DeepIQ: A Human-Inspired AI System for Solving IQ Test Problems

Jacek Mańdziuk
Faculty of Mathematics and Information Science
Warsaw University of Technology, Warsaw, Poland
mandziuk@mini.pw.edu.pl

Adam Żychowski
Faculty of Mathematics and Information Science
Warsaw University of Technology, Warsaw, Poland
a.zychowski@mini.pw.edu.pl

***Abstract*—This paper presents a neural network approach to solving the most common type of human IQ test problems – Raven's Progressive Matrices (RMs). The proposed *DeepIQ* system is composed of three modules: a deep autoencoder which is trained to learn a feature-based representation of various figure images used in IQ tests, an ensemble of shallow multilayer perceptrons applied to detection of feature differences, and a scoring module use for assessment of candidate answers. *DeepIQ* is able to learn the underlying principles of solving RMs (the importance of similarity of figures in shape, rotation, size or shading) in a domain-independent way, that allows its subsequent application to test instances constructed based on a different set of figures, never seen before, or another type of IQ problem, *with no requirement for additional training*. This transfer learning property is of paramount importance due to scarce availability of the real data, and is demonstrated in the paper on two different RM data sets, as well as two distinct types of IQ tasks (solving RMs and odd-one-out problems). Experimental results are promising, excelling human average scores by a large margin on the most challenging subset of RM instances and exceeding 90% accuracy in odd-one-out tests.**

I. Introduction

In psychology, there exist well-established and generally accepted methods to evaluate human intelligence by means of psychometric tests (IQ tests) measuring the intelligence quotient (IQ). In the case of Artificial Intelligence (AI), such comprehensive methods are yet to be developed, except for the well-known Turing Test [1]. The applicability of the Turing Test is, however, limited to assessing certain aspects of presumably intelligent, human-like machine behavior. Moreover, the assessment is subjective, and does not allow for quantitative evaluation. In 2003 Bringsjord and Schimanski attempted to bridge this gap and introduced the concept of *Psychometric AI* [2] which points to psychometric tests as a valid approach to defining and evaluating IQ of the AI systems and a viable alternative to task-oriented assessment methods.

Related work. In the mainstream AI / Machine Learning (ML) literature the number of papers related to solving IQ tests by machines is rather limited, although the topic gained visible interest in recent years. A survey article [3] on computer models solving intelligence-requiring problems mentions 30 papers, half of which have appeared since 2011. The first attempt was made over 50 years ago by [4] who designed a computer program for detecting regularities in figures in the so-called *geometric-analogy problems* (similar to IQ test instances). Subsequent papers were concentrated on various types of intelligence tests, for instance completion of sequences of numbers [5], words-related tests [6], verbal tests [7], Bennett's Mechanical Comprehension Tests [8] or general approaches to solving various types of IQ tests [9], [10].

The proposed *DeepIQ* system is evaluated on the problems which follow the rule-based construction principles of the most popular IQ tests - Raven's Progressive Matrices [11], referred to as RMs hereafter. There are three main reasons that motivated our choice of RMs as a testbed problem. Firstly, machine learning and testing on real IQ test examples is practically impossible due to their scarce public accessibility, as psychologists do not make them publicly available. Secondly, the most popular human IQ tests follow the same construction principles as RMs (but with different sets of figure shapes) [11]. Thirdly, RMs (likewise IQ tests) pose a real challenge for both humans [12] and (notably) artificial systems [3] and solving them requires specific cognitive abilities (image recognition, regularities detection, feature abstraction and generalization).

To the best of our knowledge, only a few following attempts to machine solving of IQ test problems were made so far. An approach proposed in [13] constructs a relational graph of detected image features, while in [14] a pre-defined rule-based system is applied. Both above approaches do not employ Machine Learning (ML) techniques. In a recent study [15] problems similar to RMs are used to determine the ability of certain neural network architectures - Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) or Residual Neural Networks (ResNETs) - to demonstrate abstract reasoning properties. As admitted by the authors, the IQ problems considered in [15] are only loosely inspired by the RM principles and differ from problems that humans taking Raven-style IQ tests are faced with. Another related recent paper [16] employs CNNs to generate a new image based on the rules detected in two other given images. In principle, such an ability could have been useful in solving RMs, however, only if the operational regime had been changed, i.e. the answer had not been generated by the system from scratch, but selected out of an available answer-set. The last paper [17],

Accepted for the International Joint Conference on Neural Networks (IJCNN 2019) - Budapest, Hungary, July 2019

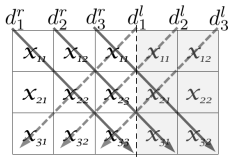
which is the closest to our approach, applies convolutional neural networks to determine stability of features across RM images and a set of rules worked out based on this stability information. Our method differs from [17] by utilizing other ML techniques (deep autoencoder and multilayer perceptrons) to assess relations among image features, and by checking the consistency of feature changes instead of their stability. Furthermore, *DeepIQ* is applied to a wider range of test problems, in terms of both *variability and difficulty level*. A distinctive feature of *DeepIQ* is its high efficacy also in the case when the training and testing sets rely on *different* sets of figures (shapes) used in RM instances.

