Learning Relationship-aware Visual Features

Nicola Messina, Giuseppe Amato, Fabio Carrara,
Fabrizio Falchi, and Claudio Gennaro

ISTI-CNR, via G. Moruzzi 1, 56124 Pisa, Italy
`<firstname>.<lastname>@isti.cnr.it`

Abstract. Relational reasoning in Computer Vision has recently shown impressive results on visual question answering tasks. On the challenging dataset called CLEVR, the recently proposed Relation Network (RN), a simple plug-and-play module, has obtained state-of-the-art performance answering relational questions. In this paper, we define a sub-field of Content-Based Image Retrieval (CBIR) in which we are interested in retrieving images with given relationships among objects. To this aim, we employ the RN architecture in order to extract relation-aware features from CLEVR images. To prove the effectiveness of these features, we extended both CLEVR and Sort-of-CLEVR generating a ground-truth for CBIR from scene graphs. Experiments show that this approach outperforms the RMAC state-of-the-art features on this specific task. Furthermore, we propose a modification of the RN module – a two-stage Relation Network (2S-RN) – that allows us to extract relation-aware features by using a preprocessing stage able to encode in a relational fashion the image content. Experiments show that our proposed features produce an overall better ranking on the same ground-truth.

Keywords: CLEVR, Content-Based Image Retrieval, Deep Learning, Relational Reasoning, Relation Networks, Deep Features

1 Introduction

Relational reasoning refers to a particular kind of reasoning process that is able to understand and process relations among multiple entities. In this regard, Krawczyk et al. [1] characterize relational reasoning as the human brain "unique capacity to reason about abstract relationships among items in our environment". Biological intelligence developed such reasoning capabilities during thousands of years of evolution: comparing objects is indeed a critical task since it triggers decisions that could influence the safety of the individual, hence the survival of the species.

In Computer Vision (CV), deep architectures obtain great performance at tasks such as classifying or recognizing objects; however, latest studies demonstrated the difficulties of such architectures to understand a complex scene, where *understand* means catching relations among objects to compare them in a spatial and temporal dimension, exactly as the biological intelligence would operate. In other words, differently from biological intelligence, as of now deep architectures can *perceive* with quite a good accuracy the world that surrounds us, but still cannot *understand* it very well.

In order to know if a particular architecture is really able to understand, hard work must be invested even only in the development of analysis methods that try to understand how the architecture is internally behaving. Starting from [2], VQA has been a very active task in the recent CV literature. Given an image and a natural language question about the image, the task is to provide an accurate natural language answer. Recently, a great interest has grown around the possibility to ask relational questions. In this scenario, questions regard the spatial arrangement of objects inside the image and the task is called Relation-oriented VQA (R-VQA). The most relevant works in this area have used CLEVR [3] for both training and testing.

The contribution of this paper is many-fold:

- we introduce the novel task of relation-oriented content-based image retrieval (R-CBIR);
- we extend the CLEVR diagnostic dataset with a benchmark intended to verify to what extent a CBIR system is able to retrieve similar images in terms of objects spatial arrangement;
- we propose a novel two-stage Relation Network that is able to produce state-of-the-art features on the newly defined task.

The rest of the paper is organized as follows. In Section 2, we review some of the related work. In Section 3, we define the novel Relational-CBIR task. In Section 4, we extend the CLEVR dataset in order to generate a R-CBIR ground-truth. Our novel two-stage Relation Network is presented in Section 5. In Section 6, we report the results of the experiments conducted using features extracted both from the original RN architecture and our proposed two-stage network solution. We make concluding remarks in Section 7. In the Appendix, we leave some in-depth details about the network setup, as well as the link to source code.

2 Related Work

Visual Relationship Detection Recent work has addressed the problem of visual relationships detection (VRD) in images in the form of triplets (*subject*, *predicate*, *object*), where *subject* and *object* are common objects present in an image, and *predicate* indicates a relationship between them out of a set of possible relationships containing verbs, prepositions, comparatives, etc.

Several datasets comprised of a large set of visual relations [4,5,6] have opened the way to approaches aimed to detect those kinds of relationships in images. In [7], a CRF model is used to ground relationships given in the form of a scene graphs to test images for image retrieval purposes. In [8], a spatial feature map is extracted from images through a CNN and then combined with an embedded natural language expression in order to produce a pixel-wise segmentation output relevant to the relational textual query.

In [5], each pair of (*subject*, *object*) proposals is scored using a visual appearance module and a language module; two CNNs are used respectively to identify the entities into play and to predict the presence of relationships between them, and a language prior is exploited to refine predictions using pre-trained word embeddings. In [6], authors presented strong yet flexible visual features that encode not only the appearance of the objects, but also explicitly encode their spatial configuration in terms of bounding box relative translation, overlap, size, and aspect ratio. This representation is then used together with language priors to assign a score to every relationship triplet.

