Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Can we automate diagrammatic reasoning?

Arif Ahmed Sekh[a,*], Debi Prosad Dogra[b], Samarjit Kar[c], Partha Pratim Roy[d], Dilip K. Prasad[a]

[a] UiT The Arctic University of Norway, Tromsø, Norway
[b] Indian Institute of Technology, Bhubaneswar, India
[c] National Institute of Technology, Durgapur, India
[d] Indian Institute of Technology, Roorkee, India

ARTICLE INFO

Article history:
Received 18 April 2019
Revised 26 April 2020
Accepted 29 April 2020
Available online 6 May 2020

Keywords:
Abstract reasoning
Raven,s Progressive Matrices (RPM)
Diagrammatic reasoning
Visual IQ test

ABSTRACT

Diagrammatic reasoning (DR) problems are well known. However, solving DR problems represented in 4 × 1 Raven's Progressive Matrix (RPM) form using computer vision and pattern recognition has not yet been tried. Emergence of deep learning techniques aided by advanced computing can be exploited to solve such DR problems. In this paper, we propose a new learning framework by combining LSTM and Convolutional LSTM to solve 4 × 1 DR problems. Initially, the elementary geometrical shapes in such problems are detected using a typical CNN-based detector. Next, relations of various shapes are analyzed and a high-level feature set is produced and processed in the LSTM framework. A new 4 × 1 DR dataset has been prepared and made available to the research community. We believe, it will be helpful in advancing this research further. We have compared our method with some of the existing frameworks that can be used for solving RPM-guided DR problems. We have recorded 18–20% increase in the average prediction accuracy as compared to the prior frameworks when applied to RPM-guided DR problems. We believe the CV research community will be interested to carry out similar research, particularly to investigate the feasibility of solving other types of known DR problems.

1. Introduction

Abstract reasoning or diagrammatic reasoning requires visual representations of objects or diagrams. It involves the understanding of concepts and ideas from images with the patterns that are used in visual IQ tests [1]. Solving such diagrammatic reasoning problems using artificial intelligence can help us to understand complex patterns of objects in images. Typically, a test in diagrammatic reasoning consists of a set of questions. The questions are usually of multiple choices. These questions generally consist of a series of pictures, each of which is different. The task is to choose another picture from a number of options to complete the series. For examples, Fig. 1 shows a typical diagrammatic reasoning problem, where the first row represents the question and the second row contains four options out of which only one is correct. The objective is to learn the rules that can be applied to a sequence and then use them to pick an appropriate answer. Solving a question requires analyzing a sequence of shapes or patterns known as the Raven's Progressive Matrices (RPM) [2]. This is also known as abstract or inductive reasoning test.

1.1. Related work

Reasoning is the ability to make sense of things by verifying facts and applying logic. We refer to machine learning-based methods for reasoning as artificial reasoning (AR). AR uses knowledge completion, value approximation, and goal-oriented reasoning to solve different forms of reasoning [3]. Knowledge completion uses knowledge graphs to express the facts and it extracts a common sense knowledge. It is used in various machine learning guided reasoning such as image captioning and question answering [4]. Value approximation is a method for extracting numeric facts. It is used in quantitative question answering from natural language texts and images [5]. Goal-oriented reasoning is a top-down approach that heuristically searches for a solution to achieve a goal. It is popular in robotics, intelligent agent, and case-based reasoning [6]. Data and knowledge-driven statistical methods [7], logic

[*] Corresponding author.
E-mail addresses: arif.ahmed.sekh@uit.no (A.A. Sekh), dpdogra@iitbbs.ac.in (D.P. Dogra), samarjit.kar@maths.nitdgp.ac.in (S. Kar), proy.fcs@iitr.ac.in (P.P. Roy), dilip.prasad@uit.no (D.K. Prasad).

https://doi.org/10.1016/j.patcog.2020.107412

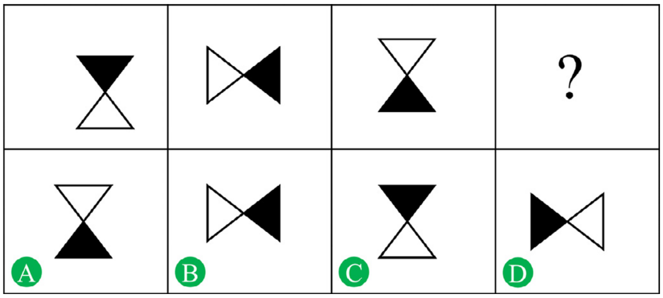


Fig. 1. A typical example of a diagrammatic reasoning problem. The first row presents the first three objects of a sequence of four objects in a particular order. The second row presents the multiple choices typically shown to an examinee. Option D is the right answer for the above problem.

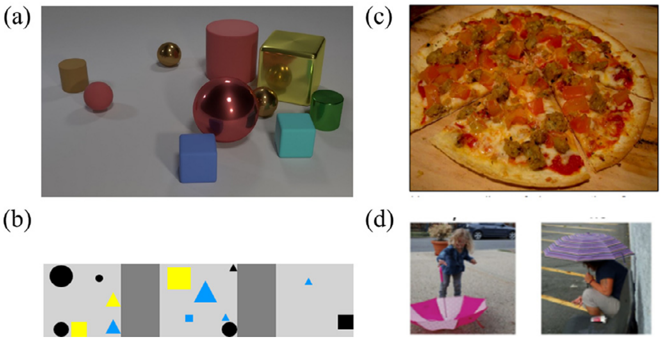


Fig. 2. Visual reasoning datasets. (a) *How many objects are either small cylinders or red things? (b) There is exactly one big yellow square not touching any edge (True/False) (c) How many slices are there in the pizza? (d) Is the umbrella upside down?.* (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

programming [8], and neural network-based approaches [9] are also popular for solving various reasoning problems.

AR methods are complex in nature and such methods require logical representation of data, common sense, statistical information, and learning techniques. Hence the learning guided reasoning methods need further improvement through the fusion of knowledge representation, reasoning and learning techniques [10]. For examples, statistical/relational learning [11] and knowledge base reasoning [12] are used in various reasoning problems. Statistical reasoning may be defined as making sense of historical data. It is widely used in psychology, health, economy, etc. In logical AI, first-order logic (FOL) and relational representations are used to gain knowledge. To know more about relational ML, the review work presented in [13] may be consulted. A thorough analysis of the literature reveals that the existing systems still suffer from a few drawbacks. For example, they often demand hand-crafted rules in the form of first-order logic, as such systems do not learn from examples [14]. Deep learning has been widely used to learn and represent the features. However, majority of the existing representations rely on low-level features and they do not consider high-level representations such as logic or knowledge. Recently, Serafini et al. [15] have proposed a logic tensor network to learn the data-driven logic. Similarly, Kazemi et al. [16] have proposed a deep neural network known as relational neural networks (RelNNs) to learn the reasoning directly from the FOL. Recently, Garcez et al. [17] proposed a neural-symbolic computing approach to combine neural networks with symbolic representation and reasoning-based learning approach. The method opened up new insights of intractability in AI. Mao et al. [18] used a similar concept to learn abstract knowledge from visual representation and language embedding. However, similar tasks in visual reasoning have not yet been tried by the CV community. Visual reasoning is not straight-forward as compared to the other types of reasoning due to the difficulty in interpreting the objects and their relations [19]. Therefore, existing logic and statistical AI methods cannot directly be applied to solve visual reasoning problems. Two similar domains of reasoning that have received the attention of CV research community are visual question answering [20] and visual reasoning [5]. Visual question answering consists of images and questions that can be answered from the images. To answer the questions, we may require prior knowledge about the objects, their colour, position, etc. In addition to these features, visual reasoning may also require shape information, count, orientation, etc. Fig. 2(a) depicts an example of visual reasoning taken from the popular Compositional Language and Elementary Visual Reasoning (CLVR) [5] dataset. CLVR dataset is used for reasoning colour, shape, quantity, and size. Fig. 2(b) depicts Cornell Natural Language Visual Reasoning (NLVR) synthetic dataset that is primarily used to categorize comments. The questions on NLVR demand understanding of natural language, shape, position, color, etc. Fig. 2(c) presents an example from the Visual Question Answering (VQA) dataset containing open-ended questions about the images [21]. These questions require an understanding of vision, language and common sense to answer. Fig. 2(d) presents reasoning of image pairs [22]. These questions require reasoning about the relation of objects.