It should be underlined that none of the previous approaches addressed the problem of limited availability of real-life test examples. Training and testing instances in the above-mentioned systems are sampled from the same distributions (the same available figure sizes, shapes, etc.) what does not reflect the real-life situation, in which the main attributes of the figures used in the test are not known in advance and the key difficulty for a human test-taker is to grasp the RM rules in an abstract (figure independent) way. The design of *DeepIQ* directly addresses this aspect.

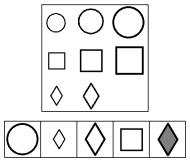
***DeepIQ* system.** The vast majority of solving methods proposed in the above-cited papers rely on application of specifically-defined sets of rules reflecting various types of possible interrelations between objects (numbers, words, shapes, etc.). In this paper we take a different approach and focus on the process of *learning* the feature-based representation and solution principles of the IQ test instances. The goal is accomplished by means of learning the relations between pairs of images in a way that allows for successful knowledge transfer and its inter-domain usage (transfer learning) [18], [19]. The main idea of transfer learning is applying knowledge gained during solving one problem (source domain) to solving different, but related problem (target domain). In a situation when no labeled data in the target domain are available and there are ample labeled data in the source domain, the problem is called *transductive transfer learning* (TTL) [19]. TTL was successfully applied, for instance, to text categorization [20], [21] or image recognition [22].

Our approach employs a deep autoencoder (AE) which is trained to learn a common feature-based representation of various figure images characterized by their *shape, rotation, size* and *shading*. Based on this common representation an ensemble of shallow multilayer perceptrons (MLPs) are trained to detect feature-related differences between a pair of figure images presented in their input layers. The outputs of these MLPs are then combined to provide the ultimate assessment of feature-based differences between the two input images. A baseline idea of *DeepIQ* is inspired by the human way of solving IQ tests [23].

Contribution. We attempt to design a system capable of efficiently solving the RM instances, based on the training process performed on the set of figures (shapes) different from those used in the test phase. The challenge consists in learning a feature-based RM representation in a general, domain-independent way that allows their subsequent application to RMs constructed based on new set of figures, previously unknown to the system. Experimental results prove high efficacy of the transfer learning approach implemented in *DeepIQ*. The system exceeds human average scores by a clear margin on a subset of the most challenging test instances.



(a) Left and right RM diagonals. Three of each kind.



(b) An RM example.

Fig. 1: RM diagonals and an RM example. A figure fitting the blank space is the middle one from the answer vector.

Furthermore, it is demonstrated that the same system, with only modified scoring function, can be successfully applied to a different type of IQ test problems (the odd-one-out problem) *with no need for additional training*.

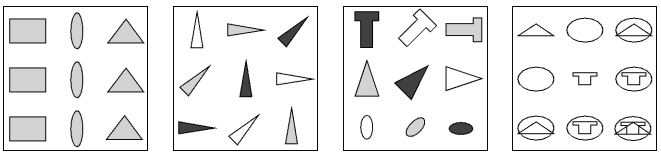
II. Raven's Matrices and human IQ tests

RMs represent the most popular kind of IQ tests. Their baseline idea was introduced by [24] and later on they have become one of the most popular methods of measuring human mental abilities [11]. Despite certain limitations, the main advantage of RMs is their universality, as they are independent of nationality, age, knowledge and language [12]. RMs are a well-established method, widely researched by psychologists [23] and used, for instance, in *Mensa* qualification tests [25].

A single test instance is in the form of a 3×3 grid, whose 8 cells contain appropriately chosen figures. The 9th, bottom-right cell is empty and the goal is to pick one out of 5 proposed candidate answers, to fill this blank cell, which fits the *set of relations* defining this particular RM. Each such relation is a rule that operates on the following attributes of the figures: shape, rotation, size and shading. These attributes may change vertically (row by row), horizontally (column by column) or diagonally, in each case according to a given rule. Please note, that each RM has 3 left and 3 right diagonals - see Figure 1a. An example of RM is presented in Figure 1b in which the size of figures changes in a column-wise manner. Clearly, the answer is the middle figure - a big-size diamond shape without shading. For further information regarding the design principles of IQ tests the reader can refer to [11].