In [9], a novel architecture for VRD is presented which integrates not only object appearance, spatial configurations, but also statistical relations between objects and relationship predicates, all in a deep neural network framework. Specifically, object categories and relationship predicates are not fused or combined together as in previous works, but their statistical relations are exploited to correctly predict relationships.

Notwithstanding approaches that solve VRD are able to detect relationships, they usually do not encode information about the relationships present in an image in a compact representation, but all possible relationships are combinatorially tested on prediction time. Recently, [10] implemented a large scale image retrieval system able to map textual triplets into visual ones (object-subject-relation inferred from the image) projecting them into a common space learned through a modified version of triplet-loss. Unlike our work, however, this system is unable to produce a compact relational descriptor for the entire image, since it only encodes relations under the form of triplets.

Visual Question Answering In contrast to VRD, in visual question answering (VQA) relationships among objects are often implicit, making it a more challenging task. However, the potentiality of deep learning approaches has led to various successful approaches that tackle VQA with a learnable end-to-end solution.

Early proposals simply concatenated question embeddings and visual features. This method constitutes the main building block behind solutions like CNN+BoW or CNN+LSTM [11]. Both methods use a CNN to analyze the image and produce visual features.

Stacked Attention (SA) layers [12] replace the raw embeddings concatenation with a simple but quite effective reasoning module. The first attention is obtained by combining embeddings from the linguistic pipeline with some of the features from the visual pipeline. The output is again combined with other image features to form the second attention.

In [13] and [14] authors propose a novel architecture specialized to think in a relational way. They introduced a particular layer called Relation Network (RN), which is specialized in comparing pairs of objects. Objects representations are learned by means of a four-layer CNN and the question embedding is generated through an LSTM. The overall architecture composed of CNN, LSTM and the RN can be trained fully end-to-end, and it is able to reach superhuman performances.

Other solutions [15,16] introduce compositional approaches, able to explicitly model the reasoning process by dynamically building a graph that states which operations must be carried out and in which order, to obtain the right answer.

Latest proposals [17] and [18] obtained state-of-the-art results on R-VQA. In particular, [17] used a conditional batch normalization module to inject question related features into the visual pipeline. Differently, [18] introduced a novel conditioning module called FiLM.

Work related to VQA is often far off from approaching CBIR tasks, with respect to works developed around VRD. Unlike experimental setups in [4,5,6,10], whose focus concentrates on the retrieval of specific relationships, our work aims at evaluating a relational descriptor defined for the full scene. [19] uses a very similar experimental setup to the one we introduced. It exploits the graph data associated with every image in order to produce a ranking goodness metric (nDCG) for evaluating the quality of the ranking produced for a given query.

3 Relational-CBIR

In this paper, we define a sub-field of Content-Based Image Retrieval (CBIR) in which we are interested in retrieving images with given relationships among objects. We call this task Relational-CBIR (R-CBIR). Typically, CBIR is performed extracting a compact descriptor from an image, namely *feature*, that is able to characterize the image.

The inclusion of relation-aware information in the learned representation is a must in order to perform R-CBIR: as it has been proved in the state-of-the-art Relation Network module, deep learning methods are able to perform relational reasoning. In doing so, the information about spatial relationships among objects is internally encoded and learned end-to-end. Hence, the main idea consists in extracting this learned relational embedding for use as a feature. If features were able to embed relational information, we can, for example, imagine comparing two city skylines not only looking at shapes, corners and architectonic elements, but also taking in consideration the three-dimensional arrangement of buildings and skyscrapers, that creates a spatial pattern that only a relational engine could figure out.

Thus, R-CBIR consists in the following: given a query image, find all images in a database containing elements spatially arranged in a similar way to respect the ones present in the query.

4 A Relational-CBIR Ground-Truth

Our major contribution consists in the introduction of a novel benchmark for the R-CBIR task. In order to evaluate the quality of any relational feature extracted from a relation-aware system, we compute a specific ground-truth, built by exploiting relational knowledge embedded into graphs (*scene-graphs*). The generation of the ground-truth, in fact, must rely on a formal and objective *a-priori* relational knowledge of the scene.

By carefully choosing a distance function between graphs, we are able to give a good estimation of the relational similarity between scenes. In order to accomplish this task, we need some datasets that include a formal and precise description of relations occurring inside the scene, so that a precise scene-graph can be derived. Synthetic datasets CLEVR and Sort-of-CLEVR perfectly fit these needs, since they come with rendered images automatically generated using a-priori built scene-graphs.