Visual IQ questions that are based on RPM [2] vary in nature and diverse in complexities. Answers of RPM-based reasoning requires common sense, idea about the shapes, and knowledge of mathematics. There exist different types of DR problems. For examples, Fig. 3(a) represents a typical 2 × 2 DR problem and Fig. 3(b) represents a typical 3 × 3 DR problem. DR problems of the order 3 × 3 are considered as formal RPMs and their solu-

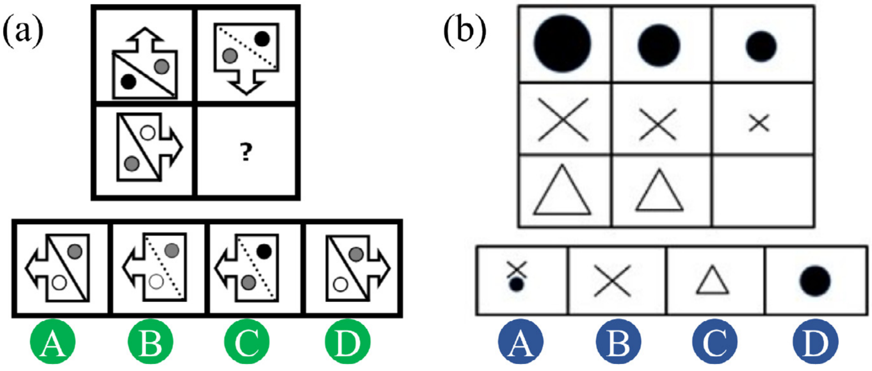


Fig. 3. Examples of various RPM-based DR problems. (a) Example of a 2 × 2 matrix reasoning problem. (b) Example of a 3 × 3 graphical reasoning problem.

Table 1
Summary of Raven's Progressive Matrices (RPM) Solving Methods.

Ref.	Problem Type	Feature	Method
Kunda et al. [25]	2 × 2	High-level(Symbol)	Model-based
Lovett et al. [26]	3 × 3	High-level (Object)	Model-based
Ragini et al. [27]	3 × 3	High-level (Structure)	Model-based
Mcgreggor et al. [28]	3 × 3	High-level (Object)	Model-based
Lovett et al. [29]	3 × 3	High-level (Structure)	Model-based
Santoro et al. [23]	3 × 3	Low-level (Raw Image)	Learning-based
Hill et al. [24]	3 × 3	Low-level (Raw Image)	Learning-based
Zhang et al. [30]	3 × 3	Low-level (Raw Image)	Learning-based
Proposed	4 × 1	High-level (Relation) + Low-level (Raw Image)	Learning-based

tions are addressed in [23] using neural network and in [24] using relational structure. Kunda et al. [25] have used two computational algorithms, namely Â Fractal Encoding Algorithm (FEA) and Affine-Extended Algorithm (AEA) to solve RPMs. FEA decomposes the images of a problem into a set of small images by applying a set of specific affine transformations (copy, rotation, or flip). Next, the algorithm generates fractal solutions to RPMs by considering all possible pairwise transforms. Lovett et al. [26] have used computational model to solve RPMs. The method uses structural information such as shape, texture, etc. It then uses Structure-Mapping Engine to find the pattern variance among images. Finally, a set of rules is applied to find a solution. Ragni et al. [27] have proposed a goal-oriented rule-based method to solve RPMs. The method first processes consecutive cells of the matrix to identify the goal (rule and texture) and then processes the solution image by analyzing the difference. McGreggor et al. [28] have proposed a confidence score for solving such problems. Firstly, each cell of the matrix is represented using relational fractal representations (feature similarity). Then, an image is expressed as a transformation (union, rotation, etc.) of a single image or multiple images. The answer is then chosen based on maximum similarity with the options. The work can be considered as a preliminary step toward structural representation of the features. Lovett et al. [29] have proposed computational model-based solution. Images are compared via structure mapping, aligning the common relational structure in 2 images to identify commonalities and differences. Barrett et al. [23] have released a dataset and proposed a neural network-based learning method to solve DR problems. Hill et al. [24] have extended the method using a neural network. They conclude that the state-of-the-art image-based deep neural networks fail to solve complex problems. However, if the rule is extracted correctly, the learning methods perform better. Zhang et al. [30] have proposed a model to generate different RPMs. They have shown that the state-of-the-art structural similarity-based, rule-based, and deep neural networks can achieve a maximum of 65% accuracy on the dataset. Different approaches for solving RPMs can be categorized by the choices of problem types, features, and the solution approaches. State-of-the-art approaches either use raw images as features or extract high-level information such as texture, shape, colour, etc. There are broadly two types of solution approaches, computational modeling approaches with rule-based system and learning-based approaches. A few methods that are similar to our work are summarized in Table 1.

Moreover, the existing approaches try to solve 2 × 2 and 3 × 3 RPMs using computational models or low-level image-based learning methods. In this paper, we have considered 4 × 1 diagrammatic problems and use high-level features for learning and reasoning. We have made the following research contributions:

- We have introduced a new feature representation that can be used by typical learning frameworks to solve diagrammatic reasoning problems.
- We propose a problem classifier and solver to address the solution extraction of typical 4 × 1 DR problems. The method can learn a concept with a knowledge-base using less number of training samples.
- We have introduced a new dataset containing 4 × 1 DR problems represented in the form of RPMs. The dataset contains state-of-the-art RPMs that can be generated using rules as well as complex problems. The dataset has been made available to the research community for further investigation.

Table 2
Visual Reasoning Dataset.

Dataset	Pattern
Abstract Reasoning [23,30]	3 × 3
Scanned Images [28]	3 × 3
Digital Images [31]	3 × 3
Proposed	4 × 1

1.2. Datasets and benchmarks

The ultimate goal of visual reasoning is to learn image understanding and interpretations. Lack of datasets makes it difficult at this stage to apply computer vision techniques to solve DR problems. Despite the advancement of deep learning frameworks, training the networks with a sufficient amount of data is a challenge. To the best of our knowledge, only a 3 × 3 DR dataset has recently been released by Barrett et al. [23]. The dataset is referred to as Procedurally Generated Matrices (PGM) dataset. McGreggor et al. [28] have used a small dataset collected from scanned images. Table 2 presents summary of various datasets. Therefore, at this juncture, we thought to prepare and release a DR dataset to the research community so that the area can advance further. Thus, we have collected images of diagrammatic reasoning from the web and prepared a dataset of 4 × 1 diagrammatic reasoning problems. The dataset contains 619 problems. We have categorized these problems into four groups, namely (i) Rotation (RT), (ii) Counting (CT), (iii) Shape Scaling (SS), and (iv) Other Type (OT). Fig. 4 depicts a sample question with possible answers from each category and Fig. 5 depicts the distribution of the problems across the various categories in our dataset.

Rest of the paper is organized as follows. In Section 2, we present the proposed DR solving method. Experiment results are presented in Section 3. Conclusion and future work are presented in Section 4.

2. Proposed architecture

The proposed method is based on a set of features and an algorithmic pipeline. Majority of the reasoning problems are tackled with relational learning [11] and reasoning capabilities [12], whereas the image-centric neural network-based learning applications do not require relational learning. We have introduced a new

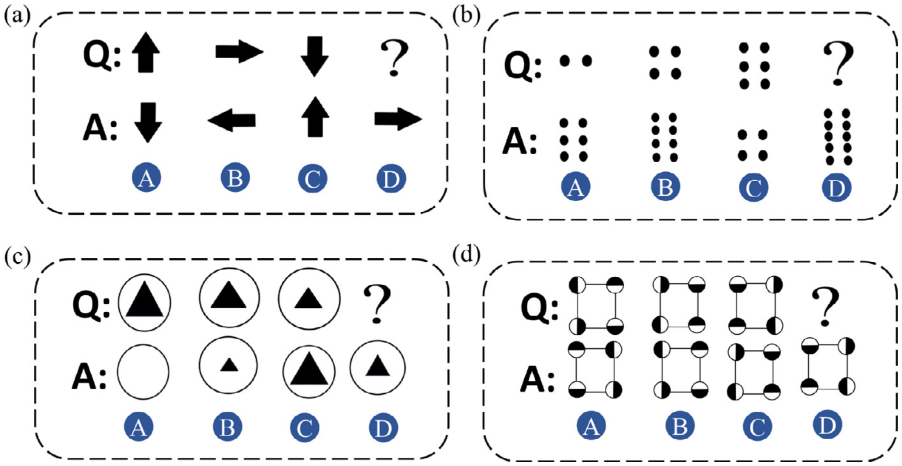


Fig. 4. Examples of four types of typical DR problems that are present in our dataset. (a) Example of a rotation problem (RT), where a pattern is rotated as compared to the first image with relative rotations that may be mentioned in a DR question as {0°, 90°, 180°, ?}. The prediction should be 270° and the correct answer is option B. (b) It is a typical problem of number series prediction (CT). The question consists of a set of filled circles. Here, the number of circles varies as 2, 4, 6, ?. Our task is to predict the picture with 8 filled-circles. The correct answer is option B. (c) Third one is an example of typical shape and scaling problem (SS). The pattern can be interpreted as { < Cicle, Large Triangle > , < Circle, Big Triangle > , < Circle, Small Triangle > , < ? > }. Our task is to predict < Circle, Tiny Triangle > which is option B. (d) The fourth one is a typical pattern understanding problem. We have categorized such problems into Other Type (OT). Our task is to predict the 4th pattern. The correct answer is option A.