Following [26] all RM tests can be categorized into 4 classes based on the number of relations that define them (cf. Figure 2):

- a single relation - only one feature of the figures changes in either rows, columns, left-diagonals or right-diagonals, the other features remain unchanged;
- two relations - exactly two features change;
- three relations - exactly three features change;



(a) RM with one relation. (b) RM with two relations. (c) RM with three relations. (d) RM with logic relation.

Fig. 2: Examples of possible types of RMs. In fig. (a), shapes of the figures change column-wise. In fig. (b), rotation of the figures and shading change respectively in left diagonals and right diagonals. In fig. (c), figure shapes in subsequent rows, rotation angles in subsequent columns and shading in subsequent left diagonals change simultaneously. In fig. (d), the right column is the result of a pixel-based logic OR operation of the left and middle columns. In this paper, test problems of types (a), (b) and (c) are considered.

- logic relations - AND, OR or XOR operations are performed on the figures in rows, columns or diagonals.

In this study we will concentrate on the first three groups. While logic relations may pose a challenge for humans, they are fairly easy to detect in autonomous way by machines, simply by checking all possible pixel-based logic operations in all 4 directions (rows, columns and diagonals).

III. *DeepIQ* APPROACH

The proposed *DeepIQ* system for solving RMs is composed of the three main modules:

- a deep AE which provides compressed representation of individual RM cells,
- four feature related MLPs,
- a scoring module.

The role of the first module is to learn a common, comprehensive representation of figures appearing in RMs. Its last layer provides a compressed 16-element real vector representation of a given RM cell presented as the input - see Figure 4.

The second module, an ensemble of 4 MLPs, is used to detect specific feature-based differences between the two cells presented in their input layers. These MLPs are trained on pairs of 16-element compressed AE representations of artificially generated RM cells (figures).

The third module computes a degree of fit of each of 5 candidate answers. First, a candidate answer is temporarily added to RM (in a blank cell), and then its level of fitness is determined based on the monotonicity of changes of all 4 figure features, separately in rows, columns, left diagonals and right diagonals.

Figure 3 presents an overview of the system information flow in the test phase. First, the RM which is to be solved is presented in the AE input layer. Next, all 8 cells are transformed into their compressed AE representations. Then, based on these AE representations, feature related MLPs detect differences between any two neighboring cells in rows, columns and diagonals. Finally, for each candidate answer, its fitness score is calculated based on the MLPs outputs, and the best fitted one is selected as an RM solution.

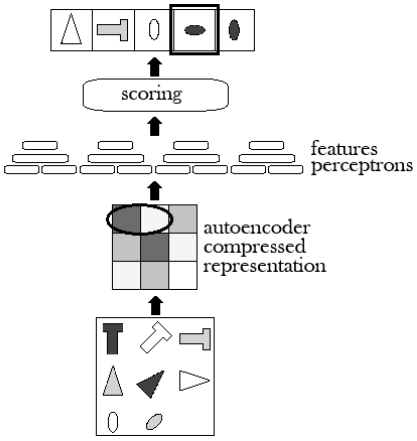


Fig. 3: *DeepIQ* information flow in the test phase.

The following subsections introduce the three above-mentioned *DeepIQ* components in more detail and further explain their usage in the test phase.

Deep autoencoder

Each of 9 cells of RM is represented as a square image of size $n \times n$ pixels which present one figure (except for the bottom-right cell, which is empty). Each of these 8 non-empty images is first transformed into a vector of n^2 integer numbers from the interval $[0, 255]$, which correspond to gray scale intensity of the pixels. Next, each such vector is compressed by AE to a 16-element real vector. AE is composed of 4 encoding layers, respectively with 2500 (input layer), 1024, 256, 16 (compression layer) neurons (as depicted in Figure 4) and 3 decompression layers with 256, 1024, 2500 (output layer) neurons, resp. All layers (shallow AEs) are trained with the *ReLU* activation function.

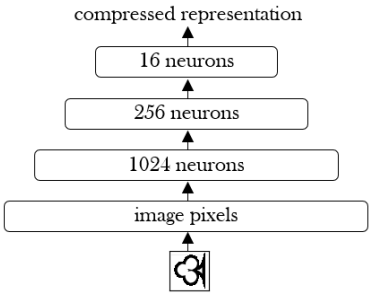


Fig. 4: Deep autoencoder architecture. Its last layer provides a compressed version (16 numbers) of the input image.