Besides the native availability of graph-structured data, we target synthetic datasets since evaluating a new retrieval method on a simpler and controlled environment is often a preferable choice than moving directly to bigger and more challenging datasets.

4.1 CLEVR

CLEVR [3] is a synthetic dataset composed of 3D rendered scenes. There are 100k rendered images, subdivided among training (70k), validation (15k) and test (15k) sets. The total number of questions is ~865k, again split among training (~700k), validation (~150k) and test (~15k).

The main concept behind CLEVR is the *scene*. A *scene* contains different simple shaped objects, with mixtures of colors, materials and sizes. There are cubes, spheres, cylinders, each one of which can have a color chosen among eight; they can be big or small, and they can be made of one of two different materials, metal or rubber. The *scene* is fully and uniquely described by a *scene graph*. The scene graph describes in a formal way all the relationships between objects.

The question is formulated under the form of a *functional program*. The answer to a question represented by its functional program on a scene is simply calculated by executing the functional program on the scene graph. Given scene graphs and functional programs, natural language questions and images must be generated. To this end, scene graphs are rendered to photo-realistic 3D scenes by using Blender, a free 3D software; instead, functional programs are converted to natural language expressions compiling some *templates* embedded in the dataset and written in English.

CLEVR dataset gives us way more control on the learning phase than other datasets present in literature. Information in each sample of the dataset is *complete* and *exclusive*. This means that no common-sense awareness is needed in order to correctly answer the questions. Answers can be given simply understanding the question and reasoning exclusively on the image, without needing external concepts.

4.2 Sort-of-CLEVR

Sort-of-CLEVR consists in a simplification of the original CLEVR dataset. It is created mainly for testing and debugging architectures that are designed to work with CLEVR. Thus, this dataset is composed of simpler building blocks with respect to the full CLEVR. Images, in fact, are simpler than 3D renders provided with the original dataset; they instead carry simple 2D scenes, consisting of a certain number of 2D shapes. Shapes can be circles or squares and come in different colors. Every object, however, is uniquely identified by its color.

Differently from the CLEVR dataset, this one splits the questions into two different subsets:

- **relational questions**, asking for the color or shape of the farthest or the nearest object with respect to the given one; example: *What is the shape of the object that is farthest from the gray object?*
- **non-relational questions**, involving specific attributes that characterize a single object, in particular the shape, or the absolute position of the object with respect to the overall scene; example: *What is the shape of the gray object?*

Questions are directly encoded into 11-dimensional vectors, so there is no need for LSTM modules processing natural language.

Even if this dataset seems extremely simple, it can help to spot out some architectural problems that inhibit the network to think in a relational way.

4.3 Scene graphs

The best way to formally describe relations inside a scene is by making use of *scene graphs*, already available both in CLEVR and Sort-of-CLEVR. More in details, a scene graph contains *nodes*, that account for objects occupying the scene and *edges*, that describe relations occurring among them. Every node or edge can be assigned a set of attributes that fully describe them.

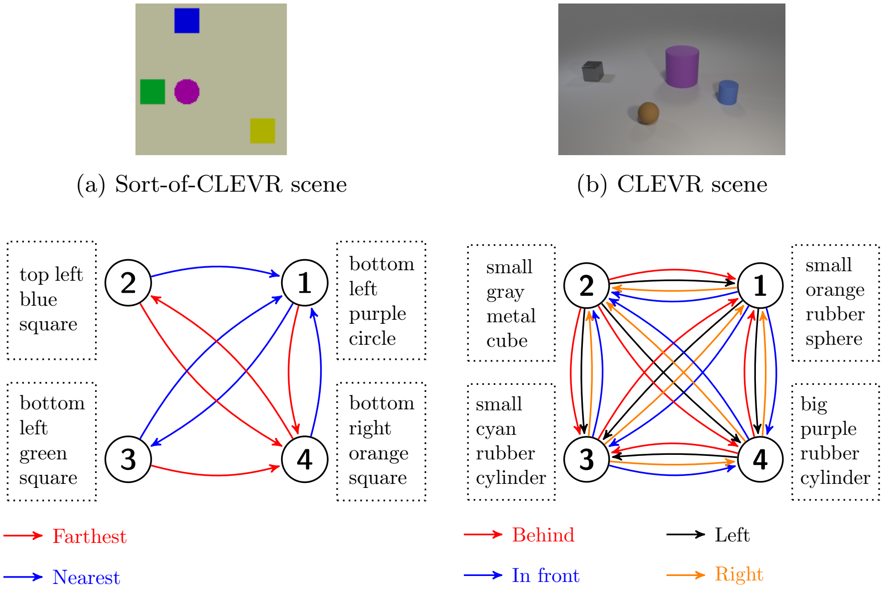
For Sort-of-CLEVR, nodes carry information regarding objects *color*, *shape* together with their absolute positions (*left/right* or *up/down* with respect to the scene). An edge, instead, carries information about the kind of relation it is describing. In Sort-of-CLEVR, an edge can refer to *farthest* and *nearest* relations.