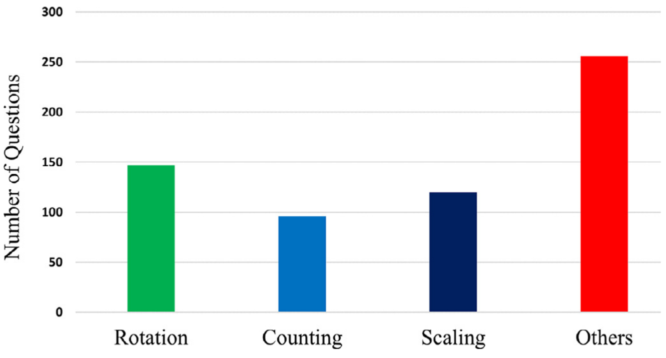


Fig. 5. Distribution of different DR problems in our dataset.

method to extract the relational features (RF) of image sequences to solve one specific type of DR problems. The proposed method consists of two major steps. During the first level of processing, the question and options are passed through a knowledge acquisition tool to construct the knowledge base. The knowledge consists of a set of image features extracted from the individual image and a set of relational features extracted from the sequence of images in the question and the options. Next, the problem type is identified using a supervised learning method. We define two types of Long Short Term Memory (LSTM) networks. Each one of them is responsible to solve one specific type of problems. One LSTM takes text-based features (for RT, CT, and SS type) and the other one takes image-based features (for OT types). Unlike the test-based reasoning problems [11], where the reasoning needs to be defined by knowledge or FOL [32], we learn the logic using LSTM through training. Fig. 6 depicts the proposed framework in detail. The pipeline consists of (i) a Knowledge extraction module, (ii) a problem classifier and LSTM chooser module, (iii) two LSTMs, and (iv) a matching module. Let the problem space (P) be defined in (1), where the question contains a set of images $(Q) = \{I_1, I_2, I_3\}$ and the options are grouped in the solution space $(O) = \{I_4, I_5, I_6, I_7\}$. Diagrammatic reasoning is to predict the answer such that $I_{answer} \in O$. First, we represent the problem using a high-level knowledge structure. The individual modules are described hereafter.

$$P = \{I_1, I_2, I_3, \dots, I_7\} \tag{1}$$

2.1. RCNN Module

First, each image of P is passed through an RCNN module to extract the shapes and the bounding box information. We have

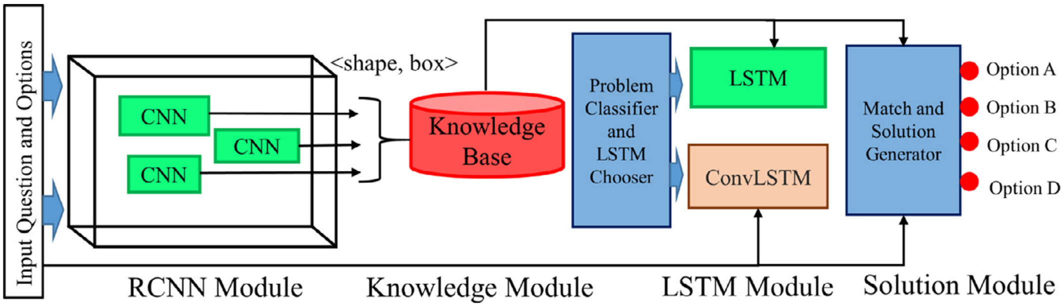


Fig. 6. Architecture of the proposed framework. The architecture consists of an RCNN module, a knowledge module, problem classifier module, two LSTM modules, and a solution module. We take raw question sequence and the options as input and construct a knowledge base by taking the output from the RCNN module. Next, the knowledge is used to classify the problems into 4 categories and select the suitable LSTM. Finally, it predicts the best possible option out of the four input options and produces a complete sequence of four patterns/images.

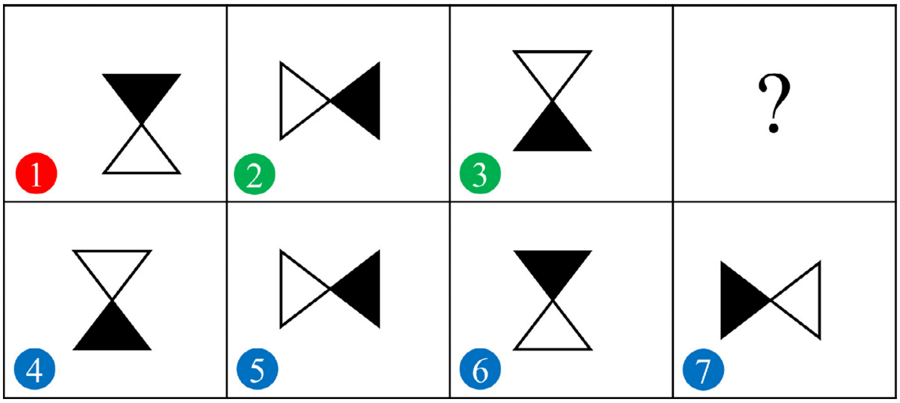


Fig. 7. We represent the rotation problem as a set of 7 images or patterns. In rotation problems, we consider the first image (red) as the reference image with 0° rotation and extract the rotation relation of other images. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

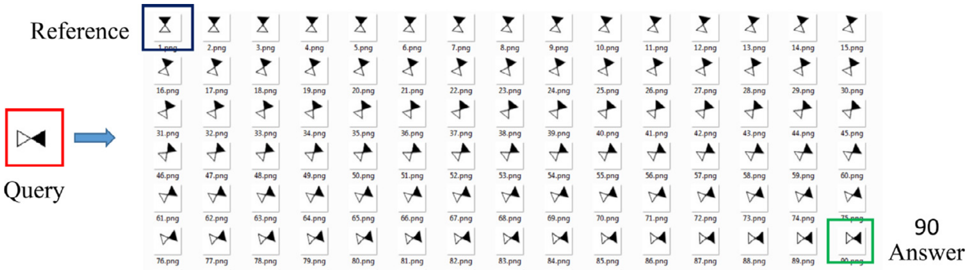

Reference
Query
1.png 2.png 3.png 4.png 5.png 6.png 7.png 8.png 9.png 10.png 11.png 12.png 13.png 14.png 15.png
16.png 17.png 18.png 19.png 20.png 21.png 22.png 23.png 24.png 25.png 26.png 27.png 28.png 29.png 30.png
31.png 32.png 33.png 34.png 35.png 36.png 37.png 38.png 39.png 40.png 41.png 42.png 43.png 44.png 45.png
46.png 47.png 48.png 49.png 50.png 51.png 52.png 53.png 54.png 55.png 56.png 57.png 58.png 59.png 60.png
61.png 62.png 63.png 64.png 65.png 66.png 67.png 68.png 69.png 70.png 71.png 72.png 73.png 74.png 75.png
76.png 77.png 78.png 79.png 80.png 81.png 82.png 83.png 84.png 85.png 86.png 87.png 88.png 89.png 90.png
90
Answer

Fig. 8. Sample images after applying rotations on the first image of *P*. Depiction of how a possible match is found at 90° for a given query image.

considered 7 common geometrical shapes, namely circle, triangle, rectangle, square, diamond, star, hexagon that are usually present in various DR problems. All the shapes are classified as either empty (only edges) or filled. We have experimented with the state-of-the-art RCNN to detect these shapes. YOLO [33] has been found to be a good recurrent classifier as compared to Resnet50/101 [34], VGG 16 [35], or GoogleNet [36].