We decided to use deep AE architecture instead of CNNs (which are generally considered to be the first choice for image feature extraction tasks) for the two following reasons. Firstly, one of the main motivations in designing the system was to address the problem of highly limited availability of real-life test examples, and therefore build the underlying feature representation in the way that abstracts from particular RM figures to the highest possible extent. Consequently, instead of

using a discriminative model (such as CNN) an unsupervised approach was preferred. Secondly, CNNs were previously applied to solve RMs in [17] and [16], so their comparison with the outcomes of AE based approach seems to be an interesting and natural direction.

Feature related perceptrons

The role of this module is to percept feature-based differences between two figure images presented in the input. To this end 4 MLPs are used, each dedicated to detection of differences in one of the 4 examined features, i.e. shape, rotation, size and shading. Input pairs are RM cells compressed by deep AE, and the MLP output describes the way in which the first image differs from the second one regarding a particular feature, but with no reference to other features. For instance, the size-related MLP takes into account differences in size regardless of the shape, rotation and shading of the figures.

All 4 networks are 3-layer MLPs with 32 inputs (two cell representations), 16 neurons in the first hidden layer, 8 neurons in the second hidden layer, and either 2 or 5 output layer neurons, as explained below. In the output units a *Softmax* activation function is used and in all other layers the *ReLU* function is applied.

The shape-related MLP has two output units. The first one is active if no shape difference is detected, the other one activates if the shapes are perceived as different.

The remaining 3 MLPs (for detection of differences in rotation (r), size (s) and shading (h), resp.) are composed of 5 output neurons. Their output is interpreted as a degree to which a given feature in the first input image differs from the same feature in the second input image. More precisely, a possible range of changes of feature $k \in \{r, s, h\}$ is divided into 5 pairwise equal disjoint intervals d^k_j, $j \in \{0, 1, 2, 3, 4\}$. Activation of the m-th output neuron ($m \in \{0, 1, 2, 3, 4\}$) in the k-th MLP denotes the case $j_1 - j_2 = m$ (mod 5), where $d^k_{j_1}$ and $d^k_{j_2}$ are intervals to which a value of feature k belongs, in the first and the second image, resp. For example, the range of figure rotation feature is divided into the following intervals: $d^r_0 = [0, \frac{2}{5}\pi), d^r_1 = [\frac{2}{5}\pi, \frac{4}{5}\pi), d^r_2 = [\frac{4}{5}\pi, \frac{6}{5}\pi), d^r_3 = [\frac{6}{5}\pi, \frac{8}{5}\pi)$ and $d^r_4 = [\frac{8}{5}\pi, 2\pi]$. If, for instance, the rotation angle of the first figure equals $\frac{\pi}{2}$, i.e. $\in d^r_1$ and of the second one is equal to $\frac{3}{2}\pi$, i.e. $\in d^r_3$, then the third output neuron ($m = 2$) should be activated in the rotation-related MLP.

For the size feature, possible values range from $\frac{n}{2}$ to n as the size is interpreted as a maximum distance (in pixels) between two non-white pixels horizontally or vertically in straight (non rotated) figure. Possible values of shading range from 0 to 255 and denote the level of grayness of pixels inside the figure shape.

Please note, that the above-mentioned feature values are computed only for training instances during their generation. In the test phase they are estimated autonomously by the DeepIQ *system.*

Figure 5 presents an architecture of the feature-related module with two example images shown in the input. Since the input figures are of the same shape, the first (leftmost) output neuron fires in the shape-related MLP. A difference in rotation angle equals $\frac{7}{4}\pi$ and therefore fits the highest values interval (the rightmost output neuron is activated in the second MLP). The figures are of the same size, hence the first neuron is active in the size-related MLP output layer. The images differ in shading, the left one is around 50% gray and the right one is white. Consequently, the third output neuron is activated in the shading-related MLP.

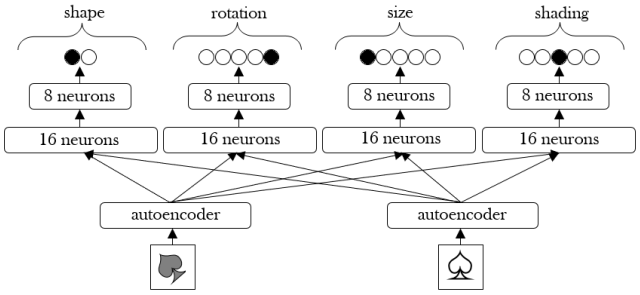


Fig. 5: An architecture of the feature-related module with two example figure images shown in the input. Figures have the same shape and size, differ "moderately" in shading and "significantly" by rotation angle.