Unlike the Sort-of-CLEVR case, CLEVR object attributes do not include absolute positions, since CLEVR deals uniquely with relational questions. Instead, possible attributes are the *color*, the *shape*, the *material* and the *size*. CLEVR also includes an higher number and more detailed spatial relations: *to the left of*, *to the right of*, *in front of*, *behind*.

In Figure 1, we report an example image for each dataset together with the associated scene-graphs. Note that, although CLEVR graph is complete, half of the edges can be removed without losing information, since *to the right of* implies an opposite edge *to the left of* and *in front of* implies an opposite edge *behind*. The ground-truth is composed of the entire set of scene graphs in the dataset. However, scene graphs must be comparable in order to be sorted. In other words, a certain kind of distance must be defined between them.

4.4 Ground-truth generation

Two scene graphs should be similar if they can depict almost the same relations between the same objects. However, evaluating the similarity between two graphs



(a) Sort-of-CLEVR scene

(b) CLEVR scene

(c) Sort-of-CLEVR scene graph

(d) CLEVR scene graph

Fig. 1: Example of scenes with associated scene-graphs

is not trivial; furthermore, it is often a subjective task, since there are aspects of the graph (e.g., the attributes associated to nodes) that weight differently, depending on the specific application.

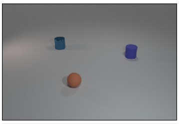
Although many solutions has been proposed in literature for defining distances between graph-structured data [20], concerning this particular use-case, we decided to employ the *graph edit-distance* (GED). It is an extension of the well-known edit-distance working on strings, that is defined as the shortest number of *delete*, *insert*, *substitute* operations needed in order to transform the source string into the target one.

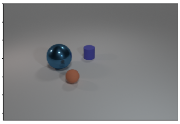
Differently from strings, edit operations on graphs include *delete*, *insert*, *substitute* for both nodes and edges, for a total of 6 edit operations. The problem is faced as an optimization problem. Since the GED problem is known to be computationally hard, in this work we will employ two different implementations [21,22]. [21] is an exact, non-approximated version of the GED algorithm. We will reference it as *Exact-GED*. While execution times are acceptable for Sort-of-CLEVR graph data, they become easily unworkable on CLEVR. For this reason, [22] is able to perform an efficient approximation of the algorithm. We will refer to this as *Approx-GED*. With respect to the exact version, Approx-GED does not consider the entire span of solutions. Instead, it looks for a tiny subset of edit sequences, obtained by first matching similar nodes using linear assignment and then matching edges on the ruled node pairing. Nevertheless, during experimentation, we measured that the resulting approximated ground-truth is perfectly comparable with the exact one.

An easy optimization on both exact and approximated GED computation on CLEVR consists, as previously discussed, in removing *behind* and *left* edges since they are redundant. Both implementations allow for the customization of the node-edge edit costs on the basis of their attributes. We applied the following policies for our application:

- nodes-edges insertion or deletion has always a cost of 1;
- edge substitution cost is 1 if edges do not belong to the same kind of relation, 0 otherwise;
- node substitution cost can be driven by two different policies:
  - *soft-match*: all attributes of a node weight equally during a substitution. So, considering a total of 4 attributes, if three attributes match the substitution cost is $3/4 = 0.75$. This is the fairest and most neutral solution since it does not prefer any attribute over all the others;
  - *hard-match*: the cost is 1 if at least one attribute value differs. It is 0 only if all attributes match.

To clarify GED algorithm functioning using our cost policies, we report below an example on CLEVR with *soft-match*. This instance of GED computation transforming the upper image into the below one will return a cost of 1.5.





Steps	Cost
1. Substitute node **small-cyan-metal-cylinder** with **big-cyan-metal-sphere** (change 2 attributes)	0.5
2. Substitute edge **small-cyan-metal-cylinder behind small-blue-rubber-cylinder** with **big-cyan-metal-sphere in front of small-blue-rubber-cylinder**	1.0

In the light of this, given a query, we compute the ground-truth ranking of the dataset by sorting all scenes using computed GED distances between scene graph of the query image and graphs from all the others. We will actually produce two ground-truths, the first generated using *soft-match* policy and the second using *hard-match*.

Given an image ranking produced by an arbitrary relation-aware system, a rank correlation metric is computed against the ground-truth ranking. In this work we will use the *Spearman-Rho* correlation index, that is a common ranking similarity measure often employed in information retrieval scenarios [23].

