 1.25, 2, 11
2014	65.83, 53.25, 3, 150	183.99, 156.97, 3, 280	5.59, 1.43, 2, 22
2019	37.70, 19.59, 3, 125	100.62, 56.54, 3, 356	5.34, 1.26, 2, 14

Table 1: Single department statistics example

department	# graphs	# unique areas	years
Moselle	2768	2298	2004, 10, 14, 19
M-et-Moselle	1968	1571	2004, 10, 14, 19
Bas-Rhin	2268	1863	2004, 10, 14, 19
Côtes-d'Armor	5029	5029	2004
total	12033	10761	4 time frames

Table 2: The characteristics of the proposed dataset for cross-time geographic area retrieval. Note that we mostly deal with single geographic area correspondence across years.

used as the edge weights in our scenario. The graphs in our dataset are therefore undirected and unweighted.

An example of the resulting graph representing the near-identical area across time is shown in Figure 2. Note the difference in the corresponding landscape and graph structure within the two corresponding dates with a 15 years gap, especially in terms of the roads. We observed that in some cases the changes are very significant.

Apart from using the three departments with matching geographic areas, we also added some clutter data without correspondences from a fourth French department, namely the Côtes-d'Armor, to make the research and retrieval scenario more challenging. Table 2 presents the data in the graph database we developed for the geographic area retrieval across time. Note that the maximum number of nodes in the graph is 150. This was done intentionally, we just removed several geo areas with a bigger number of vertices present in the data to limit the final graph size. Similarly, we removed all the graphs with less than 3 nodes. This choice is explained by the selection of the GCN model later in this work.

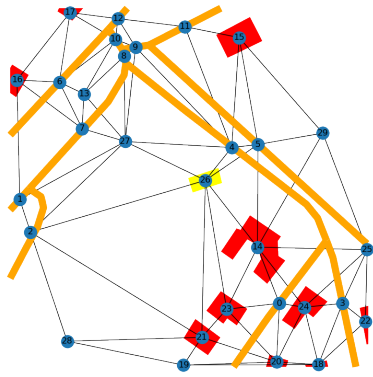
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Data statistics. An example of final graph data characteristics and distribution for Moselle department is summarized in Table 1. It is interesting to see the difference between the data distribution across the years. Note also that there is a particular change in the number of nodes and edges in 2014 - this is probably due to the change in the manual annotation process since we use the same code to convert the vector data to graph representations for all the years. We used the same procedure to create graphs for all the years.

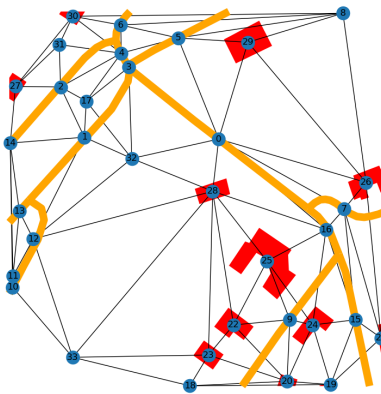
As explained above, our dataset does not always contain the exact same geographic areas (i.e the bounding boxes are shifted for a more realistic scenario) each year, we provide the statistical analysis of the similarity of the attributes between matching graph data.

³https://github.com/margokhokhlova/geomaps_with_pandas

⁴<http://alegoria.ign.fr/benchmarks>



(a) 2004 graph



(b) 2019 graph

Figure 2: Graphs (in blue) representing a geographical area in two time points. The detailed geometries are shown for the reference, their categories are color-coded: red = building, orange = road segment, yellow = building from a special category (such as church, monument, castle, etc).

Intersection over Union (IOU) between two graphs representing the same geographic area across time is used:

$$IOU_{g_1, g_2} = \frac{2 * \sum e_{X_{g_1}} == e_{X_{g_2}}}{\sum e_{X_{g_1}} + e_{X_{g_2}}} \quad (2)$$

where X_{x_1}, x_2 are geometric attributes of the nodes representing geographical entities e . The resulting distribution of the IOU between matching graphs are shown in Figure 3. The obtained distribution is not always normal, however, analyzing the resulting histograms, we can see that there the number of graphs with a smaller values of IOU is bigger when the difference in time is bigger, which seems to be logical.