Architecture and Training: YOLO predicts the bounding boxes and class confidence of a given image. It consists of 53 no. of successive 3 × 3 and 1 × 1 convolutional layers. We have used a transfer learning approach [37] to train the RCNN. We have used ImageNet as the base model. For training, we have generated 14 types of images (class) consisting of 7 types of shapes with filled and unfilled objects. The synthetic dataset consists of 100,000 shape images similar to the FourShape[1]. It is generated by the varying size and applying rotation. The model has been trained with 3000 epochs with the default parameters of YOLO V3.

2.2. Knowledge acquisition module

Knowledge acquisition has been carried out during training and solution generation. The knowledge base (κ) is extracted from the sequence of images in the given problem and the set of options. We have considered the types of shape, number of shapes, and size of the shapes as the relative features. First, the shapes in each image in *P* and the bounding boxes are extracted using the RCNN. We then introduce a new feature extraction method for solving 4 × 1 DR problems. The feature is referred to as the relational feature (RF). Unlike image-based features such as color, texture, shape or edge that are typically used in various computer vision applications, we have extracted three relational features (RF), namely rotation (ρ), counts (χ), and scaling (σ) from the set of the given images. The feature-set is given in (2). Various components of the feature-set (k) are described hereafter.

$$\kappa =< \rho(I_k), \chi(I_k), \sigma(I_k) >, \forall k, k \in P \tag{2}$$

Rotation: In a typical rotation diagrammatic reasoning problem (Fig. 7), the solution lies in rotating the figure correctly to complete the sequence. We assume the first image (I_1) as the reference with a rotation of 0°. All the other images ($I_2, \ldots, I_7$) are expressed using rotation angle with respect to the reference image. To achieve this, 360 images are generated by incrementally rotating the base image by $r°$, where $r = 1$. A few samples of the rotated images corresponding to the DR problem described in Fig. 7 are shown in Fig. 8. This set is denoted by $R = \{I_1, I_2, \ldots, I_{360}\}$. The similarity score ($\psi$) is defined in (3). First, a ResNet50 [34] network with average pooling has been used to extract features of images. The network uses pretrained imagenet as the weight vector. The score has been estimated between a query image and all images of *R* using the ResNet50 by considering chi-square distance, where I_j is query image and I_k is the image in *R*.

$$\psi_{jk} = Similarity(I_j, I_k) \tag{3}$$

The relative rotation $\rho(I_k)$ of each image of *P* is then extracted with respect to each image I_j belonging to *R*. If the images in *P* are different from each others, we categorize the question as a non-rotation problem and the not applicable (NA) flag is set. A threshold has been used to decide about the success of matching. $\rho(I_k)$ is set to the value of rotation if the matching score returned by the ResNet50 is above the threshold. However, in the event of multiple images being categorized above the threshold, the image that gives the highest value is selected and its rotation angle is taken as the final input. In the event that none is found suitable, the problem is categorized as a non-rotation diagramatic reasoning problem. For example, the relative rotations of the diagrammatic reasoning problem depicted in Fig. 4(a) are {0°, 90°, 180°} for the op-

[1] https://www.kaggle.com/smeschke/four-shapes.

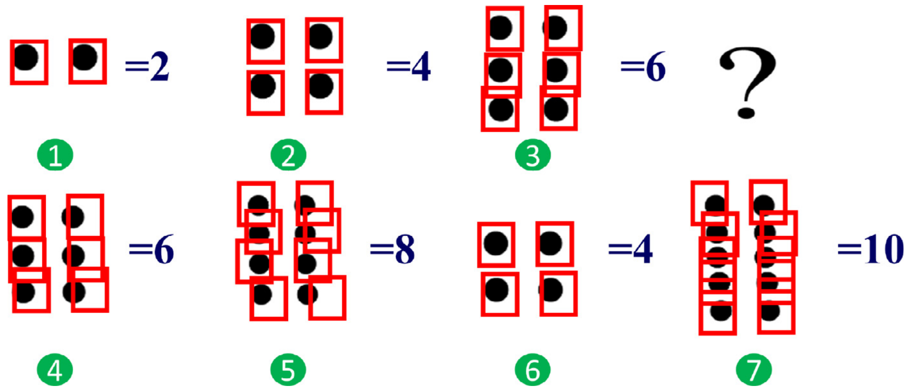


Fig. 9. A typical counting DR problem with 7 pictures. The first three patterns represent the sequence given in the question 2, 4, 6, ? and the next four patterns represent the options for the probable answer with 8 as the correct option.

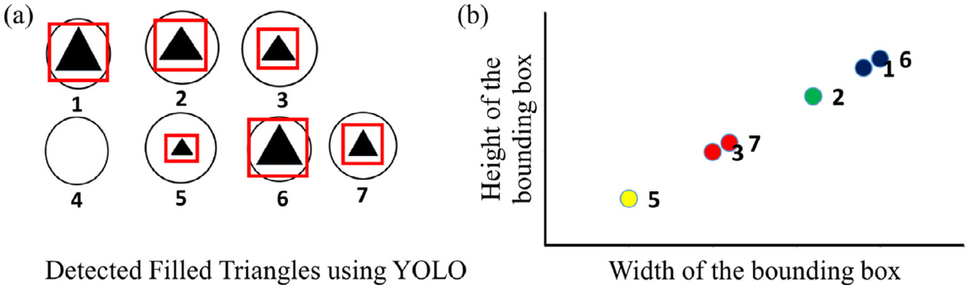


Fig. 10. (a) Detection of shapes achieved by YOLO. (b) The bounding boxes are grouped using DBSCAN. Each color represents a group of same size shapes.

tions in the question and {180°, 270°, 0°, 90°} for the options in the answer.

Counting: Counting is a reasoning problem where the solution is to extract the correct number of shapes present in the problem sequence. First, the shapes are detected and the number of the same types of shapes is estimated. For example, Fig. 9 depicts a typically filled circle detection and counting using RCNN. Each image of the problem space is expressed using the count of shapes in a sequence as {2, 4, 6, ?}. The predicted missing number is needed to select from the set {6, 8, 4, 10}.

Scaling: Relative scaling (σ) is then extracted from the bounding box of the detected shapes. First, the bounding boxes are extracted from the shapes in P. Each shape in the question image sequence and the options are represented by width (w) and height (h) of the bounding box. Next, each type of shapes are grouped using unsupervised density-based spectral clustering with application to noise (DBSCAN) [38] using w and h. The groups are then rearranged in increasing order of the area ($w \times h$) such that $area(L_1) < area(L_2) \ldots < area(L_n)$. These groups are labeled using rules as extremely large, very large, large, normal, medium, small and tiny based on the number of clusters. The grouping and labeling of shapes are described in Algorithm 1.

Fig. 10 (a) depicts a DR problem where size of the pattern is used as a clue for the solution. The DBSCAN algorithm can identify four classes or groups, where the problem has been expressed as $\{<VeryLarge>, <Large>, <Small>, ?\}$, and the solution options are $\{<Nil>, <Tiny>, <VeryLarge>, <Small>\}$.

Representation of knowledge base: For a given problem space P, the shapes are detected and the relational features (RF) are extracted as mentioned earlier. The knowledge base consists of four sets, namely shapes, rotation (ρ), counting (χ), and scaling (σ). Shapes store information about the structures and other sets represent various components of the relational features. Table 3 shows the knowledge extracted from four different 4 × 1 problems.

Algorithm 1 Scaling-based feature extraction.

```
Input: Problem Space (P) as defined in equation~(1)
Output: Relational scaling ($\sigma$) of each image
S=DetectShapes($I_k$), $\forall k, I_k \in P$
ShapeGroup=DBSCAN(S)
Extract number of cluster (c) from ShapeGroup
Rearrange group and assign label $L_1, L_2, \ldots, L_n$, where $area(L_1) < area(L_2) \ldots < area(L_n)$
if $c = 6$ then
    $L = \{Extremely\ Large,\ Very\ Large,\ Large,\ Medium,\ Small,\ Tiny\}$
else if $c = 5$ then
    $L = \{Very\ Large,\ Large,\ Medium,\ Small,\ Tiny\}$
else if $c = 4$ then
    $L = \{Large,\ Medium,\ Small,\ Tiny\}$
else if $c = 3$ then
    $L = \{Large,\ Medium,\ Small\}$
else if $c = 2$ then
    $L = \{Large,\ Small\}$
else if $c = 1$ then
    $L = \{Normal\}$
else
    $L = \{Nil\}$
end if
$\sigma_k$= Shape Label($I_k$)
Return $\sigma$
```

2.3. Problem classification module

Problem classification plays an important role as it is used to select the appropriate LSTM module. Failure in classification may lead to a wrong solution selection. The knowledge base of the relative features extracted in the previous step is used to classify the problem and based on the problem category a specific feature is chosen to represent the problem. We call the fea-

Table 3
Typical examples of knowledge base extracted using the features described earlier (The first 3 rows are correctly extracted, the last row is failure case).

