machine learning & knowledge extraction

MDPI

Article

Attributed Relational SIFT-Based Regions Graph: Concepts and Applications

Mario Manzo †

Information Technology Services, University of Naples "L'Orientale", 80121 Naples, Italy; mmanzo@unior.it; Tel.: +39-081-6909229

† Current address: Via Nuova Marina, 59, 80133 Naples, Italy.

Received: 12 June 2020; Accepted: 3 August 2020; Published: 6 August 2020

Abstract: In the real world, structured data are increasingly represented by graphs. In general, the applications concern the most varied fields, and the data need to be represented in terms of local and spatial connections. In this scenario, the goal is to provide a structure for the representation of a digital image, called the Attributed Relational SIFT-based Regions Graph (ARSRG), previously introduced. ARSRG has not been described in detail, and for this purpose, it is important to explore unknown aspects. In this regard, the goal is twofold: first, to provide a basic theory, which presents formal definitions, not yet specified above, clarifying its structural configuration; second, experimental, which provides key elements about adaptability and flexibility to different applications. The combination of the theoretical and experimental vision highlights how the ARSRG is adaptable to the representation of the images including various contents.

Keywords: graph-based image representation and analysis; image classification; kernel method; graph matching; graph embedding; bag of graph words

1. Introduction

Among issues related to human vision, the processing of visually complex entities is one of the most important. The processing of information is often based on local-to-global or global-to-local connections [1]. The local-to-global concept concerns the transitions from local details of scene to a global configuration, while global-to-local works in the reverse order, from global configuration towards the details. For example, an algorithm for face recognition, which uses the local-to-global approach, starts with eyes, nose, and ears recognition and, finally, brings face configuration. Differently, a global-to-local algorithm first identifies the face, which leads to the identification of details (eyes, nose, and ears). During the task of human recognition, the global configuration of a scene plays a key role, especially when subjects see the images for a short duration of time. Furthermore, humans leverage local information as an effective way to recognize scene categories. Higher level visual perception theories distinguish individual elements at the local and global level, in which the information on many local components is perceptually grouped [2]. Nodes and edges in a graph representation encode information with the purpose to highlight relations among raw data. Many fields such as computer vision and pattern recognition adopt data graph representations and related manipulation algorithms. Specifically, in the image processing field, graphs are used to represent digital images in many ways. The standard approach concerns partitioning of the image into dominant disjoint regions, where local and spatial features are respectively nodes and edges. Local features describe the intrinsic properties of regions (such as shape, colors, texture), while spatial features provide topological information about the neighborhood. Image representation is one of the crucial steps for systems working in the image retrieval field. Modern Content-Based Image Retrieval (CBIR) systems consider essentially the image's basic elements (colors, textures, shapes, and topological relationships) extracted from the entire image,

in order to provide an effective representation. Through the analysis of these elements, compositional structures are produced. Other systems, called Region-Based Image Retrieval [3] (RBIR), focus their attention on specific image regions instead of the entire content to extract features. In this paper, a graph structure for image representation, called Attributed Relational SIFT-based Regions Graph (ARSRG), is described, analyzed, and discussed with reference to previous works [4–7]. There are two main parts: examination of the structure, through the definition of its components, and the collection and analysis of the previously obtained results. The main goal of ARSRG is to create a connection between local and global features in the image through a hierarchical description. Global features are extracted using a segmentation technique, while local features are based on a Local Invariant Feature Extraction (LIFE) method. The structure provides different information arising from image regions, topological relations among them, and local invariant features. In order to extract stable descriptors (robust to deformations), SIFT features have been selected among all LIFE methods, as they extract salient points of an image in the scale-space. Moreover, new definitions and properties arising from the detailed analysis of the structure are introduced. This theoretical analysis, based on the introduction of different definitions, has been helpful to go into the main and secondary components of ARSRG, with the purpose to better understand the phases of construction and comparison. Finally, through a wide experimental phase, how the structure is adaptable to different types of application contexts is shown. The latter involved a collection and a depth analysis of results previously obtained in different fields both in terms of image content and application. It was a crucial phase because the goal was to identify the common aspects that mainly supported the theoretical basis. The paper is organized as follows: Section 2 includes a related research about graph-based image representation including Scale-Invariant Feature Transform (SIFT) [8]. Sections 3–5 are dedicated to ARSRG's description, definitions, and properties. Experimental results and conclusions are respectively reported in Sections 6 and 7.

2. Related Work

The literature reports many approaches that combine local and spatial information arising from SIFT features. Commonly, a graph structure encodes information about keypoints located in a certain position of an image. Nodes represent SIFT descriptors, while edges describe spatial relationships between different keypoints.

In [9], a graph G_1 represents a set of SIFT keypoints from the image I_1 and is defined as:

$$G_1 = (V_1, M_1, Y_1) \tag{1}$$

where $v_\alpha \in V_1$ is a node related to a SIFT keypoint with position $(p_1^{(\alpha)}, p_2^{(\alpha)})$, $y_\alpha \in Y_1$ is the SIFT descriptor attached to node v_α, and M_1 is the adjacency matrix. If $M_{1\,\alpha\beta} = 1$, the nodes v_α and v_β are adjacent, $M_{1\,\alpha\beta} = 0$ otherwise.

In [10], the authors combined the local information of SIFT features with global geometrical information in order to estimate a robust set of feature matches. This information is encoded using a graph structure:

$$G_0 = (V_0, B, Y) \tag{2}$$

where $v \in V_0$ is a node associated with a SIFT keypoint, B is the adjacency matrix, $B_{v,v'} = 1$ if the nodes v and v' are connected, $B_{v,v'} = 0$ otherwise, while $y_v \in Y$ is the SIFT descriptor associated with node v.

In [11], nodes were associated with N image regions related to an image grid, while edges connect each node with its four neighbors. Basic elements are not pixels, but regions extended in the x (horizontal) and y (vertical) directions. The nodes are identified using their coordinates on the grid. The spatial information associated with nodes is indices $d_n = (x_n, y_n)$. Furthermore, a feature vector F_n is associated with the corresponding image region and, then, with a node. The image is divided into overlapping regions of 32 × 32 pixels. Four 128-dimensional SIFT descriptors, for each region, are extracted and concatenated.

In [12], the graph-based image representation included SIFT features, MSER [13], and Harris-affine [14]. Given two graphs $G^P = (V^P, E^P, A^P)$ and $G^P = (V^Q, E^Q, A^Q)$, representing images I^P and I^Q, V is the set of nodes, image features extracted, E the set of edges, features' spatial relations, and A the set of attributes, information associated with features extracted.

In [15], SIFT features were combined in the form of a hyper-graph. A hyper-graph $G = (V, E, A)$ is composed of nodes $v \in V$, hyper-edges $e \in E$, and attributes $a \in A$ related to hyper-edges. A hyper-edge e encloses a subset of nodes with size $\delta(e)$ from V, where $\delta(e)$ represents the order of a hyper-edge.

In [16], an approach to 3D object recognition was presented. The graph matching framework is used in order to enable the utilization of SIFT features and to improve robustness. Different from standard methods, test images are not converted into finite graphs through operations of discretization or quantization. Then, the continuous graph space is explored in the test image at detection time. To this end, local kernels are applied to indexing image features and to enable a fast detection.

In [17], an approach to the matching features problem with the application of scene recognition and topological SLAM was proposed. For this purpose, the scene images are encoded using a particular data structure. Image representation is built through two steps: image segmentation using the JSEG [18] algorithm and invariant feature extraction with MSER and SIFT descriptors in a combined way.

In [19], SIFT features based on visual saliency and selected to construct object models were extracted. A Class Specific Hypergraph ($CSHG$) to model objects in a compact way was introduced. The hypergraphs are built on different Delaunay graphs. Each one is created from a set of selected SIFT features using a single prototype image of an object. Using this approach, the object models can be represented through a minimum of object views.

The authors in [20] provided a solution to the object recognition problem representing images through SIFT-based graph structural expression. A graph structure is created using lines to connect SIFT keypoints. $G = (V, E, X)$ represents the graph; the set E represents edges; the set V represents vertices; and the set X the associated SIFT descriptors. The node represents a keypoint detected by the SIFT algorithm, and the associated label is the 128-dimension SIFT descriptor. The edge $e_{\alpha\beta} \in E$ connects two nodes $u_\alpha \in V$ and $u_\beta \in V$. The graph can be defined as complete if all keypoints, extracted from the image, are connected among them. Formally, the set of edges is defined as follows:

$$E = \left\{ e_{ij} \mid \forall i, j \frac{\parallel p_i - p_j \parallel}{\sqrt{\sigma_i \sigma_j}} < \lambda \right\} \tag{3}$$

where $p = (p_x, p_y)$ represents the keypoint spatial coordinates, σ its scale, and λ is a threshold value. An edge does not exist when the value is greater than the threshold λ. In this way, an extra edge is not created. This formulation of the proximity graph reduces the computational complexity and, at same time, improves the detection performance.

In [21], the median *K-nearest-neighbor*(K-NN) graph $G_P = (V_P, E_P)$ was defined. A vertex v_i for each of the N points p_i is created, with $V_P = v_1, ..., v_N$. Furthermore, a non-directed edge (i, j) is created when p_j is one of the K closest neighbors of p_i and $\parallel p_i - p_j \parallel \leq \eta$. η is the median of all distances between pairs of vertices and can be defined as:

$$\eta = median_{(l,m) \in V_P \times V_P} ||p_l - p_m|| \tag{4}$$

During K-NN graph construction, a vertex p_i can be considered completely disconnected if there are no K vertices that support the structure. The graph G_P has the $N \times N$ adjacency matrix A_P, where $A_P(i, j) = 1$ when $(i, j) \in E_P$ and $A_P(i, j) = 0$ otherwise.