5 A Two-stage Relation Network

5.1 RN overview

We build upon the Relation Network (RN) module proposed in [13] in order to extract state-of-the-art features for the newly defined R-CBIR task.

RN obtained impressive results on relational tasks and in particular on CLEVR. RN modules combine input objects forming all possible pairs and applies a common transformation to them, producing activations aimed to store information about possible relationships among input objects.

For the specific task of VQA, authors used a four-layer CNN to learn visual object representations, that are then fed to the RN module and combined with the textual embedding of the question produced by an LSTM, conditioning the relationship information on the textual modality. The core of the RN module is given by the following:

$$r = \sum_{i,j} g_\theta(o_i, o_j, q) \tag{1}$$

where g_θ is a parametric functions whose parameters θ can be learned during the training phase. Specifically, it is a multi-layer perceptrons (MLP) network. o_i and o_j are the objects forming the pair under consideration and q is the question embedding vector obtained from the LSTM module.

The overall architecture composed of CNN, LSTM and the RN can be trained fully end-to-end and it is able to reach superhuman performances.

In this work, we employed this architecture in order to extract visual relational features. Relation-aware features useful for CBIR should be extracted from a stage inside the network still not conditioned to the question. Hence, valid CBIR features can be extracted from the original RN module only at the output of the convolutional layer, since, after that, questions condition entirely the remaining pipeline. Inspired by the state-of-the-art works on CBIR [24,25], we obtain an overall description for the image aggregating all object pair features in output from the CNN. More in details, we extracted $avg_{i,j}([o_i, o_j])$ and $max_{i,j}([o_i, o_j])$, where $[\cdot, \cdot]$ denotes concatenation. We will show in Section 6 that these features already embed relational knowledge able to defeat state-of-the-art CBIR solutions on this task. Also, we will use these features as baseline for evaluating the novel two-stage approach we introduced.

5.2 Two-stage RN (2S-RN)

The two-stage pipeline is aimed at decoupling visual relationships processing (*first-stage*) from the question elaboration (*second-stage*) so that activations from a layer in the first stage can be employed as visual relation-aware features.

Our contribution consists in the following: first, we consider all possible relations between objects $g_\theta(o_i, o_j)$ in the image. This is what we denoted as *first-stage*. The output from this stage is a representation of the relationships between objects in the image not conditioned on the question. Then, we combine the obtained relational representations $r_{i,j} = g_\theta(o_i, o_j)$ with the query embedding q as follows:

$$r = \sum_{i,j} h_\psi(r_{i,j}, q) = \sum_{i,j} h_\psi(g_\theta(o_i, o_j), q) \tag{2}$$

where h_ψ is the *second-stage*. It is a multi-layer perceptron network with parameters ψ. Using this solution, we constrained the network to learn relational

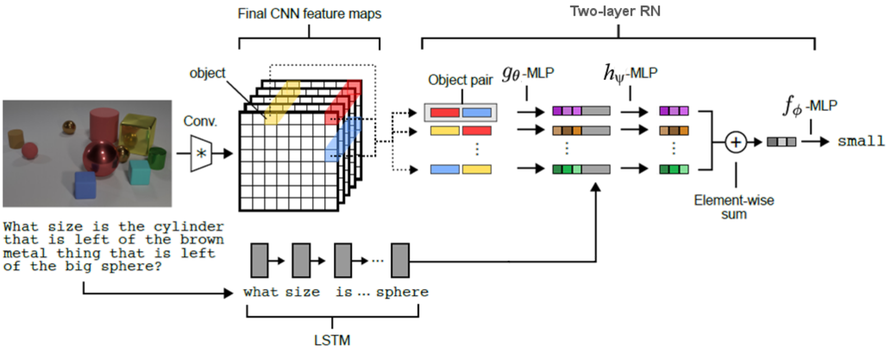


Fig. 3: The proposed two-stage Relation Network module and the whole architecture. This figure show the difference with respect to the original architecture shown in [13].

concepts without considering the questions, at least during the first stage, before the $h_\psi(\cdot)$ function evaluation. Hence, relation-aware features for the images can potentially be extracted from the output of any layer of the $g_\theta(\cdot)$ function.

The overall new architecture, named 2S-RN, is shown in Figure 3. For training, we stick to the procedure reported in [13]. Detailed configurations for 2S-RN on both CLEVR and Sort-of-CLEVR, together with the link to our implementation, are reported in Appendix A. With the proposed architecture, we obtain a representation $r_{i,j}$ for each pair of objects i, j . In Section 6, we report the results we obtained with *max* and *average* aggregations approaches, obtained by computing respectively $avg_{i,j}(r_{i,j})$ and $max_{i,j}(r_{i,j})$.