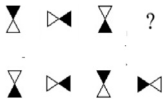
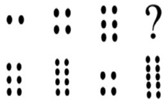
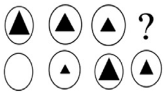
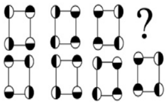
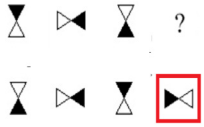
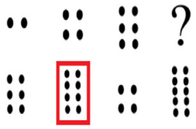
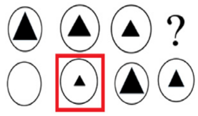
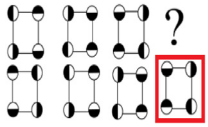
DR Problem	Constructed knowledge base
	$Shapes = \{Filledtriangle, Triangle\}$ $\rho = \{0^\circ, 90^\circ, 180^\circ, 180^\circ, 90^\circ, 0^\circ, 270^\circ\}$ $\chi = \{< 1, 1 > < 1, 1 > < 1, 1 >, < 1, 1 >, < 1, 1 >, < 1, 1 >, < 1, 1 >\}$ $\sigma = \{< N, N >, < N, N >, < N, N >, < N, N >,$ $< N, N >, < N, N >, < N, N >\}$, *where N is Normal.*
	$Shapes = \{Filled\ circle\}$ $\rho = \{NA\}$ $\chi = \{< 2 >, < 4 >, < 6 >, < 6 >, < 8 >, < 4 >, < 10 >\}$ $\sigma = \{< AN >, < AN >, < AN >, < AN >, < AN >,$ $< AN >, < AN >\}$, *where AN is All Normal.*
	$Shapes = \{Filled\ triangle\}$ $\rho = \{NA\}$ $\chi = \{< 1 >, < 1 >, < 1 >, < 1 >, < 1 >, < 1 >, < 1 >\}$ $\sigma = \{< VeryLarge >, < Large >, < Small >,$ $< Nil >, < Tiny >, < VeryLarge >, < Small > \}$
	$Shapes = \{Filled\ triangle\}$ $\rho = \{NA\}$ $\chi = \{< 4 >, < 4 >, < 4 >, < 4 >, < 4 >, < 4 >, < 4 >\}$ $\sigma = \{< AN >, < AN >, < AN >, < AN >, < AN >,$ $< AN >, < AN >\}$, *where AN is All Normal.*

Table 4
Details of the predicted knowledge and answers.

Predicted answer	Predicted knowledge	Detected category	Correct?
	$\rho = \{270^\circ\}$	Category 1 (RT)	Yes
	$\chi = \{< 8 >\}$	Category 1 (CT)	Yes
	$\sigma = \{< Tiny >\}$	Category 1 (SS)	Yes
	Not Applicable	Category 2 (OT)	No

ture as active feature (α). First, the three images of the problem are chosen and κ is extracted for those images. Next, the rotation (ρ) and counting (χ) is replaced by ”Equal/Not Equal” if all the values are equal or not. Next, all the features are encoded using the one-hot encoder and a supervised k-nearest neighbor (KNN) is applied to classify the problem into 4 classes (CT, RT, SS, and OT). We have empirically chosen $k = 10$ and it produces good results. Based on the problem type, the active feature is chosen as given below:

$$\alpha = \begin{cases} \rho & \text{if } class = RT \\ \chi & \text{if } class = CT \\ \sigma & \text{if } class = SS \\ I & \text{if } class = OT \end{cases}$$

Table 4 shows reference feature prediction (answer) of the problems shown in Table 3.

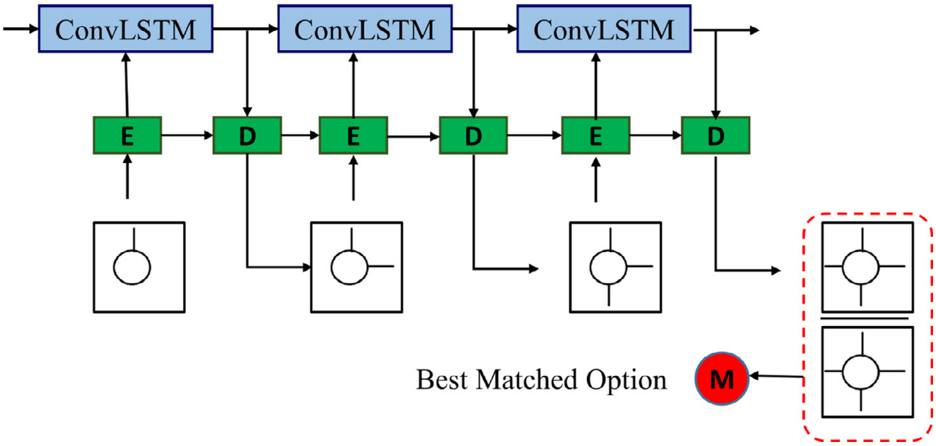


Fig. 11. Interpolation model for solving Category 2 problems. E: Encoder, D: Decoder.

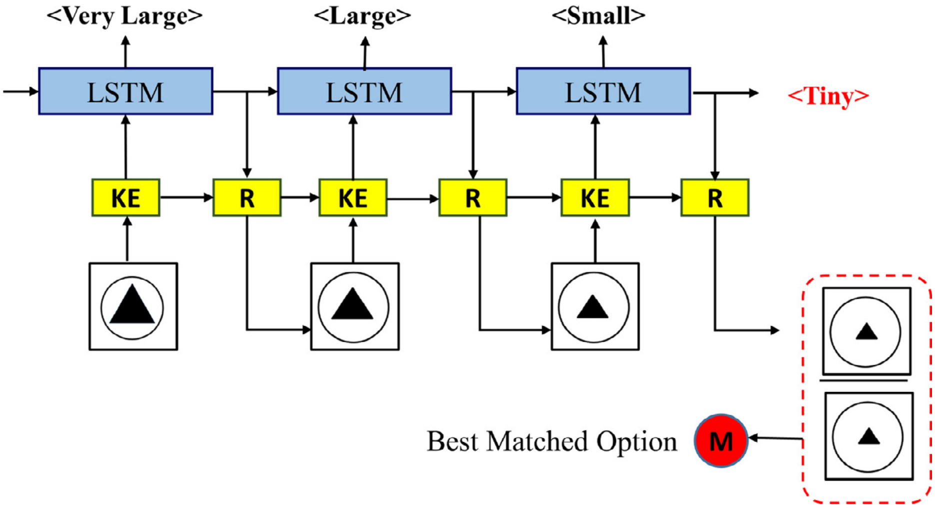


Fig. 12. Prediction model for solving Category 1 problems. Knowledge extractor is used to represent sequence of images to sequence of features (ρ, χ, σ). Depending on the problem type corresponding feature is taken as the input (active feature) to the LSTM.

2.4. LSTM-based solution generator

The final stage is to learn the pattern from the question images and predict the correct answer from the given options. We have used two variations of LSTM to predict the answer option. In the case of Category 1, a simpler variation of LSTM is used as proposed in [39]. The method has been used to generate a caption from the images. We have not used the image as input. We have used the neural architecture of the language learning and generation part in the proposed method. Rather than using the conventional image-based features [40], we have used relational features (RF) extracted by the knowledge extractor. The method is depicted in Fig. 12, where the knowledge extractor (KE) is the process of extracting RFs as discussed earlier. At the beginning, the relational features (RF) are extracted from all training samples and the active feature (α) is chosen. Next, a text-based LSTM corresponding to Category 1 (α is the input to the LSTM) is trained to build the prediction model. In the testing phase, a similar knowledge base and the active feature are extracted from the test samples. Unknown problems (Category 2) are solved by a variation of the LSTM, called Flexible Spatio-Temporal Network (FSTN) proposed in [41]. Originally the method predicts the future video frames from a set of observed sequences. In this method, image-based features are sequentially passed through a convolutional-LSTM. Fig. 11 depicts the method in detail. The method consists of a sequence of convolutional and pooling layers and LSTM modules. The method takes a sequence of images and features after convolution and pooling are passed through the LSTM. The network is trained using the image sequences consisting of the problem and correct answer images.