3. Attributed Relational SIFT-Based Regions Graph

In this section, the Attributed Relational SIFT-based Regions Graph (ARSRG) is defined based on two main steps: feature extraction and graph construction. Feature extraction consists of Region of Interest (ROI) extraction from the image through a segmentation algorithm. Connected components in the image are then detected with the aim of building the *Region Adjacency Graph (RAG)* [22], to describe the spatial relations between image regions. Simultaneously, SIFT [8] descriptors, which ensure invariance to rotation, scaling, translation, illumination changes, and projective transforms, are extracted from the original image. Graph construction consists of the building of the graph structure. Three levels can be distinguished in ARSRG: *root node*, *RAG nodes*, and *leaf nodes*. At the first level, the *root node* represents the image and is connected to all *RAG nodes* at the second level. Adjacency relationships among different image regions are encoded through *RAG nodes*. Thus, adjacent regions in the image are represented by connected nodes. In addition, each *RAG* node is connected to the *root node* at the higher level. Finally, SIFT descriptors extracted from the image are represented through *leaf nodes*. At the third level, two types of configurations can appear: *region based* and *region graph based*. In the *region-based* configuration, a keypoint is associated with a region based on its spatial coordinates, whereas the *region graph-based* configuration describes keypoints belonging to the same region connected by edges (which encode spatial adjacency). Below, the steps of feature extraction and graph construction are described in detail.

3.1. Feature Extraction

3.1.1. Region of Interest Extraction

ROIs from the image through a segmentation algorithm called JSEG [18] are extracted. JSEG performs segmentation through two different steps: color quantization and spatial segmentation. The first step consists of a coarse quantization without degrading the image quality significantly. The second step provides a spatial segmentation on the class-map without considering the color similarity of the related pixel.

3.1.2. Labeling Connected Components

The next step involves the labeling of connected components on the segmentation result. A connected component is an image region consisting of contiguous pixels of the same color. The process of connected components labeling an image B produces an output image LB that contains labels (positive integers or characters). A label is a symbol naming an entity exclusively. Regions connected by the four-neighborhood and eight-neighborhood will have the same label (represented in Algorithms 1 and 2 by the variable m containing a numerical value). Algorithm 1 shows a version of connected components' labeling.

Algorithm 1 *Connected components' labeling.*

Require: I - Image to Label;
Ensure: I - Image Labeled;

```
m=0
for y=1:I_size_y do
    for x=1:I_size_x do
        if I[i][j] == 0 then
            m=m+1
            Component Label(I, x, y, m)
        end if
    end for
end for
return I
```

Algorithm 2 *Component label*.

Require: I - Image to Label; i, j - image index; l - label;
Ensure: $\oslash$;

1: **if** I[i][j] == 0 **then**
2: I[i][j]=m
3: $Component\ Label(I, i-1, j-1, m)$
4: $Component\ Label(I, i-1, j, m)$
5: $Component\ Label(I, i-1, j+1, m)$
6: $Component\ Label(I, i, j-1, m)$
7: $Component\ Label(I, i, j+1, m)$
8: $Component\ Label(I, i+1, j-1, m)$
9: $Component\ Label(I, i+1, j, m)$
10: $Component\ Label(I, i+1, j+1, m)$
11: **end if**

3.1.3. Region Adjacency Graph Structure

The second level of the ARSRG hosts a graph-based image representation named the *Region Adjacency Graph (RAG)* [22]. In Algorithm 3, a pseudocode version of the *RAG* algorithm is shown. A region represents an elementary component of the image, based on the image segmentation result. In the *RAG*, a node is a region, and an edge describes the adjacency between two nodes. *RAG* is built with reference to spatial relations between regions. Two regions are defined to be spatially close if they share the same boundary. In this regard, a neighborhood check of the labeled region is performed between the label of the pixel under consideration, named $pixel(i, j)$ in Lines 2–4, and the labels of the pixels belonging to its neighborhood (the eight directions included in the eight-neighborhood). If the latter contain a different label with respect to $pixel(i, j)$, then this means that there are two adjacent regions represented in the RAG. The *RAG* is defined as a graph $G = (V, E)$, where nodes are regions in V and edges E are the boundaries that connect them. G is encoded through the adjacency matrix, $Adjacency\_matrix$ (Line 5), which describes the topology of the graph connections. For example, if $Adjacency\_matrix(i, j)$ contains one, this means that the regions i, j will be connected in the image. Moreover, one of the main properties of *RAG* is the invariance to translation and rotation, useful for a high-level image representation.

Algorithm 3 *Region adjacency graph*.

Require: *Labeled_image*;
Ensure: *Graph Structure* $(Adjacency\_matrix)$;

1: $Adjacency\_matrix = 0$
2: **for** $pixel(i, j) \in Labeled\_image$ **do**
3: **for** $pixel(x, y) \in 8 - neighborhood$ **do**
4: **if** $pixel(i, j) \neq pixel(x, y)$ **then**
5: $Adjacency\_matrix(pixel(i, j), pixel(x, y)) = 1$
6: **end if**
7: **end for**
8: **end for**
9: **return** $Adjacency\_matrix$

3.1.4. Scale-Invariant Feature Transform

SIFT [8] descriptors are extracted to ensure invariance to rotation, scaling, translation, partial illumination changes, and projective transform in the image description. SIFT is computed during the feature extraction phase, through a parallel task with respect to *RAG* creation.

3.2. Graph Construction

The ARSRG building process consists of the creation of three levels:

1. **Root node**: the node located at the first level of the graph structure and representing the image. It is connected to all nodes at the next level.
2. **Region Adjacency Graph (RAG) nodes**: adjacency relations among different image regions based on the segmentation result. Thus, adjacent image regions are represented by nodes connected at this level.
3. **Leaf nodes**: The set of SIFT features extracted from the image. Two types of connections are provided:
   (a) *Region based*: A leaf node represents a SIFT keypoint obtained during feature extraction. Each leaf node-keypoint is associated with a region based on its spatial coordinates in the image. At this level, each node is connected to just one *RAG* higher level node (Figure 1a).
   (b) *Region graph based*: In addition to the previous configuration, leaf nodes-keypoints belonging to the same region are connected by edges, which encode spatial adjacency, based on thresholding criteria (Figure 1b).

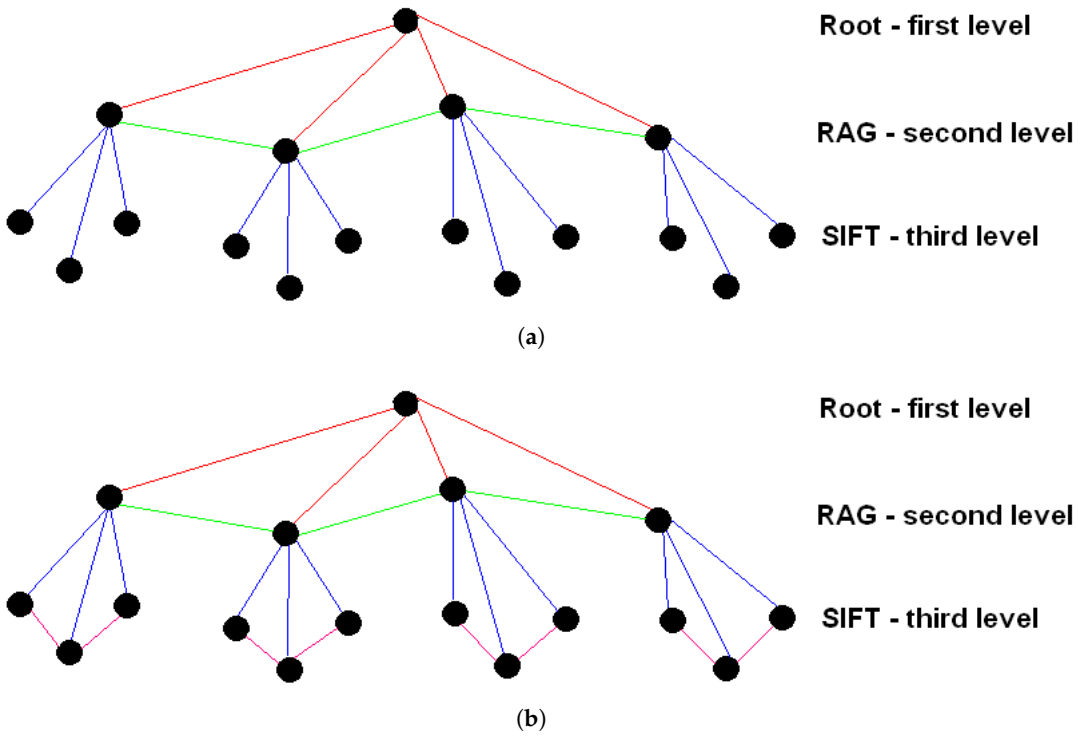


Figure 1. Region-based (**a**) and region graph-based (**b**) configurations. RAG, Region Adjacency Graph.