6 Experiments

We evaluate both convolutional and 2S-RN features against the generated ground-truth. In our experiments, features from 2S-RN are extracted from the last layer of $g_\theta(\cdot)$. We generate image rankings from relational features by calculating the Euclidean distance between the query feature and all the others, and then sorting the entire dataset by using this distance as score. Spearman-Rho is used to give a score to the obtained ranking, as explained in Section 4.4.

As a baseline for convolutional features from the original RN, we choose the ranking obtained with one of the state-of-the-art non-relational image descriptors for image instance retrieval, namely the RMAC descriptor [24]. This descriptor encodes and aggregates several regions of the image in a dense and compact global image representation exploiting a pre-trained fully convolutional network for feature map extraction. The aggregated descriptor is obtained by max-pooling the feature map over different regions and scales, and summing them together, followed by an $l2$-normalization. A similarity score between two images is obtained by computing the cosine similarity between their RMAC descriptors. In our experiments, we adopted the RMAC descriptor extracted from the trained model proposed in [25].

	Sort-of-CLEVR		CLEVR	
GT policy	soft-match	hard-match	soft-match	hard-match
RMAC [25]	0.49±0.03	0.07±0.03	−0.15±0.02	−0.18±0.02
RN [13] *max*	0.36±0.02	0.14±0.03	−0.24±0.02	−0.25±0.03
RN [13] *avg*	0.64±0.02	0.34±0.04	0.08±0.05	0.06±0.05
2S-RN *max*	**0.70**±0.02	**0.58**±0.03	−0.19±0.03	−0.21±0.03
2S-RN *avg*	0.24±0.02	0.18±0.02	**0.15**±0.04	**0.13**±0.04

Table 1: Spearman-Rho correlation index for existing features and our novel two-stage extracted features (2S-RN), both using CLEVR and Sort-of-CLEVR. We report the 95% confidence intervals for the mean over 500 queries.

We employ features extracted from the convolutional layer of the original RN as baseline for evaluating features from the first-stage of our novel two-stage approach. Table 1 reports values of Spearman-Rho for the two considered datasets [1]. They are relative to the two generated ground-truths, *soft-match* and *hard-match* obtained by ranking images using Approx-GED. Exact-GED could have been employed only for Sort-of-CLEVR, due to unacceptable computational times if applied on CLEVR graphs. Spearman-Rho correlation between rankings obtained with exact and approximated versions on Sort-of-CLEVR dataset over 500 queries give a value of 0.89, using the *soft-match* policy. Hence, we can empirically claim that this approximation is legitimate in this particular scenario. In light of this, we decided to use Approx-GED for both datasets in order to produce a fair comparison.

Correlation index has been evaluated over multiple rankings, generated using 500 query images, in order to produce statistically meaningful results. As it can be noticed, with a 95% confidence interval on the mean, convolutional relational features definitely defeat RMAC features on this relational task. Furthermore, relational features extracted from the two-stage RN are noticeably better than convolutional relational features. These results are reasonable since the original RN presents problems reasoning on the image alone, while RMAC tends to retrieve images containing the very same objects present in the query disregarding relative size, order or position.

Depending on the dataset, different aggregation methods can produce diverse optimal results. In particular, *max* aggregation seems working better on

[1] Code for reproducing CLEVR results is publicly available here:
`https://github.com/mesnico/learning-relationship-aware-visual-features`

Sort-of-CLEVR dataset, while *average* obtain the best results on CLEVR. The *average* aggregation keeps into consideration the number of identical relations happening inside the scene; and number of relations involved among objects having same attributes is quite important when considering CLEVR, since, unlike Sort-of-CLEVR, in CLEVR there is a better overall randomness and multiple instances of the same relationship could emerge (multiple relations insisting on similar objects). Hence, discriminating them by their cardinality becomes a must for an overall better ranking. Moreover, the *max* aggregation becomes unstable and sensible to outliers when the number of samples increases; a single huge activation in one of the 4,096 features in CLEVR can significantly affect the aggregation results. This is in line with findings in aggregation techniques for CNN features [24,25], where sum (and similarly avg) aggregation is preferred. Relations in Sort-of-CLEVR are significantly less and easily encoded in the feature space.