5 BASELINES

We propose several baselines to evaluate the scenario at hand and the performance of the existing graph matching approaches. To

query year	db year	map@5	t per q (sec)
2019	2004	0.827	0.05
2014	2004	0.835	0.05
2010	2004	0.924	0.04

Table 3: map@5 Faiss similarity search results, database contains clutter graphs.

form our no-learning baseline, we selected two approaches: (1) local-based descriptors retrieval and (2) graph kernels.

For all experiments unless stated otherwise, we use the graphs from 3 departments as queries to retrieve corresponding data from the database dated another year and containing the 4th department clutter data. We employ the K Nearest Neighbours (KNN) to retrieve top 5 matching results and report map average precision value (map@5): $map@K = \frac{\sum_{n=1}^N P_{av}@K}{N}$, where N is number of queries, P_{av} is the average precision for a single query, $K=5$.

No learning similarity search. The first set of experiments is configured to find out whether sole geometric attributes of the scene objects are enough for the graph matching task, without using any relational information (i.e. graph representation) and any learning.

We use Facebook AI Similarity Search (Faiss) library [20] to retrieve the geographic areas across time. Faiss is designed to search for multimedia documents that are similar to each other using the KNN algorithm. We use the L_2 distance measure to retrieve the most similar geographic areas across years based on the local geometric features and semantics of each object present in the scene. Other similarity metrics available in Faiss proved to be less performant experimentally.

The results of the Faiss-based similarity search are summarised in Table 3. Obtained map@5 scores are quite high, which means that the geometrical attributes are representative enough to describe the geographical areas. Nevertheless, there is still room for an improvement that can potentially be gained by using the relational information between the nodes, i.e the graph representation. It is interesting to see when the returned data are wrong. Figure 4 demonstrates the retrieval results for non-correct matches. Note that even if the actual correspondence geographic area from 2004 contains many entities present in 2019, the other areas were returned instead. This example shows the limitation of the attributes-only search when no relational information about the scene is used. At the same moment, we can see that there are also significant graph structure modifications across years which we ideally want to be robust to in the graph matching scenario.

Graph Kernels. We selected several popular graph kernels which were designed for graphs with discrete and continuous attributes to evaluate their performance on our geographical data. We tested numerous kernels, but here we provide only the results for the most interesting ones. All graph kernels we used are implemented in Grakel library [39]. The subgraph matching kernel counts the number of matchings between subgraphs of bounded size in two graphs [26]. The neighbourhood subgraph pairwise distance kernel extracts pairs of rooted subgraphs from each graph whose roots are located at a certain distance from each other, and which contains vertices up to a certain distance from the root [11]. It then compares graphs based on these pairs of rooted subgraphs. To avoid isomorphism checking, graph invariants are employed to encode

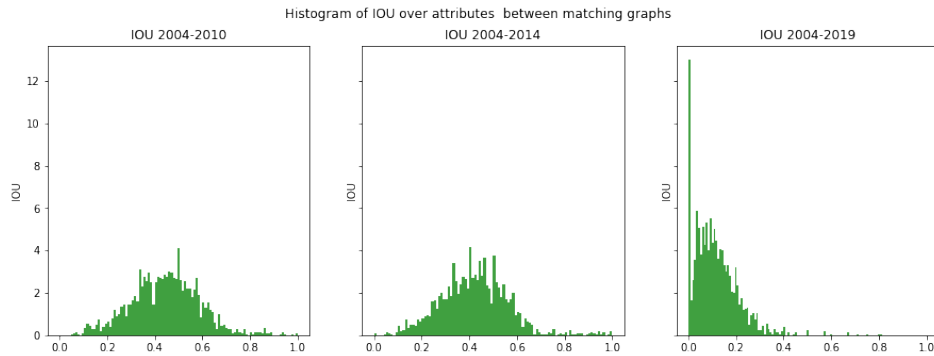


Figure 3: IOU histograms for matching graph geographic areas obtained for Moselle department. If all the nodes attributes match exactly (injective matching) across graphs, the IOU value will be 1.