4. Formal Definitions

This section introduces detailed definitions for the purpose of formally fixing the ARSRG structure. Definitions 1 and 2 describe the components related to the two configurations (as shown in Figure 1). Definitions 3 and 4 define the sets of attributes associated with the nodes, through the functions introduced by Definitions 5 and 6, of the second and third level. Definitions 7 and 8 introduce connection structures for SIFTs. From Definitions 9 to 11, different types of edges between the levels are described. Finally, Definitions 12 and 13 include the support structures for the second and third level. The ARSRG structure is defined based on two leaf node configurations.

Definition 1. ***ARSRG***$_{1st}$ *(first leaf nodes' configuration):* G *is defined as a tuple* $G = (V_{regions}, E_{regions}, VF_{SIFT}, E_{regions-SIFT})$*, where:*

- $V_{regions}$*, the set of region nodes.*
- $E_{regions} \subseteq V_{regions} \times V_{regions}$*, the set of undirected edges, where* $e \in E_{regions}$ *and* $e = (v_i, v_j)$ *is an edge that connects nodes* $v_i, v_j \in V_{regions}$*.*
- VF_{SIFT}*, the set of SIFT nodes.*
- $E_{regions-SIFT} \subseteq V_{regions} \times VF_{SIFT}$*, the set of directed edges, where* $e \in E_{regions-SIFT}$ *and* $e = (v_i, vf_j)$ *is an edge that connects source node* $v_i \in V_{regions}$ *and destination node* $vf_j \in VF_{SIFT}$*.*

Definition 2. ***ARSRG***$_{2nd}$ *(second leaf nodes' configuration):* G *is defined as a tuple* $G = (V_{regions}, E_{regions}, VF_{SIFT}, E_{regions-SIFT}, E_{SIFT})$*, where:*

- $V_{regions}$*, the set of region nodes.*
- $E_{regions} \subseteq V_{regions} \times V_{regions}$*, the set of undirected edges, where* $e \in E_{regions}$ *and* $e = (v_i, v_j)$ *is an edge that connect nodes* $v_i, v_j \in V_{regions}$
- VF_{SIFT}*, the set of SIFT nodes.*
- $E_{regions-SIFT} \subseteq V_{regions} \times VF_{SIFT}$*, the set of directed edges, where* $e \in E_{regions-SIFT}$ *and* $e = (v_i, vf_j)$ *is an edge that connects source node* $v_i \in V_{regions}$ *and destination node* $vf_j \in VF_{SIFT}$*.*
- $E_{SIFT} \subseteq VF_{SIFT} \times VF_{SIFT}$*, the set of undirected edges, where* $e \in E_{SIFT}$ *and* $e = (vf_i, vf_j)$ *is an edge that connect nodes* $vf_i, vf_j \in V_{SIFT}$

ARSRG structures, first and second leaf node configuration, are created based on Definitions 1 and 2. The nodes belonging to sets $V_{regions}$ and VF_{SIFT} are associated with features extracted from the image. In particular:

Definition 3. $F_{regions}$ *is a set of vector attributes associated with nodes in* $V_{regions}$*. An element,* $f_i \in v_i$*, is associated with a node of the ARSRG structure at the second level. It contains the region dimension (pixels).*

Definition 4. F_{SIFT} *is a set of vector attributes associated with nodes in* VF_{SIFT}*. An element,* $f_i \in vf_i$*, is associated with a node of the ARSRG structure at the third level. It contains a SIFT descriptor.*

The association between features and nodes is performed through assignment functions defined as follows:

Definition 5. *The node-labeling function* $L_{regions}$ *assigns a label to each node* $v \in V_{regions}$ *of the ARSRG at the second level. The node label is a feature attribute* d_i *extracted from the image. The label value is the dimension of the region (pixels number). The labeling procedure of a* v *node occurs during the process of the ARSRG construction.*

Definition 6. *The SIFT node-labeling function* L_{SIFT} *assigns a label to each node* $vf \in VF_{SIFT}$ *of the ARSRG at the third level. The node label is a feature vector* f_i*, the keypoint, extracted from the image. The labeling procedure of a* vf *node checks the position of the keypoint in the image compared to the region to which it belongs.*

Furthermore, the *RAG nodes* $\in V_{regions}$ are doubly linked in horizontal order, among them, and in vertical order, with nodes $\in VF_{SIFT}$. Edges $\in E_{regions}$ are all undirected from left to right, while edges $\in E_{regions-SIFT}$ are all directed from top to bottom. The *root node* maintains a list of edges outgoing to *RAG nodes*. Furthermore, each *RAG node* maintains three linked lists of edges: one for outgoing from *RAG nodes*, one for outgoing from *leaf nodes*, and one for ingoing to the *root node*. Finally, each *leaf node* maintains two linked lists of edges: one for ingoing to *RAG nodes* and one for outgoing from *leaf nodes*. The edges in each list are ordered based on the distances between end nodes: shorter

edges come first. These lists of edges have direct geometrical meanings: each node is connected to another node in one direction: left, right, top, and bottom.

A very important aspect concerns the organization of the third level of the ARSRG structure. To this end, the SIFT Nearest-Neighbor Graph ($SNNG$) is introduced.

Definition 7. *An* $SNNG = (VF_{SIFT}, E_{SIFT})$ *is defined as:*

- VF_{SIFT}*: the set of nodes associated with SIFT keypoints;*
- E_{SIFT}*: the set of edges, where for each* $v_i \in VF_{SIFT}$*, an edge* (v_i, v_{ip}) *if and only if* $dist(v_i, v_{ip}) < \tau$ *exists.* $dist(v_i, v_{ip})$ *is the Euclidean distance applied to the x and y position of the keypoints in the image;* τ *is a threshold value, and p stems from one to k, k being the size of* VF_{SIFT}*.*

This notation is very useful during the matching phase. Indeed, each $SNNG$ indicates the set of SIFT features belonging to the image region, with reference to Definition 2, and represents SIFT features organized from the local and spatial point of view. A different version of $SNNG$ is called the complete SIFT Nearest-Neighbor Graph ($SNNGc$).

Definition 8. *An* $SNNGc = (VF_{SIFT}, E_{SIFT})$ *is defined as:*

- VF_{SIFT}*: the set of nodes associated with SIFT keypoints;*
- E_{SIFT}*: the set of edges, where for each* $v_i \in VF_{SIFT}$*, an edge* (v_i, v_{ip}) *if and only if* $dist(v_i, v_{ip}) < \tau$ *exists.* $dist(v_i, v_{ip})$ *is the Euclidean distance applied to the x and y position of keypoints in the image;* τ *is a threshold value; and p stems from one to k, k being the size of* VF_{SIFT}*. In this case,* τ *is greater than the maximum distance between keypoints.*

Another important aspect concerns the difference between vertical and horizontal relationships among nodes in the ARSRG structure. Below, these relations, edges, are defined.

Definition 9. *A region horizontal edge e,* $e \in E_{regions}$*, is an undirected edge* $e = (v_i, v_j)$ *that connects nodes* $v_i, v_j \in V_{regions}$*.*

Definition 10. *A SIFT horizontal edge e,* $e \in E_{SIFT}$*, is an undirected edge* $e = (vf_i, vf_j)$ *that connects nodes* $vf_i, vf_j \in V_{SIFT}$*.*

Definition 11. *A vertical edge e,* $e \in E_{regions-SIFT}$*, is a directed edge* $e = (v_i, vf_j)$ *that connects nodes* $v_i \in V_{regions}$ *and* $vf_j \in VF_{SIFT}$ *from source node* v_i *to destination node* vf_j*.*

As can be seen, horizontal and vertical edges connect nodes of the same and different levels, respectively. Finally, these relations are represented through adjacency matrices defined below.

Definition 12. *The binary regions' adjacency matrix* $S_{regions}$ *describes the spatial relations among RAG nodes. An element* s_{ij} *defines an edge,* $e = (v_i, v_j)$*, connecting nodes* $v_i, v_j \in V_{regions}$*. Hence, an element* $s_{ij} \in S_{regions}$ *is set to one if node* v_i *is connected to node* v_j*, zero otherwise.*

Definition 13. *The binary SIFT adjacency matrix* S_{SIFT} *describes the spatial relations among leaf nodes. An element* s_{ij} *defines an edge,* $e = (vf_i, vf_j)$*, connecting nodes* $vf_i, vf_j \in VF_{SIFT}$*. Hence, an element* $s_{ij} \in S_{SIFT}$ *is set to one if node* vf_i *is connected to node* vf_j*, zero otherwise.*

Figure 2 shows the two different ARSRG structures on a sample image.