Model architecture and training : Category 1 LSTM is modelled by an RNN considering $p(S_t|K, S_0, S_1, \ldots, S_{t-1})$, where K is the knowledge, S is the word representing knowledge words, and t is the time step ($t = 4$). The hidden state or memory (h_t) is updated after receiving the input (x_t) by nonlinear function f:

$$h_{t+1} = f(h_t, x_t) \tag{4}$$

To make the above network applicable to our domain, two crucial design choices are to be made: (1) What is the exact form of f? and (2) How are the images and words fed as inputs x_t? For f, we use a Long-Short Term Memory (LSTM) network, that has shown state-of-the-art performance on sequence classification. The LSTM uses state-of-the-art modules [42] for hidden layers and the final prediction layer defined in (6), where U and V are input and output weight vectors and b is the bias.

$$h_t = \sigma(b_i + x_t U_i + h_{t-1} V_i) \tag{5}$$

The hidden unit is combined with forget gates and output gates and the final layer is a softmax layer as given in (6).

$$p_{t+1} = Softmax(last\ layer) \tag{6}$$

In our case, we have used a multi layer LSTM with 512 units per layer consisting of 2 LSTM layers. Moreover, we denote the input problem by P and the sequence of active features for the image by $S = (S_0, S_1, S_2, S_3)$. The approach uses the steps as given in 7–(9),

$$x_{-1} = Active\ Feature(\alpha) \tag{7}$$

$$x_t = W_e S_t, t \in \{0, ., N-1\} \tag{8}$$

$$p_{t+1} = LSTM(x_t), t \in \{0, ., N-1\} \tag{9}$$

where each knowledge descriptor word (S_t) uses one-hot word embedding (W_e). We have used sequence loss function minimization in each step as given in (10).

$$Loss(K, S) = \sum_{t=1}^{N} \log p_t(S_t) \tag{10}$$

The model has been trained using supervised active features extracted from the training sets. The problems have been solved by the volunteers to obtain the ground truths and the active features are recorded and used for training. For example, the training sequence used for the first three problems described in Table 4 are {0°, 90°, 180°, 270°}, { < 2 > , < 4 > , < 6 > , < 8 > }, and { < *VeryLarge* > , < *Large* > , < *Small* > , < *Tiny* > }. The proposed LSTM is then trained to learn and predict from the active features of the training samples.

Category 2 RNN is a Convolutional LSTM that consists of a spatio-temporal autoencoder, which in turn consists of an image-based encoder-decoder with an LSTM cell acting as a temporal encoder. The encoder (E) contains one convolutional layer, leaky ReLU non-linearity, and a spatial max-pooling layer. The decoder (D) mirrors the encoder, except for the non-linearity layer, and uses spatial upsampling to bring the output back to the size of the original input. The proposed method uses 64 × 64 input image sequences with 5 × 5 kernel and 3 × 3 pooling layer with batch normalization. The LSTM modules are multilayered (we have used 2 layers) time distributed layer. We have used two types of losses as reported in [43] and [41]. The first loss is a l^2 loss applied on decoder output as $\mathcal{L}_t^D = \|\hat{X}_{t+1} - X_{t+1}\|_2^2$, where X is input image and $\hat{X}$ is predicted image. The second loss is an encoder loss applied on encoder output as $\mathcal{L}_t^E = \|E(\hat{X})_{t+1} - E(X_{t+1})\|^2$, where $E(\hat{X})$ is encoder feature output and $E(X)$ input feature to the encoder. The global loss is defined in (11).

$$\mathcal{L} = \sum_i \mathcal{L}^E + \mathcal{L}^D \tag{11}$$

The network has been trained using the image sequences of other types of problems (OT) with the supervised correct answer image. The method has been trained using a learning rate of 0.1 in 3000 epochs.

2.5. Solution module

The solution module consists of an active feature matching module and an image similarity module. In the case of Category 1 problems, the predicted active feature is matched with the active feature of the available options. For example in the first problem presented in Table 4, if $\alpha = \rho$ and the predicted solution is 270°, the match module searches for the option when $\rho = 270°$, i.e. option 4 is the correct answer. In the case of Category 2 problems, the predicted image is compared with all the option images using ResNet50 feature extractor. The solution is chosen based on the maximum matched image options.

Table 5
Results of shape detection.

Algorithm	Accuracy
ResNet50 (baseline)	57.19
ResNet101 [34]	62.19
VGG16 [35]	71.11
GoogleNet [36]	77.22
YOLO [33]	**86.76**

3. Experiment results

We present the experiment results in this section. The proposed architecture starts with a shape detection method followed by problem classification and solution selection.

3.1. Shape detection results

The first step of the method is to detect shapes from a given image. We have experimented with state-of-the-art convolutional networks including ResNet50, ResNet101 [34], VGG16 [35], GoogleNet [36] and YOLO [33]. YOLO has been found to be the best architecture for the present case. 70% of the data have been used for training and 30% for testing across all experiments. Results using 10-fold cross validation have been reported. Table 5 summarizes the shape detection results.

3.2. Problem classification

In the next stage, an analysis of the results of classification has been carried out. The confusion matrix for four types of problems is depicted in Fig. 13. It may be observed that the KNN with the proposed feature can successfully classify the problems with reasonably high accuracy. We have performed a 10-fold cross-validation and observed that the proposed classifier classifies counting and rotation problems with 92% and 87% accuracy respectively. The accuracy of scaling and other types of problems has been found to be 88%. This decline in accuracy is due to the complex nature and diverse variety of the problems in the other group. In our proposed method, the identification of scaling problems involves scaling factor identification and clustering. A failure in any step may affect the classification outcome. Moreover, majority of the complex other type of problems can be classified with 81% accuracy.

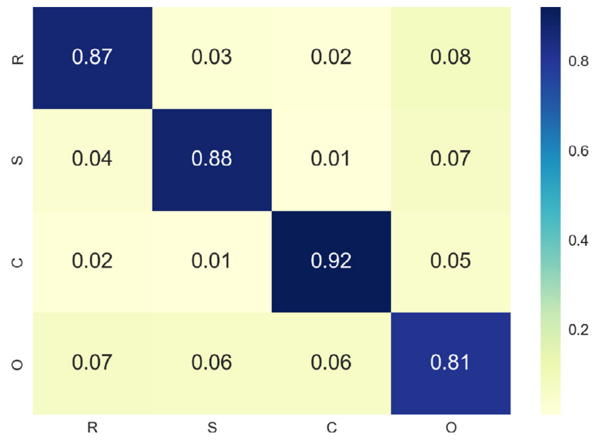


Fig. 13. The confusion matrix for classifying DR problem. R: Rotation, C: Counting, S: Scaling, O: Other.

Table 6
Comparative results of DR problem solving.

Algorithm	Rotation (RT)	Counting (CT)	Scaling (SS)	Other (OT)	Average
Image+LSTM (baseline)	57.12	42.13	62.11	36.5	49.46
Image+Encoder/Decoder [40]	62.11	41.12	61.11	37.89	50.55
Image+Deep feature [44]	64.39	47.19	41.91	42.86	49.08
Image+RNN [45]	56.80	41.19	54.91	32.20	46.27
Image+FSTN [41]	66.11	37.19	66.91	34.90	51.27
Proposed RF+LSTM	**76.21**	**77.00**	**74.31**	**66.81**	**73.58**

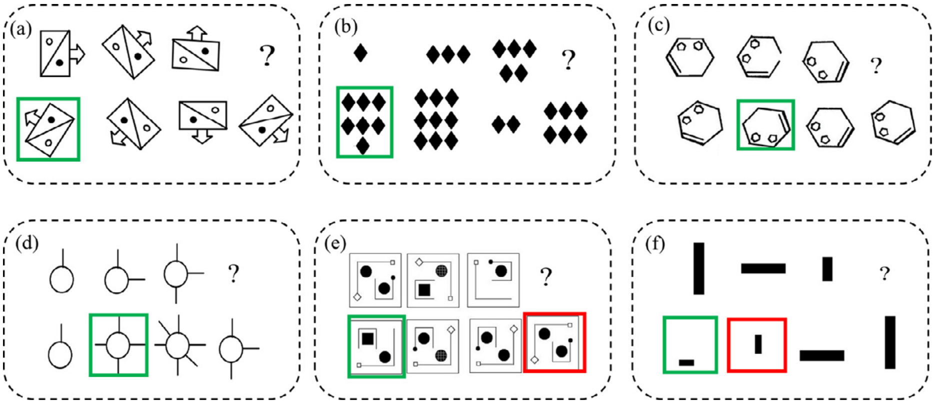


Fig. 14. A few samples from the our DR dataset when the proposed method correctly identifies the answers. The green boxes represent the ground truths and/or the correctly chosen answers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