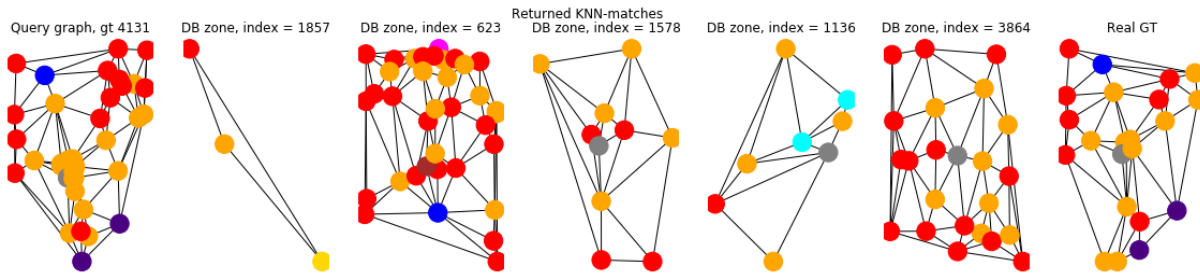


Figure 4: An example of the wrong top 5 similar graphs in 2004 returned for the query from 2019. The node colors represent the semantics of geographic entities.

each rooted subgraph [38]. The graphlet sampling kernel decomposes graphs into graphlets (i.e. small connected non-isomorphic subgraphs) and counts matching graphlets in the input graphs. The Weisfeiler-Lehman kernel [37] is based on Weisfeiler-Lehman algorithm, which core idea is to replace the discrete label of each vertex with a multiset label consisting of the original label of the vertex and the sorted set of labels of its neighbours. The resultant multiset is compressed into a new label. This relabeling procedure is then repeated for a pre-defined number of iterations. The final score is defined based on the number of nodes with the same labels across two graphs. Graph kernels do not use the ground truth corresponding labels to fit the kernel to the data, but the hyper-parameters should be tuned for the optimal performance. It leads to the selection of the time/accuracy ratio desired for the application at hand. We limited the graph kernel tests to the data with the biggest time gap, using graphs dated 2004 as a database and graphs dated 2019 as queries. The results are summarized in Table 4.

The obtained results show that the graph kernels perform rather well for the cross-time graph matching task, outperforming the Faiss which was based solely on the attributes and did not use the graph structure. The results allow us to say that the kernels which use the node attributes perform the best in our application scenario. It is interesting to see that the Weisfeiler-Lehman and Graphlet Sampling Kernels, which are both structure-based and use discrete node labels, do not show good results. The former is commonly used to check the graphs for the isomorphism, so it signifies that the structure of the graphs in our dataset differs significantly through the years, while many attributes stay the same.

Node attributes robustness in the presence of the noise. In our database, we have very precise node attributes with the precision of the six decimals thanks to access to the manually labeled and designed vector data. If the same information should have been extracted from the images automatically, the precision would have dropped due to the errors in the segmentation and vectorization stages. To simulate this realistic scenario, we made a group of tests by decreasing the number of decimals in the node attributes up to one, two, and three decimals, and by adding normally distributed noise with $\mu=0$ and $\sigma = [0.1, 0.01, 0.001]$ to the queries attributes. Faiss method results with added Gaussian noise and less precise query features are summarised in Table 5.

We performed a similar tests for the two best performing graph Kernels. The results can be found in Table 4.