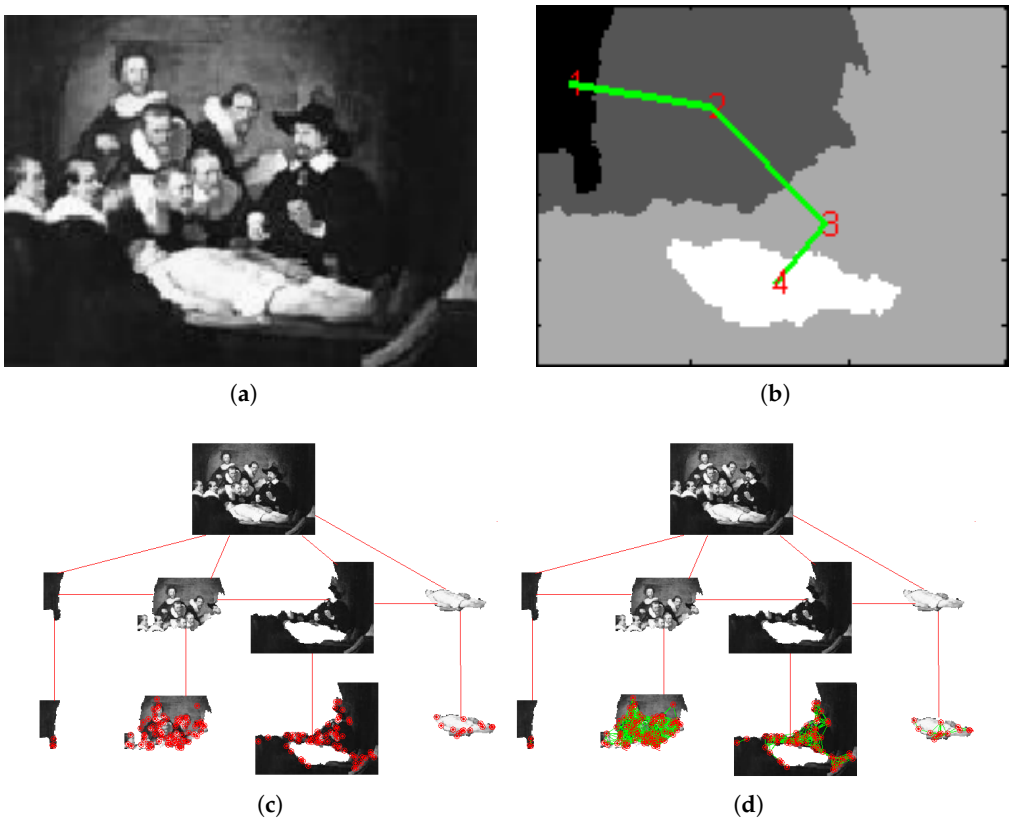


Figure 2. (**a**) Original image; (**b**) *RAG* composed of four regions; (**c**) region-based leaf node configuration; (**d**) region graph-based leaf node configuration. The red points in (**c**,**d**) represent SIFT keypoints belonging to regions, while the green lines in (**d**) represent the edges of the graph-based leaf node configuration.

5. Properties

In this section, the ARSRG structure's properties arising from the feature extraction and graph construction steps are highlighted.

Region features and structural information: The main goal of the ARSRG structure is to connect regional features and structural information. The first step concerns image segmentation in order to extract ROIs. This is a step towards the extraction of semantic information from a scene. Once the image has been segmented, the *RAG* structure is created. This feature representation highlights individual regions and spatial relations existing among them.

Horizontal and vertical relations: The ARSRG structure presents two types of relations (edges) between image features: horizontal and vertical. Vertical edges define the image topological structure, while horizontal edges define the spatial constraints for node (region) features. Horizontal relations (Definitions 9 and 10) concern ROIs and SIFT features located at the second level of the structure. The general goal is to provide the information of spatial closeness, define spatial constraints on the node attributes, and characterize the feature map of a specific resolution level (detail) on a defined image and that can be differentiated according to the computational complexity and the occurrence frequency. Their order is in the range $\{1, \ldots, n\}$, where n is the number of features specified through the relations. In a different way, vertical relations (Definition 11) concern connections between individual regions and their features. The vertical directed edges connect nodes among the second and third levels of ARSRG (*RAG* nodes to *leaf* nodes) and provide a parent-child relationship. In this context, the role of the ARSRG structure is to create a bridge between the defined relations. This aspect leads to some advantages, i.e., the possibility to explore the structure both in breadth and in depth during the matching process.

Region features invariant to point of view, illumination, and scale: Building local invariant region descriptors is a hot topic of research with a set of applications such as object recognition, matching, and reconstruction. Over the last few years, great success has been achieved in designing descriptors invariant to certain types of geometric and photometric transformations. Local Invariant Feature Extraction (LIFE) methods work in order to extract stable descriptors starting from a particular set of characteristic regions of the image. LIFE methods were chosen, for region representation, in order to provide invariance to certain conditions. These local representations, created by using information extracted from each region, are robust to certain image deformations such as illumination and viewpoint changing. The ARSRG structure includes SIFT features, identified in [23] as the most stable representations between different LIFE methods.

Advantages due to detailed information located on different levels: The detailed image description, provided by the ARSRG structure, represents an advantage during the comparison phase. In a hierarchical way, the matching procedure explores global, local, and structural information, within the ARSRG. The first step involves a filtering procedure for regions based on size. Small regions, containing poor information, are removed. Subsequently, the matching procedure goes to the next level of the ARSRG structure, analyzing features of single regions to obtain a stronger match. The goal is to solve the mapping on multiple SNNGs (Definition 7) of the ARSRGs. In essence, this criterion identifies partial matches among SNNGs belonging to ARSRGs. During the procedure, different combinations of graph SNNGs are identified, and a hierarchy of the matching process is constructed. In this way, the overall complexity is reduced, which is expected to show a considerable advantage especially for large ARSRGs.

Advantages due to matching region-by-region: Region-Based Image Retrieval (RBIR) [24] systems work with the goal of extracting and defining the similarity between two images based on regional features. It has been demonstrated that users focus their attention on specific regions rather than the entire image. Region-based image representation has proven to be more close to human perception. In this context, in order to compare the ARSRG structures, a region matching scheme based on the appearance similarities of image segmentation results can be adopted. The region matching algorithm exploits the regions provided by segmentation and compares the features associated with them. The pairwise region similarities are computed from a set of SIFT features belonging to regions. The matching procedure is asymmetric. The input image is segmented into regions, and its groups of SIFT keypoints can be matched within a consistent portion of the other image. In this way, the segmentation result is used to create regions of candidate keypoints, avoiding incompatible regions for two images of the same scene.

False matches' removal: One of the main issues of LIFE methods concerns the removal of false matches. It has been shown that LIFE methods produce a number of false matches, during the comparison phase, that significantly affect accuracy. The main reason concerns the lack of correspondence among image features (for example due to partial background occlusion of the scene). Standard similarity measures, based on the features' descriptor, are widely used, even if they rely only on region appearance. In some cases, it cannot be sufficiently discriminating to ensure correct matches. This problem is more relevant in the presence of low or homogeneous textures and leads to many false matches. The application of the ARSRG structure provides a solution for this problem. In order to reduce false matches, small ARSRG region nodes and the associated SIFT descriptors are removed. Indeed, small regions and their associated features are not very informative, neither in image description nor matching. The ratio test [8] or graph matching [25] can be applied to perform a comparison among remaining regions. This filtering procedure has a strong impact on experiments, resulting in a relevant accuracy improvement.

6. Experimental Results

This section provides experimental results arising from different application fields. In particular:

1. Graph matching [4]: The ARSRG is adopted to address the art painting retrieval problem. The ARSRG similarities, exploiting local information and topological relations, are measured through a graph matching algorithm.
2. Graph embedding [5]: The ARSRG is adopted to effectively tackle the object recognition problem. A framework to embed graph structures into the vector space is built;
3. Bag of graph words [6]: A vector encoding a histogram, the frequency of ARSRGs, for image representation within the classification task is adopted.
4. Kernel graph embedding [7]: The ARSRG is adopted to effectively tackle the imbalanced classification problem. The Kernel Graph Embedding on Attributed Relational Scale-Invariant Feature Transform-based Regions Graph (KGEARSRG) provides a vector-based image representation.

6.1. Graph Matching

This section reviews the results previously obtained in [4]. Three datasets are adopted to compare the ARSRG with the LIFE methods, graph matching algorithms, and a CBIR system. The first dataset, described in [26], is obtained by the union of images of Olga's gallery (http://www.abcgallery.com/index.html) and Travel Webshots (http://travel.webshots.com). The second dataset, described in [27], is obtained by painting photos taken from the Cantor Arts Center (http://museum.stanford.edu/). The third dataset, described in [28], contains 1002 images. Figure 3 shows some examples.

(**a**) (**b**)

Figure 3. In (**a**,**b**) some examples of art painting are reported.

Discussion

LIFE methods are compared on the dataset adopted in [26] and in terms of the Mean Reciprocal Rank (MRR). Being an experiment based on punctual features, the goal is to find a solution to the false positives problem during the matching phase, which is typical of this field. Table 1 provides the best results reached by the ARSRG. The improvement is connected to the topological relationships among features and the filtering over the complete set of features extracted from the image. Indeed, with the purpose to discard many false matches, descriptors belonging to regions are compared instead of the entire image, as proposed in standard approaches. The best values for the ρ parameter, which controls the tolerance of false matches both in graph matching and the ratio test, are found through a tuning

procedure, as described in [8,26] (0.6 and 0.7 were accepted in [26], while values greater than 0.8 were rejected in [8]; ρ values of 0.7 and 0.8 are optimal for the ARSRG matching).

Table 1. Quantitative comparison using the Mean Reciprocal Rank (MRR) measure among SIFT [8], SURF [29], ORB [30], FREAK [31], BRIEF [32], and the Attributed Relational SIFT-based Regions Graph (ARSRG) matching on the dataset in [26].