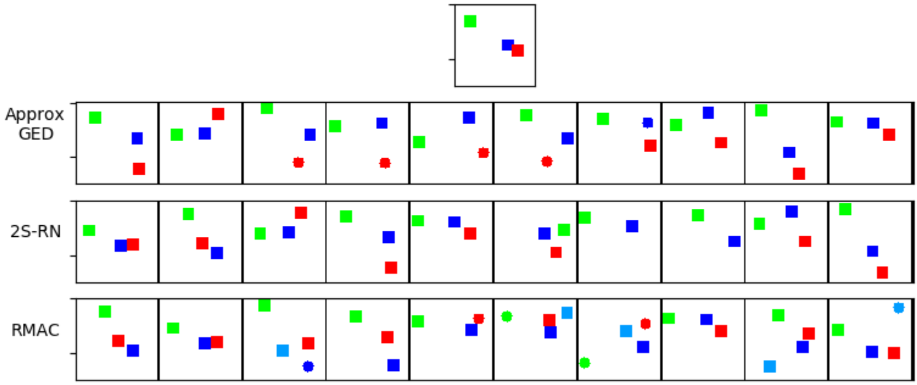


Fig. 4: Top 10 Sort-of-CLEVR images using our solution (2S-RN) and RMAC against our ground-truth for a given query (on top).

Even if it is quite difficult to give an objective evaluation of RN features and RMAC ones by only looking at the first 10 more relevant images, visual evaluation reported in Figures 4 and 5 are useful for giving an intuition beyond statistics (the query is not considered in the retrievable results).

7 Conclusions and Future Work

State-of-the-art methods for relational reasoning evaluate their capabilities on VQA tasks. In this work, we defined the sub-task of R-CBIR in which retrieved images should be similar to the query in terms of relationships among objects. This was motivated by the fact that current image retrieval systems, performing traditional CBIR, are not able to infer relations among the query and the retrieved images.

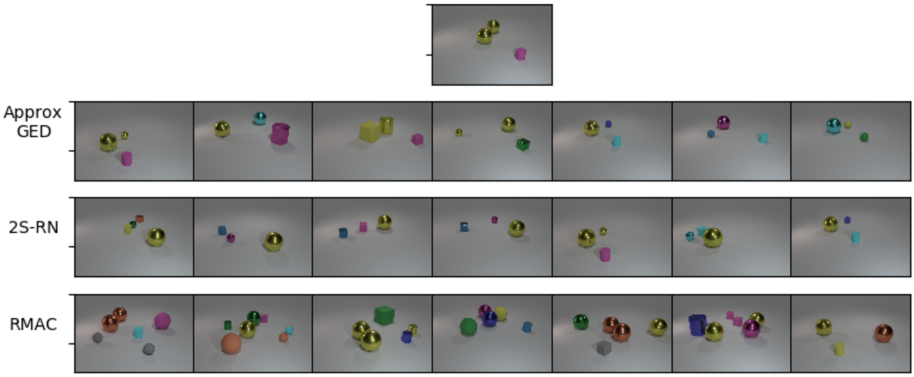


Fig. 5: Top 7 CLEVR images using our solution (2S-RN) and RMAC against our ground-truth for a given query (on top).

Given the novelty of the proposed task, we had to generate a benchmark. To this aim, we extended both CLEVR and Sort-of-CLEVR considering scene graphs of their images and generating a ground-truth for the R-CBIR task. We also proposed to employ the RN module, a state-of-the-art architecture for Relational VQA for extracting relational features suitable for the novel R-CBIR task. Experiments we conducted on this benchmark show that features extracted from the RN module are able to outperform state-of-the-art R-MAC features on this specific task.

We also proposed an extension to the RN module, called two-stage RN. This modification aims at decoupling visual relationships processing (*first-stage*) from the question elaboration (*second-stage*) so that layers in the first stage are unconditioned to the question and can be consequently used as candidate extraction points for obtaining good visual relation-aware features. We proved that features from our two-stage RN are able to encode relationships between objects in the image that neither traditional visual features nor features extracted from original RN formulation are able to detect.

Moving from these promising results to a scenario in which relationships between objects in real photos are encoded in features pose the same issues ongoing research on relational reasoning is facing on Relational VQA. To this aim, we will have to move from artificial images (CLEVR) to photos (e.g., VisualGenome). Results obtained on relational reasoning should also be combined with the approaches used for visual relationship detection. Moreover, the usage of the proposed features for cross-media retrieval and for finding all the images that contain specific relationships expressed in a textual query is also challenging.

Acknowledgments

This work was partially supported by Smart News, Social sensing for breaking news, co-founded by the Tuscany region under the FAR-FAS 2014 program, CUP CIPE D58C15000270008, and Automatic Data and documents Analysis to enhance human-based processes (ADA), CUP CIPE D55F17000290009.

We are very grateful to the DeepMind team (Santoro et al.), that kindly assisted us during the replication of their work on Relation Networks.