Conclusion on baseline methods. Our experiments have shown that the entity’s attributes are extremely important for research retrieval. The map@5 average precision results based on geometric attributes are rather good both for the Faiss search and graph kernels. However, the presence of noise or less precise attributes leads to a significant decrease in the performance in terms of map@5 retrieval. Structure-similarity based kernels do not work on our data. The fact that local attribute-based methods are not robust to noise severely limits the application of these methods. Moreover, for some graphs, the retrieval results purely based on the node attributes are not correct, which leaves room for further improvement. The main reason for the mistakes is the objects with similar geometrical attributes and significant landscape changes in time. In practice, in the Computer Vision domain, this problem is commonly

query year	year db	kernel	parameters	map@5	t per query, sec	map@5, noise $\sigma = 0, 01$	t per query, sec
2019	2004	neighbourhood Subgraph Pairwise Dist [11]	norm, $r=3, d=2$	0.887	0.5	0.000143	24.36
2019	2004	Subgraph Matching [26]	norm, $k = 5$	0.886	3.73	0.000163	31.45
2019	2004	Weisfeiler-Lehman [37]	norm, $i=2$, Vertex-Histogram kernel	0.060	0.01	n/a	n/a
2019	2004	Graphlet sampling [38]	norm, $k=5, n = 10$	0.002	0.01	n/a	n/a

Table 4: map@5 for graph kernels. k is the upper bound for the maximum size of subgraphs/graphlets, r is the maximum considered radius between vertices, d is neighbourhood depth, n the number of randomly drawn random samples, i is the number of iterations. Note that attribute-based kernels give good results but stop working in the presence of noise.

query year	db year	map@5 precision, decimals			map@5 with noise, σ		
		1	2	3	0.1	0.01	0.001
2019	2004	0.007	0.254	0.825	0.009	0.126	0.647
2014	2004	0.007	0.261	0.835	0.010	0.170	0.778
2010	2004	0.011	0.455	0.924	0.011	0.224	0.902

Table 5: No learning retrieval baseline using Faiss similarity search and modified node attributes.

solved by doing a geometric verification between the matched key-points of two images, the golden standard pipeline being SIFT [31] + RANSAC [15]. In [14] the authors replaced the SIFT keypoints with the CNN-based descriptors. Recently, researchers proposed more advanced fully trainable neural network that matches two sets of local features by jointly finding correspondences and rejecting non-matchable points called SuperGlue [36].

However, all these methods require additional steps and processing time, which may be significant for large databases, even with adequate index structures. Therefore, in the next part, we pursue research of a learning-based model that would be able to provide a compact and effective graph matching across time. A successful model for this problem should, therefore (1) exploit the graph structures but be robust to the appearance and disappearance of new entities and noise, and (2) be able to reason about the similarity of graphs both from the graph structures as well as from learned semantics and attributes.

6 PROPOSED SIAMESE MODEL

We suggest that the relational information encoded in a graph structure can be very important when we need to distinguish between two different geographic areas with a similar set of objects and similar geometric attributes. Therefore, we propose a novel learning pipeline using a GCN [24] network to learn to match geographic areas represented as graphs across time. In the original work, GCN model was proposed to perform a node classification task for the big sparse graphs. Our model aims to learn an embedding space for variable size geographic graphs by exploring the notion of deep graph matching.

Model architecture. Given the graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we want a model that produces the learning function $f : G \rightarrow D$ through the GCN with learnable parameters w , to compare them next with the similarity score $s(D_1, D_2)$ between them in a new vector space. The encoding function f takes the A and X values of a current entity and all geographic entities within the reference area R as inputs and outputs the embedded geospatial contextual information.

Our model allows to convert graphs to descriptors, which enables efficient retrieval with fast nearest neighbour search data structures. In our experiments, we adapt Siamese networks to handle graphs to learn their embeddings. We propose the following graph matching embedding model inspired by GNN [24], which comprises 3 main parts: (1) a GCN based feature aggregation layers (with added fully-connected layers, see the detailed explanation in Annex ??), (2) a pooling layer, and (3) a final fully-connected layer. The architecture is schematically shown in Figure 5. The aggregation layers l follow the formulation of the GCN by [24] and are defined as:

$$X^{L+1} = \sigma(AX^L W) \quad (3)$$

Where A is the normalized and modified as in [24] adjacency matrix and W are weights to be trained and σ is a ReLu activation function. As in the original work by [24] we use two layers of propagation, so that the representation for each node will accumulate information in its local 2-hop neighbourhood.