ρ	*SIFT*	*SURF*	*ORB*	*FREAK*	*BRIEF*	$\mathbf{ARSRG_{1st}}$	$\mathbf{ARSRG_{2nd}}$
0.6	0.7485	0.8400	0.6500	0.3558	0.4300	0.6700	0.6750
0.7	0.7051	0.6800	0.6116	0.3360	0.3995	0.7133	0.7500
0.8	0.6963	0.5997	0.5651	0.2645	0.4227	0.6115	0.8000

Precision and recall are adopted to measure the performance on the dataset in [27]. As can be seen in Table 2, the SIFT-based approach is better in terms of recall. Differently, the ARSRG matching yields comparable results with ρ equal to 0.8. In contrast, the ARSRG matching, in Table 3, is the best approach in terms of precision with ρ equal to 0.6, 0.7, and 0.8. The results are obtained as the consequence of the application of the image structural representation. Indeed, the ARSRG nodes, the image regions, provide a partitioning rule across the entire SIFT set. In this way, the subsets are selected separately during the processing. This strategy removes most of the false matches with the purpose to discard several images as candidates for the final ranking.

Table 2. Quantitative comparison, using the recall measure, among SIFT [8], SURF [29], ORB [30], FREAK [31], BRIEF [32], and the ARSRG matching on the dataset in [27].

ρ	*SIFT*	*SURF*	*ORB*	*FREAK*	*BRIEF*	$\mathbf{ARSRG_{1st}}$	$\mathbf{ARSRG_{2nd}}$
0.6	1.0	0.8666	0.8000	0.7333	0.7666	0.7333	0.7333
0.7	1.0	0.9000	0.8666	0.7333	0.8666	0.7666	0.7333
0.8	1.0	1.0	1.0	0.8333	1.0000	0.8000	0.8000

Table 3. Quantitative comparison using the precision measure, among SIFT [8], SURF [29], ORB [30], FREAK [31], BRIEF [32], and the ARSRG matching on the dataset in [27].

ρ	*SIFT*	*SURF*	*ORB*	*FREAK*	*BRIEF*	$\mathbf{ARSRG_{1st}}$	$\mathbf{ARSRG_{2nd}}$
0.6	0.0674	0.0820	0.2051	0.05584	0.10689	1.0	1.0
0.7	0.0401	0.0441	0.0742	0.04671	0.05664	1.0	1.0
0.8	0.0312	0.0338	0.0348	0.04072	0.03452	1.0	1.0

The datasets described in [26,28] are adopted for graph SIFT-based matching algorithms' comparisons (HGM [15], RRWGM [33], TM [34]). Results, in term of MRR, are reported in Tables 4 and 5. Again, the ARSRG reaches better results by adopting the region matching approach. It provides false matches' removal and hence improves the final results. The critical point of the graph matching problem concerns the correspondence rule among nodes to be associated with and belonging to different sets, represented by the images to be compared. In the standard case, the choice falls on two whole sets. Differently, the ARSRG provides partitioning and thinning of the main set of features, extracted from the entire image, providing an improvement in performance and execution time.

Table 4. Quantitative comparison, using the *MRR* measure, among the HGM [15], RRWGM [33], and TM [34] algorithms and the ARSRG matching on the dataset in [26].

HGM	**RRWGM**	**TM**	$\mathbf{ARSRG_{1st}}$	$\mathbf{ARSRG_{2nd}}$
0.2600	0.1322	0.1348	0.6115	1.0

Table 5. Quantitative comparison, using the *MRR* measure, among the HGM [15], RRWGM [33], and TM [34] algorithms and the ARSRG matching on the dataset in [28].

HGM	*RRWGM*	*TM*	$\mathbf{ARSRG_{1st}}$	$\mathbf{ARSRG_{2nd}}$
0.1000	0.0545	0.0545	0.20961	0.39803

Table 6 describes the performance comparison, on the dataset presented in [26] and in terms of MRR, with the Lucene Image Retrieval (LIRe) [35] system and related features: *MPEG7* [36], Tamura [37], *CEDD* [38], *FCTH* [39], *ACC* [40]. The LIRe system proves unsuitable for art paint retrieval as shown by the poor performance. This behavior consists of a bad discrimination of relevant and irrelevant images with a final ranking containing inadequate results. Differently, the ARSRG proves to be very suitable for the problem faced. Surely, the aspect that provides a surge in performance is linked to the dual information, local and structural, included in the ARSRG. In this way, the content of the image is described with respect to the relation of its parts. Otherwise, it happens with the features adopted by LIRe, which only provides localized information.

Table 6. Quantitative comparison using the *MRR* measure, among some features available in the Lucene Image Retrieval (LIRe) [35] system and the ARSRG matching on the dataset in [26].

MPEG7	*Tamura*	*CEDD*	*FCTH*	*ACC*	$\mathbf{ARSRG_{1st}}$	$\mathbf{ARSRG_{2nd}}$
0.2645	0.1885	0.2329	0.1924	0.1879	0.7133	0.7500

6.2. Graph Embedding

This section reviews the results previously obtained in [5]. The three datasets, different in size, design, and topic, adopted to test the ARSRG are described below:

1. The Columbia Image Database Library (COIL-100) [41] is composed of 100 objects. Each object is represented by 72 colored images that show it under different rotation points of view. The objects were located on a black background.
2. The Amsterdam Library Of Images (ALOI) [42] is a color image collection composed of 1000 small objects. In contrast to COIL-100, where the objects are cropped to fill the full image, in ALOI, the images contain the background and the objects in their original size. The objects were located on a black background.
3. The ETH-80 [43] is composed of 80 objects from eight categories, and each object is represented by 41 different views, thus obtaining a total of 3280 images. The objects were located on a uniform background.

Figure 4 reports some examples of objects.

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 4. Example images from the Columbia Image Database Library (COIL-100) dataset (**a**,**b**), the Amsterdam Library Of Images (ALOI) dataset (**c**,**d**), and the ETH-80 dataset (**e**,**f**).

Discussion

Results obtained on the ETH-80 database and the setup related to [44] are summarized in Table 7. The training set is composed of 240 images, for each category (apples, cars, cows, cups, horses, and tomatoes), 4 objects, and for each object, 10 different views. The testing set is composed of the

remaining images, 60 per category (15 views per object). The results are achieved by Logistic Label Propagation (*LLP*) [45] + Bag of Words (BoW) [46]) and those described in [44] by applying the approaches in [47] (gdFil), in [48] (APGM), and in [49] (VEAM). The best performance, in Table 7, is achieved by the ARSRG embedding adopting the *LLP* classifier. This case confirms that the ARSRG embedding correctly deals with object view changes.

Table 7. Recognition accuracy on the ETH-80 database.

Method	Accuracy
LLP+ARSRG*emb*	89.26%
LLP+BoW	58.83%
gdFil	47.59%
APGM	84.39%
VEAM	82.68%

Results obtained on the COIL-100 database with the setup related to [44,50] are summarized in Table 8. In particular, the training set is composed of 11% of 25 of objects randomly selected, and the remaining ones are the testing set. The results are achieved by Logistic Label Propagation (*LLP*) [45] + Bag of Words (BoW) [46]) and those described in [44,50] by applying their approach (VFSR) and the approaches proposed in [47] (gdFil), in [48] (APGM), in [49] (VEAM), in [51] (DTROD-AdaBoost), in [52] (RSW+boosting), in [53] (sequential patterns), and in [54] (LAF). In this case as well, the accuracy of ARSRG embedding confirms its qualities.

Table 8. Recognition accuracy on the COIL-100 database.

Method	Accuracy
LLP+ARSRG*emb*	99.55%
LLP+BoW	51.71%
gdFil	32.61%
VFSR	91.60%
APGM	99.11%
VEAM	99.44%
DTROD-AdaBoost	84.50%
RSW+Boosting	89.20%
Sequential Patterns	89.80%
LAF	99.40%

Results obtained on the ALOI database with the setup related to [55] are summarized in Table 9. In particular, only the first 100 objects are adopted. Color images are converted to gray levels, and for training, the second image of each class is adopted and the remaining for testing. A total of 200 images are obtained, considering two images of each class. During each iteration, for each class, one additional training image is attached. Table 9 shows only the results considering a batch of 400 images since the intermediate results do not provide great differences. The results achieved by baseline Logistic Label Propagation (*LLP*) [45] + Bag of Words (BoW) [46] and those obtained in [55] by applying some variants of Linear Discriminant Analysis (ILDAaPCA, batchLDA, ILDAonK, and ILDAonL) are reported. As can be seen, *LLP*+ARSRG*emb* performs with a small training set, and it is little affected by overfitting problems.

Table 9. Recognition accuracy on the ALOI database.

Method	200	400	800	1200	1600	2000	2400	2800	3200	3600
LLP+ARSRG*emb*	86.00%	90.00%	93.00%	96.00%	95.62%	96.00%	88.00%	81.89%	79.17%	79.78%
LLP+BoW	49.60%	55.00%	50.42%	50.13%	49.81%	48.88%	49.52%	49.65%	48.96%	49.10%
batchLDA	51.00%	52.00%	62.00%	62.00%	70.00%	71.00%	74.00%	75.00%	75.00%	77.00%
ILDAaPCA	51.00%	42.00%	53.00%	48.00%	45.00%	50.00%	51.00%	49.00%	49.00%	50.00%
ILDAonK	42.00%	45.00%	53.00%	48.00%	45.00%	51.00%	51.00%	49.00%	49.00%	50.00%
ILDAonL	51.00%	52.00%	61.00%	61.00%	65.00%	69.00%	71.00%	70.00%	71.00%	72.00%

Moreover, capturing local information and preserving the spatial relationships among them provide a strong improvement of performance in the object recognition field. The only computational overhead concerns the building of graph-based representation, while the classification can be performed very quickly. Dimensionality reduction is of great importance, that is the transition from a graph to a vector space. Essentially, there are three goals of embedding: first, to capture the graph topology, node-to-node relationship, and other relevant information about the subgraphs of the main structure; second, to reduce the speed of the process regardless of the size of the graph (graphs are generally large, and a good approach must be efficient); third, to decide on the right dimensionality. A longer vector representation retains more information while inducing greater complexity over time and space than a more organized approach. It is important to find a balance according to the requirements. In this case, the method first tries to preserve all the structural properties of ARSRGs and, second, to take advantage of the tools included in the destination space. The embedding process allows representing and analyzing information more easily as the vector space includes many processing tools compared to the starting space.