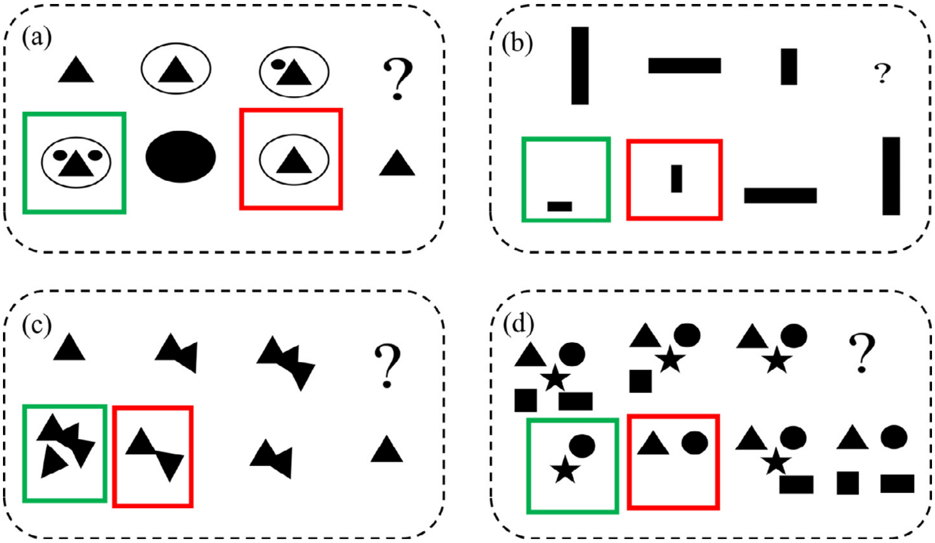


Fig. 15. A few samples taken from our DR dataset where the proposed method fails to choose the correct option. The green boxes represent the ground truths and the red boxes represent wrongly predicted answers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.3. Solution selection

In the final stage, LSTM is used to find the correct solution. We have compared the proposed architecture with the state-of-the-art image-based reasoning solvers. The results are summarized in Table 6. Fig. 14 depicts some success cases, where (a) and (d) represent rotation problems, (b) and (f) represent counting problems, (e) represents a typical scaling problem and (c) represents other type problems. It may be observed that the proposed method can solve different types of DR problems with better accuracy when compared with existing techniques. Fig. 15 presents some failure cases. It has been observed that the FSTN is applicable when the sequence of the image contains continuous visual changes such as human motion in video or completeness problem as shown in Fig. 14(c). It is not suitable for reasoning problems that contain high-level logic information. Reasoning for high-level concepts demands knowledge of shape, counting, relation, etc. It may be observed that the proposed method has failed to learn complex patterns of reasoning problems.

There are problems which are more complex than counting, scaling, and rotation, such as involving XOR and AND operations or pattern-based problems involving line or figures [28]. Such problems may be solved using rules [23]. These types of problems follow simple patterns and neural network can easily learn the logic and apply it to unknown problems. The others type of problems may be related to completeness (Fig. 15(a) and (c)), where each figure is incrementally completed or reversed by adding or subtracting different parts or may be mixed problems (Fig. 15(d)) that are usually combinations of various complex concepts. Solving such problems demands a higher degree of cognitive skills that humans possess. To solve such problems, neural networks not only require training on how to deal with the problems, but also need to apply the knowledge of numbers, operators, logic, visual patterns, and mathematical rules to obtain results.

3.4. Comparative analysis

Computational model-based approaches such as structure-mapping [26], cognitive modeling [27], reasoning-based [28], and modelling approach [29] are based on a set of fundamental rules for solving RPMs. The methods use high-level features and are restricted to specific types of reasoning problems that can be solved by a set of well-defined rules. These methods aim to model the rules to automate the solving process. Such methods are not learning-based methods. Various rule-based problem generators such as relation preserving model [23], rule-based structure generator [24] and analogical reasoning [30] are proposed in literature. The authors generate a large volume of dataset by applying a set of rules and use a single neural network to solve various types of DR problems. The methods utilize low-level image-based features for learning that demands a large volume of training samples. We have made a bridge between the modeling-based methods and learning-based reasoning. We have used a new feature representation of the RPMs, referred to as relational feature (RF) to construct a knowledge-base. RF shares similar fundamental concept of feature representation used in modeling-based methods discussed earlier. The features are extracted from low-level images using computer vision methods. They are then represented in a structured manner such that they can be used in a typical learning framework such as LSTM. Our method can learn new knowledge via training and solve the RPMs without computational modeling and rules. We have also found that different neural networks result into different accuracy for different problems, and there is no clear winner. It has been observed that high-level features are highly suitable for solving reasoning problems. However, extracting high-level information from low-level images can be complex. It has also been observed that different learning methods can be used to learn the low-level features such as the image and also the high-level features such as objects and relations. We have addressed the problem of extracting high-level features from low-level images, representation of the features as knowledge, and proposed a framework for learning low-level and high-level features.

The concept of relational feature is new and the proposed method can be useful in various image understanding problems in computer vision [5,22]. Further, the proposed framework can be useful for different artificial reasoning tasks such as intelligent tutor, digital assistant [46], intelligent robot [47], etc.

4. Conclusion

In this paper, we have introduced a new dataset for solving 4 × 1 DR problems using machine learning and computer vision. The dataset can be used by the CV research community for extending the research in this domain. We have experimented with several state-of-the-art learning frameworks to solve a variety of 4 × 1 DR problems. It has been observed that the image-based analysis usually fails to answer correctly in many cases. We have introduced a new feature-set referred to as relational features to solve 4 × 1 DR problems. Supervised learning with the help of LSTM has been used to classify the DR questions. Results reveal that the proposed framework outperforms existing image-based analysis.

It has been observed that the algorithmic pipeline defined in this work can be highly effective as it requires less samples for learning. However, the knowledge-base proposed in this work is relatively simple in nature and it may not be sufficient to solve complex DR problems. Therefore, it may be necessary to redefine the feature-set for solving complex DR problems. In particular, other types (OT) of DR problems need further attention of the research community.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors. Informed consent: Informed consent was obtained from all individual participants included in the study.

Declaration of Competing Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P5000 GPU used for this research.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2020.107412.

References

[1] P.A. Carpenter, M.A. Just, P. Shell, What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test., Psychol. Rev. 97 (3) (1990) 404–431.
[2] H.R. Burke, Raven'S progressive matrices: a review and critical evaluation, J. Genet. Psychol. 93 (2) (1958) 199–228.
[3] C. Diamantini, A. Freddi, S. Longhi, D. Potena, E. Storti, A goal-oriented, ontology-based methodology to support the design of aal environments, Expert Syst. Appl. 64 (2016) 117–131.
[4] Y. Zhou, Y. Sun, V. Honavar, Improving image captioning by leveraging knowledge graphs, in: IEEE Winter Conference on Applications of Computer Vision, 2019, pp. 283–293.
[5] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C.L. Zitnick, R. Girshick, Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1988–1997.
[6] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, R. Sebastiani, Reasoning with goal models, in: International Conference on Conceptual Modeling, Springer, 2002, pp. 167–181.
[7] G.W. Mineau, R. Godin, Automatic structuring of knowledge bases by conceptual clustering, IEEE Trans. Knowl Data Eng 7 (5) (1995) 824–829.
[8] L.D. Raedt, K. Kersting, S. Natarajan, D. Poole, Statistical relational artificial intelligence: logic, probability, and computation, Synth. Lect. Artif. Intell.Mach. Learn. 10 (2) (2016) 1–189.
[9] C.-U. Shin, J.-W. Cha, End-to-end task dependent recurrent entity network for goal-oriented dialog learning, Comput. Speech Lang. 53 (2019) 12–24.
[10] C. Huang, J. Li, C. Mei, W.-Z. Wu, Three-way concept learning based on cognitive operators: an information fusion viewpoint, Int. J. Approximate Reasoning 83 (2017) 218–242.
[11] M. Verbeke, V. Van Asch, R. Morante, P. Frasconi, W. Daelemans, L. De Raedt, A statistical relational learning approach to identifying evidence based medicine categories, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012, pp. 579–589.
[12] F. Yang, Z. Yang, W.W. Cohen, Differentiable learning of logical rules for knowledge base reasoning, in: Advances in Neural Information Processing Systems, 2017, pp. 2319–2328.
[13] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, Proc. IEEE 104 (1) (2015) 11–33.
[14] H. Jaeger, Artificial intelligence: deep neural reasoning, Nature 538 (7626) (2016) 467–468.
[15] L. Serafini, A.S.d. Garcez, Learning and reasoning with logic tensor networks, in: Conference of the Italian Association for Artificial Intelligence, Springer, 2016, pp. 334–348.
[16] S.M. Kazemi, D. Poole, Relnn: A deep neural model for relational learning, in: 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 6367–6375.
[17] A. Garcez, M. Gori, L. Lamb, L. Serafini, M. Spranger, S. Tran, Neural-symbolic computing: an effective methodology for principled integration of machine learning and reasoning, J. Appl. Logics 6 (4) (2019) 611–632.