After we obtained the final node representations, we aggregate across them to get graph-level representations. This could be implemented by a simple maxpooling followed by a MLP operation that reduces the node representations into a single vector and then transforms it:

$$D = MLP_G(\maxpool_{x \in n}(X_i^L)) \quad (4)$$

Where X are the learned graph nodes n features.

The proposed architecture mainly differs from [24] in the point (3) where we don't calculate the node-level features, but compute a graph level representation instead by performing a maxpooling operation over the nodes in a graph to obtain the whole graph descriptors D_G similar to [25]. The pooling layer maps the input graph of any structure and size to a fixed size-structured output.

The Siamese network consists of two identical networks (with shareable weight parameters). In our case, each of the networks is essentially a GCN with maxpooling depicted in Figure 5. During training, the embedding model will jointly reason about the graph structure as well as the graph features to come up with an embedding that reflects the notion of similarity described by the training examples. The proposed Siamese GCN model is endowed with the contrastive loss to train on the data with the ground truth correspondence. The NT-Xent [9] loss function for a positive pair of examples of the matching geographic areas through time (i, j) is defined as:

$$l_s = -\log \frac{\exp(\text{sim}(D_1, D_2)/\tau)}{\sum_{k=1}^{2N} 1_{k \neq i} \exp(\text{sim}(D_1, D_2)/\tau)} \quad (5)$$

Where: τ is the temperature, $\text{sim}(D_i, D_j)$ - cosine similarity, i, j - two graphs in batch of size N . The final loss is computed as an

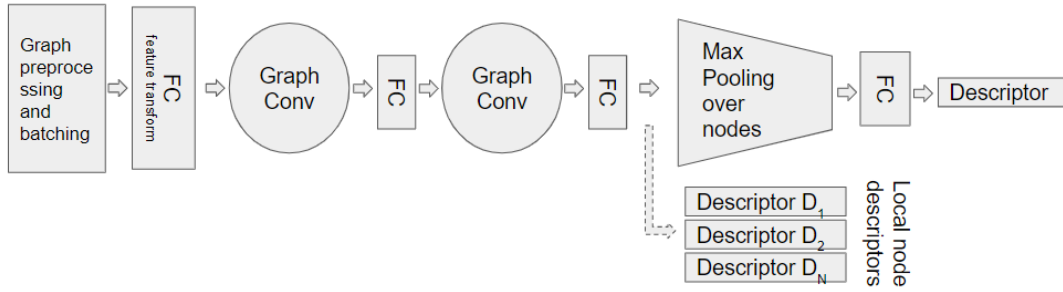


Figure 5: Schematic architecture of the model proposed to train the graph embeddings.

arithmetic mean across all positive pairs, both (i, j) and (j, i) , in a mini-batch.

$$L_f = \frac{1}{2N} [l_s(2k-1, 2k) + l_s(2k, 2k-1)] \quad (6)$$

Following the idea of [9], we created the batches of random graphs to train the model. However, instead of altering them to use as an input for a second branch of the Siamese GCN, we take the graphs representing the same geographic area but from a different time frame to form positive samples. Then the loss encourages the embeddings for the same geographic area to be closer in the embedding space in terms of the cosine distance; and the embeddings of different areas to be farther apart.

7 EXPERIMENTS

We consider two following scenarios of the graph similarity learning problem:

- Learning and testing on the data coming from the same departments across time but separating them by date
- A general across-time retrieval learning pipeline, where we separate the departments for training and testing

The first scenario aims to learn the embedding using the data with a temporal gap but using the same regions for training, validating, and testing. The second scenario aims to learn the embeddings for temporally misaligned data using different departments for training, testing, and validation.

We compute the similarity between the final descriptors using a similarity metric in the vector space from Faiss library as described above and evaluate the results using map@K metric. We also report the average retrieval time for a single query as previously.

Throughout the experiments we fixed the dimensionality of graphs embeddings to 512, trying the following commonly used values: 128, 256, 512. Our experiments have shown that when the final descriptor size is lower than 512, it's learning capacities stay the same, but generalization capacities are much lower. We can obtain the same map@5 values on the training set with all the descriptor sizes. However, the validation map@5 reaches the plateau and stop increasing on the lower map@5 values in case of the smaller descriptor size. The weights in the GCN layers are equal to 512.