6.3. Bag of ARSRG Words

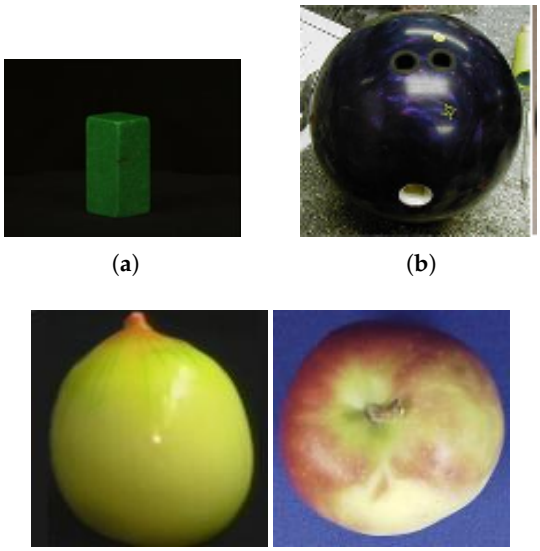
This section reviews the results previously obtained in [6], named Bag of ARSRG Words (BoAW). Same datasets described in Section 6.2 are adopted and, in addition, the dataset reported below:

- Caltech 101 [56]: This is an object image collection composed of 101 categories, with about 40 to 800 images per category. Most categories have about 50 images.

The classification stage is less difficult on the ALOI, COIL-100, and ETH-80 datasets because the objects are represented on a simple background, unlike the Caltech 101 dataset, where images have an uneven background. Figure 5 reports some examples of objects.

Discussion

The following setup, reported in [55] and the same as Table 9, is adopted: *LLP* [45] for classification stage, the One-versus-All (OvA) paradigm for 30 executions, the shuffling operation on the training and test set, image scaling on a size of 150 $\times$ 150 pixels. Table 10 reports the accuracy results on the ALOI dataset and achieved by Bag of Visual Words (BoVW) [46] and those obtained in [55] using some variants of linear discriminant analysis (ILDAaPCA, batchLDA, ILDAonK, and ILDAonL) and in [5] (ARSRG*emb*).



(**a**) (**b**)

(**c**) (**d**)

Figure 5. Dataset images: (**a**) ALOI, (**b**) Caltech 101, (**c**) COIL-100, and (**d**) ETH-80.

Table 10. Results on the ALOI dataset. BoAW, Bag of ARSRG Words; BoVW, Bag of Visual Words.

Method	200	400	800	1200	1600	2000	2400	2800	3200	3600
BoAW	**98.29%**	92.83%	98.80%	96.80%	96.76%	98.15%	89.52%	82.65%	79.96%	79.88%
ARSRG*emb*	86.00%	90.00%	93.00%	96.00%	95.62%	96.00%	88.00%	81.89%	79.17%	79.78%
BoVW	49.60%	55.00%	50.42%	50.13%	49.81%	48.88%	49.52%	49.65%	48.96%	49.10%
batchLDA	51.00%	52.00%	62.00%	62.00%	70.00%	71.00%	74.00%	75.00%	75.00%	77.00%
ILDAaPCA	51.00%	42.00%	53.00%	48.00%	45.00%	50.00%	51.00%	49.00%	49.00%	50.00%
ILDAonK	42.00%	45.00%	53.00%	48.00%	45.00%	51.00%	51.00%	49.00%	49.00%	50.00%
ILDAonL	51.00%	52.00%	61.00%	61.00%	65.00%	69.00%	71.00%	70.00%	71.00%	72.00%

The best performance is reached by BoAW, which is able to adapt to object the recognition task. Indeed, the main contribution is the combination of local and spatial information, which improves the phases of image representation and matching.

Results obtained on the Caltech 101 dataset, for particular image categories (bowling, cake, calculator, cannon, cd, chess-board, joy-stick, skateboard, spoon, and umbrella) and comparing with BoVW based on pyramidal representation [46], are summarized in Table 11. The best average accuracy is obtained with a split of 60/40% training and test set, respectively.

Table 11. Results on the Caltech 101 dataset.

Method	Accuracy
BoAW	74.00%
BoVW	83.00%

It is easy to notice that the performance when images are composed of non-uniform backgrounds is different. Due to this aspect, which distorts image representation and consequently affects the classification phase, BoVW is more powerful than BoAW. A possible solution could be a segmentation step, during the preprocessing, to isolate the uninformative background, with the purpose to work exclusively on the object to be represented. This loophole is not always efficient because removing the background in some cases is tricky. Table 12 provides the average accuracy on the COIL-100 dataset based on the same setup of Table 8. Therefore, the results are related to BoVW and those obtained in [44,50] by applying their solution (VFSR) and the approaches proposed in [47] (gdFil),

in [48] (APGM), in [49] (VEAM), in [51] (DTROD-AdaBoost), in [52] (RSW+boosting), in [53] (sequential patterns), in [54] (LAF), and in [5] (ARSRG*emb*). Furthermore, this experiment confirms that BoAW provides the best performance.

Table 12. Results on the COIL-100 dataset.

Method	Accuracy
BoAW	**99.77%**
ARSRG*emb*	99.55%
BoVW	51.71%
gdFil	32.61%
VFSR	91.60%
APGM	99.11%
VEAM	99.44%
DTROD-AdaBoost	84.50%
RSW+Boosting	89.20%
Sequential Patterns	89.80%
LAF	99.40%

Table 13 shows the results on the ETH-80 dataset and the setup related to Table 7. In addition to BoVW, accuracy is related to the tests achieved in [44] by employing the solution proposed in [5] (ARSRG*emb*), [47] (gdFil), in [48] (APGM), and in [49] (VEAM). As can be seen, the view point changes case does not affect the performance of BoAW compared to the competitors.

As can be noted, ARSRG is suitable for particular types of images where the background is uniform. Specifically, the object to be represented can be considered as the foreground, while the background represents irrelevant information. Feature points are not detected, or just a few, in scanning the background of the image. The background is divided from the foreground, through a filtering step, also because it usually does not contain distinctive feature points useful for object coding. This procedure is always effective except for the Caltech 101 dataset, in which it fails, in some cases, as shown by the performance.

Table 13. Results on the ETH-80 dataset.

Method	Accuracy
BoAW	89.29%
ARSRG*emb*	89.26%
BoVW	58.83%
gdFil	47.59%
APGM	84.39%
VEAM	82.68%

6.4. Kernel Graph Embedding

This section reviews the results previously obtained in [7]. The classification performance through Support Vector Machine (SVM) and Asymmetric Kernel Scaling (AKS) [57] over the standard OvA setup on low, medium, and high imbalanced image classification problems is tested, with art painting classification application [58]. The datasets adopted are the same described in Section 6.1. Tables 14 and 15 show the settings for the classification problems. Notice that the last column includes the Imbalance Rate (IR), the ratio of the percentage of images belonging to the majority class and the minority class, calculated through Equation (5).

$$IR = \frac{\%maj}{\%min} \tag{5}$$

Table 14. One-versus-All (OvA) configuration for the dataset in [26].

Problem	Classification Problem	(%min, %maj)	IR
1	Artemisia vs. all	(3.00, 97.00)	32.33
2	Bathsheba vs. all	(3.00, 97.00)	32.33
3	Danae vs. all	(12.00, 88.00)	7.33
4	Doctor_Nicolaes vs. all	(3.00, 97.00)	32.33
5	HollyFamilly vs. all	(2.00, 98.00)	49.00
6	PortraitOfMariaTrip vs. all	(3.00, 97.00)	32.33
7	PortraitOfSaskia vs. all	(1.00, 99.00)	99.00
8	RembrandtXXPortrai vs. all	(2.00, 98.00)	49.00
9	SaskiaAsFlora vs. all	(3.00, 97.00)	32.33
10	SelfportraitAsStPaul vs. all	(8.00, 92.00)	11.50
11	TheJewishBride vs. all	(4.00, 96.00)	24.00
12	TheNightWatch vs. all	(9.00, 91.00)	10.11
13	TheProphetJeremiah vs all	(7.00, 93.00)	13.28
14	TheReturnOfTheProdigalSon vs. all	(9.00, 91.00)	10.11
15	TheSyndicsoftheClothmakersGuild vs. all	(5.00, 95.00)	19.00
16	Other vs. all	(26.00, 74.00)	2.84

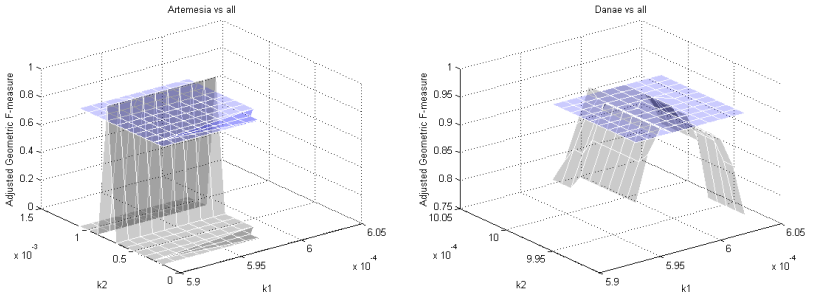
Table 15. The OvA configuration for the dataset in [4].