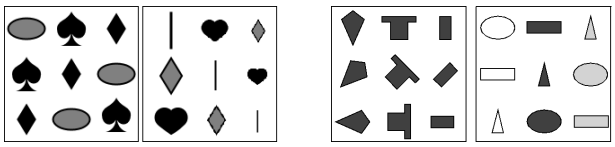
Answers scoring module

The third module evaluates a degree of fitness of each candidate answer. For each candidate answer, for each feature, patterns of changes in rows, in columns and in diagonals are detected by multiple use of the first two modules, and consequently, the estimated fitness score is computed for that answer. For instance, for a given feature (say rotation), the pattern of changes in a given row (say the middle one) is calculated by applying the above-described pair-based comparison twice: for cells $(2, 1)$ and $(2, 2)$, and cells $(2, 2)$ and $(2, 3)$ of a given RM, using the rotation-related MLP.

Formally, let $f^k(x_{i_1 j_1}, x_{i_2 j_2})$ be the output neuron with the highest activation value in the k-th feature related MLP (where $k \in \{p, r, s, h\}$, p denotes *shape*) in response to the compressed representation of images at positions (i_1, j_1) and (i_2, j_2) in the RM, respectively. f^k is interpreted as a *distance of feature* k between these two images. Scoring s_{a_i} of candidate answer $a_i, i = 1, \ldots, 5$ is computed as follows. Image a_i is placed in the blank field of RM (position $(3, 3)$) and the following procedure is executed:

$s_{a_i} := 0$
if $f^k(x_{11}, x_{12}) = f^k(x_{12}, x_{13})$ **and**
  $f^k(x_{21}, x_{22}) = f^k(x_{22}, x_{23})$ **then**
    if $f^k(x_{31}, x_{32}) = f^k(x_{32}, x_{33})$ **then**
      $s_{a_i} := s_{a_i} + 1$
    else
      $s_{a_i} := s_{a_i} - 1$

In the above listing, first the consistency of changes, related to feature k, between the left and middle vs. middle and right elements in the first two rows is verified. If the changes are



Generated tests. Sandia tests.

Fig. 6: Example G-set and S-set instances.

consistent, the same pattern of changes is expected to appear also in the last row. If this expectation is confirmed, the scoring is incremented, otherwise, it is decremented. In the case the changes tested in either of the two *if statements* are inconsistent, the score does not change, as it is assumed that the rules defining this particular RM are not applied in rows but in other direction(s).

The above procedure is repeated for each of the 4 features. Analogous procedures are also executed for columns, right and left diagonals and the scores s_{a_i} are summed up. Consequently, the final score is an integer from interval $[-16, 16]$. The highest possible score would mean a consistence of monotonicity of all 4 features in all 4 directions. A candidate answer with the highest score is selected as the RM solution.

In summary, the proposed system is motivated by design principles of IQ test problems and furthermore inspired by the human way of solving these puzzles. First of all, people are very efficient in extraction of plain features from the images. In *DeepIQ* this task is accomplished by the first module - deep AE. In the next step, humans determine differences between particular figures with respect to the extracted features what corresponds to the MLP-based feature comparison. The last module (a scoring method) is motivated by the results of an experiment with the gaze focus movement tracing of humans solving IQ tests [23], which revealed that in most of the cases people first examine an RM by rows, columns and diagonals, and only then select an answer.

IV. Experimental evaluation

As we have mentioned in the Introduction, one of the biggest impediments when working on ML approaches to solving human IQ tests is the lack of publicly available representative sets of example problems. IQ tests are strictly confidential and larger databases are only available to qualified psychologists and kept in high secrecy by them. For this reason we could not rely on real IQ test examples, and therefore, training of *DeepIQ* was performed on a set of artificially generated instances (denoted by *G-set*) that follow design principles of IQ tests, but use a set of figures different from that utilized in the real tests. In the test phase, two scenarios were considered. In the first one, new examples from G-set (unknown to the system) were used to test the in-domain system efficiency. In the other one, an entirely different data set (*S-set*), with figure images not related to the training data and not presented to the system beforehand, was used.

Training data and training procedure

Training set was composed of 5000 figures with shapes randomly sampled from a predefined set of 15 figures, presented in Figure 7 (top row). For each sampled figure, its size, rotation and shading were randomly selected from the following ranges: size (width and height) from 25 to 50 pixels, rotation from $[0, 2\pi)$ interval (a real value), and shading - integer from $[0, 255]$. Each figure was then placed centrally in a 50×50 pixels square cell to form the final figure image. Deep AE was trained on 5000 samples of the above-described through 1000 epochs. Examples of AE post-training reconstruction quality are presented in Figure 8.

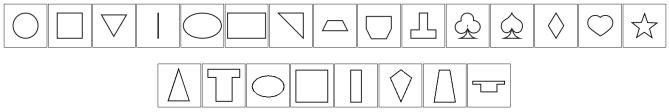


Fig. 7: A set of 15 figure images used to design artificially generated RMs in the training phase (top row), and figure images from Sandia tests (bottom row).