In the graph pre-processing step, the discrete labels of the nodes were one-hot encoded, and the continuous attributes were left intact and concatenated with the one-hot encoded ones. Since the graphs have a different number of nodes, we use padding to create the

graphs of the same size as an input to the network. The number of nodes in a single graph is then equal to 150.

The training graph pairs are selected during the run time and are shuffled randomly at the end of each epoch. The network is trained for approximately 200 epochs until the moment the validation map@5 accuracy reaches the plateau and starts to decrease. Further training leads to overfitting on the training set, so we take the model which shows the best validation score and report its results. We used the batch size of 64 graphs through all the experiments. Batches of a bigger size led to a decrease in overall model performance.

We do not use any data augmentation techniques and NT-Xent loss allows us to avoid hard samples mining. The temperature parameter in the contrastive loss is equal to 0.5. Adam [23] optimizer is chosen for the optimization of the learning weights. The learning rate is equal to 0.1^3 , with a decay 0.1^5 and a multi-step learning rate scheduler. All the training is performed on a CPU and does not require excessive calculating power, mainly thanks to the small size of our graphs. It takes about a day to train the model end-to-end on our dataset, along with calculating map@5 precision for training and validation data after each epoch. A single branch of the model is used in the inference step.

Cross-time matching learning. We use all three departments for training, validation, and testing, separating the dataset by the year. We train on 2019-2010, validate on 2019-2014, and finally, the pairs 2019-2004 and 2010-2004 form the test set. The idea of this experiment is that we want to use known cross-time correspondences to later retrieve older data.

Generalized learning approach. In this scenario, we evaluate the generalization capabilities of our model, by using different sets of data for training and testing. We use 2 departments for learning and one for testing. The most difficult data with the biggest time gap are selected. This allows to demonstrate whether the proposed models can be used for completely new, unseen data.

7.1 Results

Table 6 summarises the results for the across-year learning scenario for our GCN-based descriptors. We report the performance of the global descriptors which corresponds to our training objective, and also the local descriptors, taking the resulting node embeddings before the maxpooling layer. We observed that the latter take much longer to compute and perform less good than the global ones, which corresponds to our learning objective. On average, the retrieval time for the query represented as a single descriptor is twice

query year	db year	role	global		local	
			map@5	t per q	map@5	t per q
2019	2010	training	0.660	0.017	0.303	0.880
2019	2014	validation	0.554	0.017	0.317	0.412
2019	2004	testing	0.552	0.017	0.371	1.140
2010	2004	testing	0.684	0.019	0.638	0.970

Table 6: The map@5 results for the global and local descriptors. Note that training and validation data do not contain clutter, but the testing data does. Time is indicated in sec.

query year	db year	role	global		local	
			map@5	t per q	map@5	t per q
2019	2010	training	0.577	0.011	0.247	0.421
2019	2014	validation	0.483	0.011	0.314	0.517
2019	2004	testing	0.487	0.014	0.305	0.521
2010	2004	testing	0.628	0.014	0.578	0.314

Table 7: The map@5 results for the global and local descriptors with noise $\sigma = 0.01$.

query year	db year	role	global		local	
			map@5	t per q	map@5	t per q
2019	2004	training	0.538	0.01	0.399	0.6
2019	2004	testing	0.576	0.003	0.371	0.2

Table 8: The map@5 results for the global and local descriptors, cross-department learning.

smaller than the one obtained with Faiss local features earlier. The obtained map@5 values are lower than the ones we got earlier with the no learning baselines.

Table 7 shows how the noise added to query graphs attributes is affecting the results in the inference stage. Here we can see that in contrast to the baseline methods, the GCN-based descriptor is relatively robust to noise, with map@5 values decreasing up to 10% in contrast with the 70% decrease of Faiss noisy local features search.