Problem	Classification Problem	(%min, %maj)	IR
1	Class 4 vs. all	(1.00, 9.00)	9.00
2	Class 7 vs. all	(1.00, 9.00)	9.00
3	Class 8 vs. all	(1.00, 9.00)	9.00
4	Class 13 vs. all	(1.00, 9.00)	9.00
5	Class 15 vs. all	(1.00, 9.00)	9.00
6	Class 19 vs. all	(1.00, 9.00)	9.00
7	Class 21 vs. all	(1.00, 9.00)	9.00
8	Class 27 vs. all	(1.00, 9.00)	9.00
9	Class 30 vs. all	(1.00, 9.00)	9.00
10	Class 33 vs. all	(1.00, 9.00)	9.00

Discussion

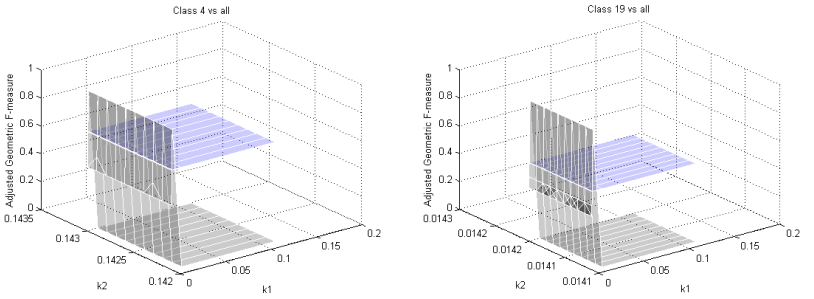
AKS and standard SVM are compared in terms of the adjusted F-measure [59]. It can be seen in Figure 6 that AKS outperforms standard SVM, and in order to reach noteworthy performance, a fine tuning is needed. Differently, in Figure 7, the performance presents only a single peak of exceedance with respect to SVM. Further comparisons have been performed with C4.5 [60], RIPPER [61], L2 loss SVM [62], L2 regularized logistic regression [63], and Ripple-Down Rule learner (RDR) [64] on OvA classification problems. Due to the distortion introduced by the imbalance rates, the results related to the datasets are different. The dataset in [26], in which the configuration includes approximately low, medium, and high rates, is great for a robust testing phase because it covers full cases of class imbalance problems. Differently, for the dataset in [4], the imbalance rates are identical for all configurations. Results are reported in Table 16, for the dataset in [26], and Table 17, for the dataset in [4]. It can be noted that the performances are significantly higher than the competitors. In particular, the main improvement, provided by AKS, concerns the accuracy of the classification of patterns belonging to the minority class, positive, which, during the relevance feedback evaluation, have a greater weight. Indeed, these latter are difficult to classify compared to patterns belonging to the majority class, negative. The results reach a high level of correct classification due to two aspects. The first involves the vector-based image representation, KGEARSRG, adopted. Graph kernels aim at bridging the gap between the high representational power and the flexibility of graphs in terms of feature vector representation. KGEARSRG provides a fixed-dimensional vector space image representation in order to process the data for classification purposes. The second concerns the AKS method for the classification stage. It has the intrinsic ability to more efficiently address classification problems that

are extremely imbalanced. In other words, the AKS classifier retains the ability to correctly recognize patterns originating from the minority class compared to the majority class.



(**a**) (**b**)

Figure 6. Parameter Choice 1. The x and y axes represent the values of the parameters of the two methods, while on the z axis is plotted the Adjusted F-measure (AGF) for two of the OvA configurations of the dataset in [26]: (**a**) Artemisia vs. all and (**b**) Danae vs. all. The gray and blue surfaces represent, respectively, the results with the Asymmetric Kernel Scaling (AKS) and SVM classifiers.



(**a**) (**b**)

Figure 7. Parameter Choice 2. The x and y axes represent the values of the parameters of the two methods, while on the z axis is plotted the AGF for two of the OvA configurations on the dataset in [4]: (**a**) Class 4 vs. all and (**b**) Class 19 vs. all. The gray and blue surfaces represent, respectively, the results with the AKS and SVM classifiers.

Table 16. Comparison results on the dataset in [26] and Table 14. RDR, Ripple-Down Rule learner.

			AGF			
Problem	**AKS**	**C4.5**	**RIPPER**	**L2-L SVM**	**L2 RLR**	**RDR**
1	0.9414	0.5614	0.8234	0.6500	0.5456	0.8987
2	0.9356	0.8256	0.6600	0.8356	0.8078	0.7245
3	0.9678	0.8462	0.8651	0.4909	0.6123	0.7654
4	0.9746	0.8083	0.6600	0.4790	0.4104	0.6693
5	0.9654	0.7129	0.9861	0.8456	0.4432	0.6134
6	0.9342	0.5714	0.9525	0.8434	0.9525	0.5554
7	0.9567	0.6151	0.7423	0.5357	0.4799	0.6151
8	0.8345	0.4123	0.3563	0.7431	0.5124	0.7124
9	0.9435	0.9456	0.9456	0.8345	0.6600	0.6600
10	0.8456	0.4839	0.5345	0.4123	0.4009	0.5456
11	0.9457	0.9167	0.9088	0.9220	0.8666	0.9132
12	0.6028	0.5875	0.5239	0.4124	0.4934	0.5234
13	0.8847	0.7357	0.6836	0.7436	0.7013	0.5712
14	0.9376	0.9376	0.8562	0.8945	0.8722	0.8320
15	0.9765	0.8630	0.8897	0.8225	0.7440	0.8630
16	0.7142	0.5833	0.3893	0.4323	0.5455	0.5111

Table 17. Comparison results on the dataset in [4] and Table 15.

			AGF			
Problem	**AKS**	**C4.5**	**RIPPER**	**L2-L SVM**	**L2 RLR**	**RDR**
1	0.9822	0.6967	0.5122	0.4232	0.4322	0.6121
2	0.9143	0.5132	0.4323	0.4121	0.4212	0.5323
3	0.9641	0.4121	0.4211	0.4213	0.3221	0.4323
4	0.9454	0.4332	0.1888	0.4583	0.3810	0.3810
5	0.9554	0.3810	0.2575	0.5595	0.3162	0.6967
6	0.9624	0.3001	0.1888	0.1312	0.3456	0.3121
7	0.9344	0.3810	0.5566	0.4122	0.4455	0.2234
8	0.9225	0.4333	0.1112	0.2575	0.1888	0.1888
9	0.9443	0.6322	0.1888	0.1888	0.6122	0.6641
10	0.9653	0.1897	0.5234	0.6956	0.1888	0.1121

7. Conclusions

In this paper, Attributed Relational SIFT-based Regions Graph (ARSRG), a structure for image representation, is explored through the description and analysis of new aspects. Starting from previous works and performing a thorough study, theoretical notions are introduced in order to clarify and deepen the structural design of the ARSRG. It is demonstrated how the ARSRG can be adopted in disparate fields such as graph matching, graph embedding, bag of graph words, and kernel graph embedding with the applications of object recognition and art painting retrieval/classification. The experimental results amply show how the performances on different datasets are better than state-of-the-art competitors. Future developments certainly include the exploration of additional application fields, the introduction of additional algorithms (mainly graph matching) to improve performance comparison, and a greater enrichment of image features to include within the ARSRG.

Funding: This research received no external funding.

Acknowledgments: This work is dedicated to Alfredo Petrosino. With him, I took my first steps in the field of computer science. During these years spent together, I learned firmness in achieving goals and love and passion for the work. I will be forever grateful. Thank you my great master.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Love, B.C.; Rouder, J.N.; Wisniewski, E.J. A structural account of global and local processing. *Cogn. Psychol.* **1999**, *38*, 291–316. [CrossRef] [PubMed]
2. Koffka, K. *Principles of Gestalt Psychology;* Routledge: Abingdon, UK, 2013.
3. Liu, Y.; Zhang, D.; Lu, G.; Ma, W.Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognit.* **2007**, *40*, 262–282. [CrossRef]
4. Manzo, M.; Petrosino, A. Attributed relational sift-based regions graph for art painting retrieval. In Proceedings of the International Conference on Image Analysis and Processing, Naples, Italy, 9–13 September 2013; pp. 833–842.
5. Manzo, M.; Pellino, S.; Petrosino, A.; Rozza, A. A novel graph embedding framework for object recognition. In Proceedings of the European Conference on Computer Vision, Zürich, Switzerland, 6–12 September 2014; pp. 341–352.
6. Manzo, M.; Pellino, S. Bag of ARSRG Words (BoAW). *Mach. Learn. Knowl. Extr.* **2019**, *1*, 871–882. [CrossRef]
7. Manzo, M. KGEARSRG: Kernel Graph Embedding on Attributed Relational SIFT-Based Regions Graph. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 962–973. [CrossRef]
8. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]