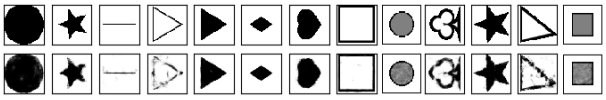


Fig. 8: Sample images (top row) and their autoencoder reconstructions (bottom row). High degree of pairwise similarity illustrates high quality of AE compressed representation.

The second *DeepIQ* module, composed of 4 feature-related MLPs, was trained on pairs of images. For each MLP a training pair was generated in the following way. First, a new image with randomly selected figure features was generated according to the same procedure as in the case of AE training. Then, the second image was obtained by copying the first one and randomly changing the value of one particular feature to which a given MLP was devoted. Images were presented in the MLPs input layers in their compressed form, encoded by AE. Each of the 4 MLPs was trained on 5000 pairs of images through 250 epochs. These parameters were set experimentally based on a couple of runs and error rate observation on a validation data (a different set of 500 pairs). Yet another set of 1000 pairs was generated for post-training evaluation, whose results are presented in Table I. As can be seen in the table, differences between two input images with respect to each of the 4 features were detected with high accuracy close to or above 90%.

Feature	shape	rotation	size	shading
Training	92.2% ± .2%	94.3% ± .2%	93.8% ± .3%	95.7% ± .1%
Post-training	87.3% ± .4%	91.4% ± .5%	90.3% ± .3%	92.4% ± .4%

TABLE I: Averaged accuracy of feature-related MLPs in the training and post-training evaluation.

Testing data

Two sources of RMs were used in the final evaluation of the *DeepIQ* system.

G-set. The first set (G-set), already mentioned in the context of AE training, was generated in accordance with the RM design principles described in the literature [23], [14]. While creating a new test instance, our RM generator randomly sampled direction (rows, columns, or any of the two types of diagonals) and type (increase, decrease) of a change and then applied it to a randomly sampled feature. A total of 1500 RMs with different difficulty levels (500 for each number of relations described previously and illustrated in Figure 2) were generated [27]. Figure images in these tests were randomly selected using the same procedure and the same set of 15 shapes depicted in the top row of Figure 7, as in the training process.

S-set. Sandia matrices (denoted by S-set) were developed at Sandia National Laboratory by [26] with the matrix generation software. The set of figure shapes used for creating the S-set is presented in the bottom row of Figure 7. A subset of 500 problems from this study was used in our experiments for the evaluation of both the *DeepIQ* performance and the transductive transfer learning properties of the system.

Please observe that although both sets refer to the same base problem (solving RMs) their *figure distributions* are completely different. Not only the S-set contains figures of different shapes than in G-set, but also uses *discrete sets of possible feature values*: 4 degrees of shading, 8 values of rotation angle (multiples of $\frac{\pi}{4}$), and 5 predefined values for size. On the contrary, feature values in G-set are *within their full-range*, i.e. all 256 shading intensities, all rotation angles between 0 and 2π (real values), as well as all sizes between 25 and 50 pixels in each dimension are possible.

Test results

Due to severely limited availability of the real IQ test data, training procedure was performed on G-set instances, and both G-set (its another part) and S-set were used for testing the accuracy of the system and its *transfer learning* ability.

Experimental results of *DeepIQ* evaluation on both RM sets with various problem difficulty levels are presented in Table II. On G-set problems *DeepIQ* achieved the average accuracy of 74.7%, which dropped slightly to 71.8% on S-set instances. Considering the fact that the shapes, as well as shadings, rotations and sizes of G-set figures clearly differ from those of Sandia set, we believe that qualitatively comparable performance on both sets should be mainly attributed to the design of the training procedure of the *DeepIQ* modules which facilitates transfer learning. Universality of AE figure representation and the focus on differences between feature attributes instead of their absolute values in the MLPs training, are most probably the decisive factors in the overall very promising performance.

The last row of Table II presents human results on Sandia matrices [26]. Surprisingly, a correlation between a difficulty level and accomplished results differs between humans and *DeepIQ*. Intuitively, the more relations in the test, the more difficult it is, and this rule clearly applies to humans. However, for *DeepIQ*, the higher difficulty (in human standards) means the higher correctness. A possible explanation is that in the tests with more relations, a score calculated by the third *DeepIQ* module is potentially higher, since the correct answer fulfills more rules. Consequently, the gap between scores assigned to correct and incorrect answers is bigger, what results in greater tolerance of the system to potential errors made in the lower modules.