Table 8 shows the generalizability of our GCN-based descriptors. Note that the map@5 is for the similarity search in two (training) and one (testing) regions correspondingly. We kept the hyperparameters tuned for the cross-time learning scenario previously, and can see that the network is capable to create a meaningful descriptor for the new region it did not see during training, so it generalizes rather well.

Although the overall obtained map@5 results are lower than the ones from the baselines, we insist that the GCN-based network works better in the case of less precise attributes and hence has more potential for the real-case geographical graph matching task we aim at. Table 7 proves that the resulting descriptors are robust to the noise and return the correct geographic area in the 50% of the queries. Note that with the no learning baselines noise was dramatically affecting the map@5 results and GCN model is robust to noise. This property is the key property that allows the model to be used in a real-world scenario, where it is yet impossible to obtain ideal segmentation results fully automatically, and even human annotation might vary from person to person and database to database. Saying that, we assume, that although the first results obtained are not yet on the point with classical methods, it is a path to a new research direction.

8 CONCLUSION

With the availability of large volumes of geographic data, developing neural models for geographic area similarity learning is a relevant research direction and graph representation is a very powerful and intuitive way to encode geoinformation.

This paper proposes approaches for the problem of vertical image research and retrieval from a new side: we transform the semantic information into a connected graph. We create an original dataset associated with a novel deep learning-based method to learn geo graph representations across time. The proposed algorithm is currently outperformed by the classical methods, but in contrast to them, it is robust to the presence of the noise in the attributes, which makes it useful in a real-world scenario, such as research and retrieval of automatically segmented and vectorized aerial images or even a correspondence search in different databases. Besides, the proposed method can be directly used to learn the embeddings for any attributed graph similarity problem.

There are still many interesting challenges to resolve, for example, to improve the efficiency of the matching models to achieve the graph kernels accuracy, study different matching architectures, adapt the GCN capacity to graphs of different sizes, and applying these models to new application domains. The graph models with attention can intuitively work well in the aimed application and we plan to adopt an attention mechanism in the future. Another possible direction is the creation of the optimal graph representations which can lead to the best retrieval results. Finally, on the experimental part, we plan to experiment further with different splitting schemes to estimate the bias introduced by splitting train, validate, and test sets according to the year. This work can spur further research in geographical graph matching and provides the first benchmark.

ACKNOWLEDGMENTS

This work is supported by ANR, the French National Research Agency, within the ALEGORIA project, under Grant ANR-17-CE38-0014-01.

REFERENCES

- [1] [n. d.]. Le portail IGN. <https://geoservices.ign.fr/>.
- [2] Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B Aditya Prakash. 2018. Sub2vec: Feature learning for subgraphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 170–182.
- [3] Rami Al-Rfou, Bryan Perozzi, and Dustin Zelle. 2019. Ddgg: Learning graph representations for deep divergence graph kernels. In *The World Wide Web Conference*, 37–48.
- [4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.
- [5] Karsten M Borgwardt and Hans-Peter Kriegel. 2005. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*. IEEE, 8–pp.
- [6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "siamese" time delay neural network. In *Advances in neural information processing systems*, 737–744.
- [7] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).
- [8] Ushasi Chaudhuri, Biplab Banerjee, and Avik Bhattacharya. 2019. Siamese graph convolutional network for content based remote sensing image retrieval. *Computer Vision and Image Understanding* 184 (2019), 22–30.
- [9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* (2020).