[18] J. Mao, C. Gan, P. Kohli, J.B. Tenenbaum, J. Wu, The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision, in: International Conference on Learning Representations, 2019, pp. 1–28.

[19] J. Wang, W. Wang, L. Wang, Z. Wang, D.D. Feng, T. Tan, Learning visual relationship and context-aware attention for image captioning, Pattern Recognit. 98 (2020) 107075–107086.

[20] W. Wang, Y. Huang, L. Wang, Long video question answering: a matching-guided attention model, Pattern Recognit. 102 (2020) 107–248.

[21] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, D. Parikh, Vqa: Visual question answering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2425–2433.

[22] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, D. Parikh, Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering, in: Conference on Computer Vision and Pattern Recognition, 2017, pp. 6904–6913.

[23] A. Santoro, F. Hill, D. Barrett, A. Morcos, T. Lillicrap, Measuring abstract reasoning in neural networks, in: International Conference on Machine Learning, 2018, pp. 4477–4486.

[24] F. Hill, A. Santoro, D. Barrett, A. Morcos, T. Lillicrap, Learning to make analogies by contrasting abstract relational structure, in: International Conference on Learning Representations, 2019, pp. 1–14.

[25] M. Kunda, K. McGreggor, A. Goel, Addressing the ravens progressive matrices test of ǣgeneralg intelligence, in: AAAI Fall Symposium Series, 2009, pp. 22–27.

[26] A. Lovett, K. Forbus, J. Usher, A structure-mapping model of raven's progressive matrices, in: Proceedings of the Annual Meeting of the Cognitive Science Society, 32, 2010, pp. 2761–2766.

[27] M. Ragni, S. Neubert, Solving raven's iq-tests: an ai and cognitive modeling approach, in: Proceedings of the 20th European Conference on Artificial Intelligence, IOS Press, 2012, pp. 666–671.

[28] K. McGreggor, A. Goel, Confident reasoning on raven's progressive matrices tests, in: 28th AAAI Conference on Artificial Intelligence, 2014, pp. 380–386.

[29] A. Lovett, K. Forbus, Modeling visual problem solving as analogical reasoning., Psychol Rev. 124 (1) (2017) 60.

[30] C. Zhang, F. Gao, B. Jia, Y. Zhu, S.-C. Zhu, Raven: A dataset for relational and analogical visual reasoning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5317–5327.

[31] M. Kunda, K. McGreggor, A.K. Goel, A computational model for solving problems from the ravens progressive matrices intelligence test using iconic visual representations, Cogn. Syst. Res. 22 (2013) 47–66.

[32] A. Soni, D. Viswanathan, J. Shavlik, S. Natarajan, Learning relational dependency networks for relation extraction, in: International Conference on Inductive Logic Programming, Springer, 2016, pp. 81–93.

[33] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, Z. Liang, Apple detection during different growth stages in orchards using the improved yolo-v3 model, Comput. Electron. Agric. 157 (2019) 417–426.

[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations, 2015, pp. 19–36.

[36] P. Tang, H. Wang, S. Kwong, G-Ms2f: googlenet based multi-stage feature fusion of deep cnn for scene recognition, Neurocomputing 225 (2017) 188–197.

[37] Y. Tamaazousti, H. Le Borgne, C. Hudelot, M.E.A. Seddik, M. Tamaazousti, Learning more universal representations for transfer-learning, IEEE Trans. Pattern Anal. Mach. Intell. (2019) 1–12.

[38] T.N. Tran, K. Drab, M. Daszykowski, Revised dbscan algorithm to cluster data with dense adjacent clusters, Chemometr. Intell. Lab. Syst. 120 (2013) 92–96.

[39] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3156–3164.

[40] V. Vukotić, S.-L. Pintea, C. Raymond, G. Gravier, J.C. van Gemert, One-step time-dependent future video frame prediction with a convolutional encoder-decoder neural network, in: International Conference on Image Analysis and Processing, Springer, 2017, pp. 140–151.

[41] C. Lu, M. Hirsch, B. Schölkopf, Flexible spatio-temporal networks for video prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6523–6531.

[42] S. Yousfi, S.-A. Berrani, C. Garcia, Contribution of recurrent connectionist language models in improving lstm-based arabic text recognition in videos, Pattern Recognit 64 (2017) 245–254.

[43] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, in: Advances in Neural Information Processing Systems, 2015, pp. 802–810.

[44] T. Lan, T.-C. Chen, S. Savarese, A hierarchical representation for future action prediction, in: European Conference on Computer Vision, Springer, 2014, pp. 689–704.

[45] S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer, Scheduled sampling for sequence prediction with recurrent neural networks, in: Advances in Neural Information Processing Systems, 2015, pp. 1171–1179.

[46] A. Maedche, C. Legner, A. Benlian, B. Berger, H. Gimpel, T. Hess, O. Hinz, S. Morana, M. Söllner, Ai-based digital assistants, Bus. Inf. Syst. Eng. (2019) 1–10.

[47] L. Hu, Y. Miao, G. Wu, M.M. Hassan, I. Humar, Irobot-factory: an intelligent robot factory based on cognitive manufacturing and edge computing, Fut. Gen. Comput. Syst. (2019) 569–577.

Sk. Arif Ahmed has obtained his PhD from NIT Durgapur, India. Formally he was Assistant Professor of Computer Applications at Haldia Institute of Technology, India and currently working as a Postdoctoral research fellow in the Department of Physics and Technology, UiT The Arctic University of Norway, Norway. His areas of interestsinclude computer vision, image processing, microscopy and nanoscopy image analysis and Natural Language Understanding. He has already published several research articles in various reputed international journals and conferences. He is a member of IEEE and Digital Life Norway (DLN).

Debi Prosad Dogra is an Assistant Professor of Computer Science and Engineering in the School of Electrical Sciences at IIT Bhubaneswar, India. He received PhD from IIT Kharagpur in 2012, M.Tech from IIT Kanpur in 2003 and BTech from HIT in 2001, all in Computer Science and Engineering. He worked with ETRI, South Korea and Samsung Research Institute Noida prior to joining IIT Bhubaneswar. Dr. Dogra has published more than 85 research papers in international journals and conferences His research areas are computer vision, visual surveillance, and AR. He is a member of IEEE.

Samarjit Kar is currently a professor in the Department of Mathematics, National Institute of Technology, Durgapur, India. He received his Ph.D in Inventory Management in Uncertain Environment from Vidyasagar University in 2001. His current research interests include operations research and optimization, soft computing, uncertainty theory and financial modelling. He has published over 80 referred articles in international journals including Information Sciences, Applied Soft Computing, Applied Mathematical Modelling, European Journal of Operational Research, Computers and Operations Research, Computers and Industrial Engineering, International Journal of Production Economics, Applied Mathematics and Computation, and Soft Computing.

Partha Pratim Roy received his Ph.D. degree in computer science in 2010 from Universitat Autonoma de ‘ Barcelona, (Spain). He worked as postdoctoral research fellow in the Computer Science Laboratory (LI, RFAI group), France and in Synchromedia Lab, Canada. Presently, Dr. Roy is working as Assistant Professor at Indian Institute of Technology (IIT), Roorkee. His main research area is Pattern Recognition.

Dilip K. Prasad received the B.Tech. and Ph.D. degrees in computer science and engineering from the Indian Institute of Technology (ISM), Dhanbad, India, and Nanyang Technological University, Singapore, in 2003 and 2013, respectively. He is currently a senior research fellow with the Nanyang Technological University, Singapore. He has authored over 65 internationally peer-reviewed research articles. His current research interests include image processing, machine learning, and computer vision. Dr. Roy has won the best student paper award in International Conference on Document Analysis and Recognition (ICDAR) in 2009. His main research area is Pattern Recognition. He has published more than 120 research papers in various international journals, conference proceedings. He has gathered industrial experience while working as an Assistant System Engineer in TATA Consultancy Services (India) from 2003 to 2005 and as Chief Engineer in Samsung, Noida from 2013 to 2014.