	TR $\rightarrow$ TS	1 relation	2 relations	3 relations
DeepIQ	G $\rightarrow$ G	73.3% $\pm$ .7%	74.1% $\pm$ .5%	76.0% $\pm$ .6%
DeepIQ	G $\rightarrow$ S	70.2% $\pm$ .4%	71.9% $\pm$ .6%	73.2% $\pm$ .2%
Humans	$\rightarrow$ S	87.0%	72.0%	55.0%

TABLE II: *DeepIQ* accuracy on the RMs generated by the authors (G-set) and on Sandia RMs (S-set). The results are averaged over 10 independent runs, i.e. in each case the system was trained 10 times on randomly generated training sets and tested on a common test set to verify repeatability of the training process. The last row presents human results on the S-set.

To the best of our knowledge, there is only one work devoted to solving RMs with deep learning techniques [17], however relying on different RM difficulty categorization, what makes a direct comparison infeasible. On a subset of 108 Sandia RMs including both 1-relation and 2-relation problems (no results were reported for 3-relation RMs) the accuracy reported in [17] reached the level of 78.7%, clearly higher than that of *DeepIQ*. At the same time, it should be stressed that while the system presented in [17] was particularly tuned to solving S-set RMs with 1 or 2 relations, our approach allows solving also more difficult RMs with 3 relations, for which the system really demonstrates its strength and advantage over humans. Furthermore, the range of feature changes in *DeepIQ* design is *continuous*, as opposed to *discrete* steps of changes implemented in [17]. Moreover, a design of the proposed system allows its training on one data set and testing on another data set, composed of figures of different shapes. To the best of our knowledge such a property has not been demonstrated in any of the previous works.

Supplementary tests

In order to make a comparison with [17] fair, the experiment was repeated with a new D-set, created in exactly the same way as G-set, except that possible values of features were restricted to predefined small discrete sets - similarly to the S-set generation. The results are presented in Table III. Clearly, the restriction of feature values to discrete-valued subsets, improved *DeepIQ* accuracy on test instances from both D-set (out-of-sample) and S-set (whose design parameters were restricted in a similar way).

While these results are encouraging and on par with those of [17], we would like to emphasize that the main assets of *DeepIQ* are *transfer learning capability* and the ability of dealing with *continuous ranges of features*. Both these aspects give promise for system application to other types of IQ problems, as well as its utilization in other domains.

Due to continuous ranges of features, their certain combinations may lead to RMs which are very demanding, also for humans. This is particularly the case when figure sizes in the answer set are close to the threshold value and fall into

	TR → TS	1 relation	2 relations	3 relations
DeepIQ	D → D	78.1% ± .4%	79.2% ± .3%	80.3% ± .3%
DeepIQ	D → S	76.7% ± .2%	78.2% ± .2%	79.4% ± .4%

TABLE III: Accuracy of *DeepIQ* on the generated RMs *with restricted feature ranges* (D-set). In each case the results are averaged over 10 independent runs - see caption of Table II.

different categories despite tiny size differences. Three test instances of this kind are presented in Figure 9.

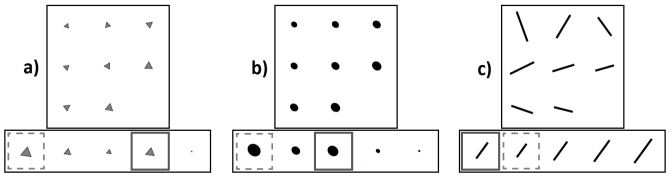


Fig. 9: Examples of challenging RM tests. A correct answer is outlined with a solid line and the one proposed by *DeepIQ* with a dashed line. In all three cases the differences are minute.

Odd-one-out

In order to support our claim about potential applicability of *DeepIQ* to other types of IQ tests, the system was employed to solving the *odd-one-out* (O1O) problem, which is less popular than RMs, but still used in IQ tests [28]. The task in O1O is to point the *oddest* figure image out of given n images, i.e. the one which is the most distinct with respect to size, shape, shading, rotation. Besides its usage in IQ tests, O1O has certain practical applications, for instance in waste selective sorting [29].

Due to the lack of formal definition of the problem, there exist some ambiguity in interpretation of the *oddest* term [30]. In order to address this problem and eliminate the possibility of ambiguous answers we adopted the following O1O construction rule. In each O1O instance composed of n figures yielded by our generator, there exists exactly one subset of $n-1$ figures with a common subset of $c \in \{1,2,3\}$ features, and these features are different in the remaining figure image. Under such assumption the remaining figure is, by definition, treated as the *oddest* one. Even though in certain cases the above definition of the oddest figure may be disputable, it was adopted in this work as it leads to well-defined problem instances with solutions generally consistent with intuition. The number of figures (n) and the value of c determine the level of difficulty of an instance. Four examples of generated O1O problems are presented in Figure 10.