We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

1. Krawczyk, D.C., McClelland, M.M., Donovan, C.M.: A hierarchy for relational reasoning in the prefrontal cortex. Cortex (2011) 588–597
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2425–2433
3. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning (2017)
4. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. (2016)
5. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: European Conference on Computer Vision. (2016)
6. Peyre, J., Laptev, I., Schmid, C., Sivic, J.: Weakly-supervised learning of visual relations. In: ICCV 2017- International Conference on Computer Vision 2017, Venice, Italy (October 2017)
7. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3668–3678
8. Hu, R., Rohrbach, M., Darrell, T.: Segmentation from natural language expressions. Proceedings of the European Conference on Computer Vision (ECCV) (2016)
9. Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2017) 3298–3308
10. Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A.M., Elhoseiny, M.: Large-scale visual relationship understanding. CoRR **abs/1804.10660** (2018)
11. Zhou, B., Tian, Y., Sukhbaatar, S., Szlam, A., Fergus, R.: Simple baseline for visual question answering. CoRR **abs/1512.02167** (2015)
12. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.J.: Stacked attention networks for image question answering. CoRR **abs/1511.02274** (2015)
13. Santoro, A., Raposo, D., Barrett, D.G.T., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.P.: A simple neural network module for relational reasoning. CoRR **abs/1706.01427** (2017)
14. Raposo, D., Santoro, A., Barrett, D.G.T., Pascanu, R., Lillicrap, T.P., Battaglia, P.W.: Discovering objects and their relations from entangled scene representations. CoRR **abs/1702.05068** (2017)
15. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. CoRR **abs/1704.05526** (2017)
16. Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Li, F., Zitnick, C.L., Girshick, R.B.: Inferring and executing programs for visual reasoning. CoRR **abs/1705.03633** (2017)
17. Perez, E., de Vries, H., Strub, F., Dumoulin, V., Courville, A.C.: Learning visual reasoning without strong priors. CoRR **abs/1707.03017** (2017)
18. Perez, E., Strub, F., de Vries, H., Dumoulin, V., Courville, A.C.: Film: Visual reasoning with a general conditioning layer. CoRR **abs/1709.07871** (2017)

19. Belilovsky, E., Blaschko, M.B., Kiros, J.R., Urtasun, R., Zemel, R.: Joint embeddings of scene graphs and images. ICLR (2017)
20. Cai, H., Zheng, V.W., Chang, K.C.: A comprehensive survey of graph embedding: Problems, techniques and applications. CoRR **abs/1709.07604** (2017)
21. Abu-Aisheh, Z., Raveaux, R., Ramel, J.Y., Martineau, P.: An exact graph edit distance algorithm for solving pattern recognition problems. **1** (01 2015)
22. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing **27**(7) (2009) 950 – 959 7th IAPR-TC15 Workshop on Graph-based Representations (GbR 2007).
23. Melucci, M.: On rank correlation in information retrieval evaluation. SIGIR Forum **41**(1) (June 2007) 18–33
24. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879 (2015)
25. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. arXiv preprint arXiv:1610.07940 (2016)

A Detailed Configuration

Hyper-parameters for the 2S-RN architecture have been tuned starting from the configurations given in [13]. We first replicated experiments from [13], so that we were able to get a solid starting point. Training with these configurations we obtained an overall accuracy of 93.6% on CLEVR and 94.0% on Sort-of-CLEVR, quite enough to use the learned weights from the first stage as a feature. Code for training CLEVR architecture and extracting features is made publicly available at `https://github.com/mesnico/RelationNetworks-CLEVR`.

A.1 2S-RN on CLEVR

Hyper-parameters for 2S-RN architecture working on CLEVR are the following:

- CNN is composed of 4 convolutional layers each with 24 kernels, ReLU non-linearities and batch normalization;
- g_θ, h_ψ and f_ϕ are multilayer perceptrons each composed of 2 fully-connected layers, 256 neurons each and ReLU non-linearities;
- a final linear layer with 29 units produces logits for a softmax layer over the answers vocabulary;
- dropout with 50% dropping probability is inserted after the penultimate layer of f_ϕ;
- the gradient norm is clipped to 50;
- the learning rate follows an exponential step increasing policy, that doubles it every 20 epochs, from 5e-6 up to 5e-4.

A.2 2S-RN on Sort-of-CLEVR

Hyper-parameters for 2S-RN architecture working on Sort-of-CLEVR are the following:

- CNN is composed of 4 convolutional layers each with 24 kernels, ReLU non-linearities and batch normalization;
- g_θ is a multi-layer perceptron composed of 4 fully-connected layers, containing respectively 2048, 1024, 512, and 256 neurons with ReLU non-linearities;
- h_ψ is a single-layer perceptron with 256 neurons and ReLU non-linearities;
- f_ϕ is a multi-layer perceptron composed of 2 fully-connected layers, 256 neurons each and ReLU non-linearities;
- a final linear layer with 10 units produces logits for a softmax layer over the answers vocabulary;
- a dropout with 50% dropping probability is inserted after the penultimate layer of f_ϕ;
- the learning rate is set to 1e-4.