- [10] Dahjung Chung, Khalid Tahboub, and Edward J Delp. 2017. A two stream siamese convolutional neural network for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*. 1983–1991.
- [11] Fabrizio Costa and Kurt De Grave. 2010. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 26th International Conference on Machine Learning*. Omnipress; Madison, WI, USA, 255–262.
- [12] I Couloigner, KPB Thomson, Y Bedard, B Moulin, E LeBlanc, C Djima, C Latouche, and N Spicher. 2002. Towards automating the selection of ground control points in Radersat images using a topographic database and vector-based data matching. *Photogrammetric engineering and remote sensing* 68, 5 (2002), 433–440.
- [13] Boris Delaunay et al. 1934. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7, 793-800 (1934), 1–2.
- [14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 224–236.
- [15] Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.
- [16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.
- [17] Norbert Haala and George Vosselman. 1993. Recognition of road and river patterns by relational matching. *International Archives of Photogrammetry and Remote Sensing* 29 (1993), 969–969.
- [18] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. 2018. Cvmnet: Cross-view matching network for image-based ground-to-aerial geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7258–7267.
- [19] Zhenlu Jin, Xuezhi Wang, Bill Moran, Quan Pan, and Chunhui Zhao. 2016. Multi-region scene matching based localisation for autonomous vision navigation of UAVs. *The Journal of Navigation* 69, 6 (2016), 1215–1233.
- [20] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).
- [21] Pascal Kaiser, Jan Dirk Wegner, Aurélien Lucchi, Martin Jaggi, Thomas Hofmann, and Konrad Schindler. 2017. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing* 55, 11 (2017), 6054–6068.
- [22] Margarita Khokhlova, Valérie Gouet-Brunet, Nathalie Abadie, and Liming Chen. 2020. Cross-year multi-modal image retrieval using siamese networks. In *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1–5.
- [23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [24] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [25] Boris Knyazev. 2018. Graph Classification with Graph Convolutional Networks in PyTorch.
- [26] Nils Kriege and Petra Mutzel. 2012. Subgraph matching kernels for attributed graphs. *arXiv preprint arXiv:1206.6483* (2012).
- [27] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. 2020. A survey on graph kernels. *Applied Network Science* 5, 1 (2020), 1–42.
- [28] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. 2018. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8971–8980.
- [29] Jia Li, Yu Rong, Hong Cheng, Helen Meng, Wenbing Huang, and Junzhou Huang. 2019. Semi-supervised graph classification: A hierarchical graph perspective. In *The World Wide Web Conference*. 972–982.
- [30] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. 2019. Graph matching networks for learning the similarity of graph structured objects. *arXiv preprint arXiv:1904.12787* (2019).
- [31] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
- [32] Carlos A Martinez-Ortiz. 2010. 2D and 3D shape descriptors. (2010).
- [33] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. 2017. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005* (2017).
- [34] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. 2016. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning* 102, 2 (2016), 209–245.
- [35] Blake Regalia, Krzysztof Janowicz, and Grant McKenzie. 2019. Computing and querying strict, approximate, and metrically refined topological relations in linked geographic data. *Transactions in GIS* 23, 3 (2019), 601–619.
- [36] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4938–4947.
- [37] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* 12, 9 (2011).
- [38] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*. 488–495.
- [39] Giannis Sgolidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. 2020. GraKeL: A Graph Kernel Library in Python. *Journal of Machine Learning Research* 21, 54 (2020), 1–5.
- [40] Onur Tasar, Yuliya Tarabalka, and Pierre Alliez. 2019. Incremental learning for semantic segmentation of large-scale remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12, 9 (2019), 3524–3537.
- [41] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 297–304.
- [42] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [43] Kirill Veselkov, Guadalupe Gonzalez, Shahad Aljifri, Dieter Galea, Reza Mirnezami, Jozef Youssef, Michael Bronstein, and Ivan Laponogov. 2019. HyperFoods: Machine intelligent mapping of cancer-beating molecules in foods. *Scientific reports* 9, 1 (2019), 1–12.
- [44] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. 2010. Graph kernels. *The Journal of Machine Learning Research* 11 (2010), 1201–1242.
- [45] Peter Wills and François G Meyer. 2020. Metrics for graph comparison: A practitioner’s guide. *Plos one* 15, 2 (2020), e0228728.
- [46] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.
- [47] Bo Yan. 2019. *Geographic knowledge graph summarization*. Vol. 39. IOS Press.
- [48] Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Song Gao. 2017. From itdl to place2vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *Proceedings of the 25th ACM SIGSPATIAL international conference on advances in geographic information systems*. 1–10.
- [49] Pinar Yanardag and SVN Vishwanathan. 2015. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1365–1374.