9. Sanromà, G.; Alquézar, R.; Serratosa, F. Attributed graph matching for image-features association using SIFT descriptors. In *Structural, Syntactic, and Statistical Pattern Recognition*; Springer: Berlin, Germany, 2010; pp. 254–263.
10. Sanroma, G.; Alquézar, R.; Serratosa, F. A discrete labelling approach to attributed graph matching using SIFT features. In Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; pp. 954–957.
11. Duchenne, O.; Joulin, A.; Ponce, J. A graph-matching kernel for object categorization. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 1792–1799.
12. Cho, M.; Lee, K.M. Progressive graph matching: Making a move of graphs via probabilistic voting. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–24 June 2012; pp. 398–405.
13. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [CrossRef]
14. Mikolajczyk, K.; Schmid, C. Scale & affine invariant interest point detectors. *Int. J. Comput. Vis.* **2004**, *60*, 63–86.
15. Lee, J.; Cho, M.; Lee, K.M. Hyper-graph matching via reweighted random walks. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 1633–1640.
16. Revaud, J.; Lavoué, G.; Ariki, Y.; Baskurt, A. Learning an efficient and robust graph matching procedure for specific object recognition. In Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; pp. 754–757.
17. Romero, A.; Cazorla, M. Topological slam using omnidirectional images: Merging feature detectors and graph-matching. In *Advanced Concepts for Intelligent Vision Systems*; Springer: Berlin, Germany, 2010; pp. 464–475.
18. Deng, Y.; Manjunath, B. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 800–810. [CrossRef]
19. Xia, S.; Hancock, E. 3d object recognition using hyper-graphs and ranked local invariant features. In *Structural, Syntactic, and Statistical Pattern Recognition*; Springer: Berlin, Germany, 2008; pp. 117–126.
20. Hori, T.; Takiguchi, T.; Ariki, Y. Generic object recognition by graph structural expression. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 1021–1024.
21. Luo, M.; Qi, M. A New Method for Cartridge Case Image Mosaic. *J. Softw.* **2011**, *6*, 1305–1312. [CrossRef]
22. Trémeau, A.; Colantoni, P. Regions adjacency graph applied to color image segmentation. *IEEE Trans. Image Process.* **2000**, *9*, 735–744. [CrossRef]
23. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630. [CrossRef] [PubMed]
24. Liu, Y.; Zhang, D.; Lu, G.; Ma, W.Y. Region-based image retrieval with perceptual colors. In *Advances in Multimedia Information Processing-PCM 2004*; Springer: Berlin, Germany, 2004; pp. 931–938.
25. Sanromà Güell, G.; Alquézar Mancho, R.; Serratosa Casanelles, F. Graph matching using SIFT descriptors—An application to pose recovery of a mobile robot. In Proceedings of the Fifth International Conference on Computer Vision Theory and Applications, Angers, France, 17–21 May 2010; pp. 249–254.
26. Haladová, Z.; Šikudová, E. Limitations of the SIFT/SURF based methods in the classifications of fine art paintings. *Comput. Graph. Geom.* **2010**, *12*, 40–50.
27. Chang, C.; Etezadi-Amoli, M.; Hewlett, M. A Day at the Museum. 2009. Available online: http://www.stanford.edu/class/ee368/Project07/reports/ee368group06.pdf (accessed on 6 August 2020).
28. Ruf, B.; Kokiopoulou, E.; Detyniecki, M. Mobile museum guide based on fast SIFT recognition. In *International Workshop on Adaptive Multimedia Retrieval*; Springer: Berlin, Germany, 2008; pp. 170–183.
29. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 404–417.
30. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G.R. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; Volume 11, p. 2.

31. Alahi, A.; Ortiz, R.; Vandergheynst, P. Freak: Fast retina keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 510–517.
32. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, 5–11 September 2010; pp. 778–792.
33. Cho, M.; Lee, J.; Lee, K.M. Reweighted random walks for graph matching. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, 5–11 September 2010; pp. 492–505.
34. Duchenne, O.; Bach, F.; Kweon, I.S.; Ponce, J. A tensor-based algorithm for high-order graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2383–2395. [CrossRef] [PubMed]
35. Lux, M.; Chatzichristofis, S.A. Lire: Lucene image retrieval: an extensible java cbir library. In Proceedings of the 16th ACM International Conference on Multimedia, Vancouver, BC, Canada, 27–31 October 2008; pp. 1085–1088.
36. Chang, S.F.; Sikora, T.; Purl, A. Overview of the MPEG-7 standard. *IEEE Trans. Circuits Syst. Video Technol.* **2001**, *11*, 688–695. [CrossRef]
37. Tamura, H.; Mori, S.; Yamawaki, T. Textural features corresponding to visual perception. *IEEE Trans. Syst. Man Cybern.* **1978**, *8*, 460–473. [CrossRef]
38. Chatzichristofis, S.A.; Boutalis, Y.S. CEDD: Color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In Proceedings of the International Conference on Computer Vision Systems, Santorini, Greece, 12–15 May 2008; pp. 312–322.
39. Chatzichristofis, S.A.; Boutalis, Y.S. Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval. In Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services, Klagenfurt, Austria, 7–9 May 2008; pp. 191–196.
40. Huang, J.; Kumar, S.; Mitra, M.; Zhu, W.J.; Zabih, R. *Image Indexing Using Color Correlograms;* 1997; Volume 97, p. 762. Avialable online: http://www.cs.cornell.edu/~rdz/Papers/Huang-CVPR97.pdf (accessed on 6 August 2020).
41. Nayar, S.K.; Nene, S.A.; Murase, H. *Columbia Object Image Library (Coil 100)*; Technical Report No. CUCS-006-96; Columbia University: New York, NY, USA, 1996.
42. Geusebroek, J.M.; Burghouts, G.J.; Smeulders, A.W. The Amsterdam library of object images. *Int. J. Comput. Vis.* **2005**, *61*, 103–112. [CrossRef]
43. Leibe, B.; Schiele, B. Analyzing appearance and contour based methods for object categorization. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 16–22 June 2003; Volume 2, pp. 2–409.
44. Morales-González, A.; Acosta-Mendoza, N.; Gago-Alonso, A.; García-Reyes, E.B.; Medina-Pagola, J.E. A new proposal for graph-based image classification using frequent approximate subgraphs. *Pattern Recognit.* **2014**, *47*, 169–177. [CrossRef]
45. Kobayashi, T.; Watanabe, K.; Otsu, N. Logistic label propagation. *Pattern Recognit. Lett.* **2012**, *33*, 580–588. [CrossRef]
46. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2169–2178.
47. Gago-Alonso, A.; Carrasco-Ochoa, J.A.; Medina-Pagola, J.E.; Martínez-Trinidad, J.F. Full duplicate candidate pruning for frequent connected subgraph mining. *Integr. Comput.-Aided Eng.* **2010**, *17*, 211–225. [CrossRef]
48. Jia, Y.; Zhang, J.; Huan, J. An efficient graph-mining method for complicated and noisy data with real-world applications. *Knowl. Inf. Syst.* **2011**, *28*, 423–447. [CrossRef]
49. Acosta-Mendoza, N.; Gago-Alonso, A.; Medina-Pagola, J.E. Frequent approximate subgraphs as features for graph-based image classification. *Knowl.-Based Syst.* **2012**, *27*, 381–392. [CrossRef]
50. Morales-González, A.; García-Reyes, E.B. Simple object recognition based on spatial relations and visual features represented using irregular pyramids. *Multimed. Tools Appl.* **2013**, *63*, 875–897. [CrossRef]
51. Wang, Y.; Gong, S. Tensor discriminant analysis for view-based object recognition. In Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, China, 20–24 August 2006; Volume 3, pp. 33–36.

52. Marée, R.; Geurts, P.; Piater, J.; Wehenkel, L. Decision trees and random subwindows for object recognition. In *ICML Workshop on Machine Learning Techniques for Processing Multimedia Content (MLMM2005)*; University of Liege: Liege, Belgium, 2005.
53. Morioka, N. Learning object representations using sequential patterns. In *AI 2008: Advances in Artificial Intelligence*; Springer: Berlin, Germany, 2008; pp. 551–561.
54. Obdrzalek, S.; Matas, J. Object Recognition using Local Affine Frames on Distinguished Regions. In Proceedings of the British Machine Vision Conference 2002, Cardiff, UK, 2–5 September 2002; Volume 2, pp. 113–122.
55. Uray, M.; Skocaj, D.; Roth, P.M.; Bischof, H.; Leonardis, A. Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches. In Proceedings of the British Machine Vision Conference 2007, Warwick, UK, 10–13 September 2007; pp. 1–10.
56. Li, F.F.; Fergus, R.; Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.* **2007**, *106*, 59–70.
57. Maratea, A.; Petrosino, A. Asymmetric kernel scaling for imbalanced data classification. In *International Workshop on Fuzzy Logic and Applications*; Springer: Berlin, Germany, 2011; pp. 196–203.
58. Čuljak, M.; Mikuš, B.; Jež, K.; Hadjić, S. Classification of art paintings by genre. In Proceedings of the 2011 34th International Convention MIPRO, Opatija, Croatia, 23–27 May 2011; pp. 1634–1639.
59. Maratea, A.; Petrosino, A.; Manzo, M. Adjusted F-measure and kernel scaling for imbalanced data learning. *Inf. Sci.* **2014**, *257*, 331–341. [CrossRef]
60. Quinlan, J.R. *C4. 5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
61. Cohen, W.W. Fast effective rule induction. In *Machine Learning Proceedings 1995*; Elsevier: Amsterdam, The Netherlands, 1995; pp. 115–123.
62. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152.
63. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.
64. Dazeley, R.; Warner, P.; Johnson, S.; Vamplew, P. The Ballarat incremental knowledge engine. In *Pacific Rim Knowledge Acquisition Workshop*; Springer: Berlin, Germany, 2010; pp. 195–207.