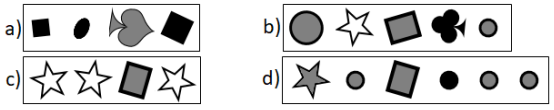


Fig. 10: Example odd-one-out tests. a) the third image is an answer (different shading - all the others are black). b): all figures but the last one are big. c): a rectangle has a different shape and shading. d): the answer is the 4th figure - all the others are gray.

		Number of figures (n)					
		4		5		6	
		G-set	S-set	G-set	S-set	G-set	S-set
c	1	93.2% ± .4%	91.5% ± .2%	93.1% ± .2%	91.3% ± .1%	93.3% ± .4%	91.1% ± .2%
	2	95.2% ± .2%	91.2% ± .5%	95.5% ± .4%	92.1% ± .2%	95.2% ± .4%	92.1% ± .1%
	3	96.2% ± .4%	92.5% ± .1%	96.4% ± .3%	92.7% ± .4%	96.3% ± .3%	92.9% ± .3%

TABLE IV: *DeepIQ* accuracy on G1-set and S1-set for various problem sizes (n) and numbers of common features in the subset of $n-1$ figures (c). In each case the results are averaged over 10 independent runs - see caption of Table II.

The first two modules (deep AE and feature related MLPs) of the trained instance of *DeepIQ* previously used to solving RMs were applied to the O1O task, *with no additional training or tuning of any kind*. Knowledge about relations between pairs of images, learnt in the RM related experiment based on G-set training figures, was *used directly* in the new type of problem. Only the scoring module was adopted to address specificity of the new problem.

Test results. Similarly to solving RMs, tests were executed on two data sets - the instances generated based on the shapes presented in the top row of Figure 7 (G1-set) and Sandia-inspired shapes (S1-set) depicted in the bottom row of Figure 7.

Table IV presents *DeepIQ* accuracy for various selections of c and n, for both data sets - in each case based on 1000 test images [27]. In all cases the results exceed 90%. Similarly to the case of solving RMs, the accuracy on G1-set is slightly higher (about $1-2\%$) than on the S1-set, and additionally, the higher number of relations defining an instance (more features in common and/or bigger problem size) results in higher correctness. Despite our efforts, we could not find any other AI system designed to solving O1O problems and therefore cannot make any literature comparisons. We believe, however, that the accuracy above 90% without any re-training or tuning of the system, confirms potential of *DeepIQ* to solve other types of relation-based IQ test problems, and consequently its generality. Please recall that, similarly to the case of RMs, continuous ranges of possible figure transformations may lead to puzzles which are challenging even for humans. Three examples of this kind are presented in Figure 11.

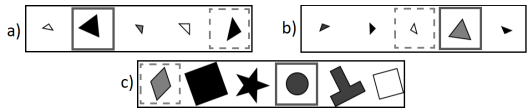


Fig. 11: Examples of challenging odd-one-out tests. A correct / *DeepIQ* answer is outlined with a solid / dashed line, resp. a) all triangles are right except for the 2nd one which is equilateral. b) a distinctive feature is *size* (*not shading* as chosen by the system). c) all figures but the 4th one are in the same *size* category (the 4th one falls into smaller range).

V. Conclusions

The paper presents an ML approach to solving IQ test problems in a realistic scenario, in which the availability of training samples is strictly limited and the exact design of figures

(shape, size, rotation and shading) is not known a priori. For this reason, the first two components of the system (deep AE and feature-related MLPs) are trained on artificially generated images, with no knowledge about the appearance of target images. The results of experimental evaluation of *DeepIQ* indicate that the system is able to *transfer feature-related knowledge learnt in one set of images (G-set) to another set of – significantly different – images (S-set)*. The first two system components are to a large extent universal and could be applied to other types of IQ test problems or to other tasks which are based on discrimination between feature values, e.g. in Biometrics. The above statement about universality of the system was supported by additional experiments performed with odd-one-out problems, whose underlying concept and specificity are different from those of RMs. While solving RMs relies on detection and quantification of similarities, the odd-one-out tests are focused on feature based differences between figure images. *Both tasks are solved using the same system with one common training phase and without any additional problem-related tuning or re-training.* The only changes are introduced in the scoring module which, by definition, is devoted to reflect specificity of the solved problem.

Promising results (accuracy above 90%) support the claim about the system's ability to transfer knowledge not only between different data sets within the same IQ task, but also between different types of IQ problems.

The paper demonstrates *DeepIQ* ability to detect differences (and assess their range) in figure shapes, sizes, rotations and shadings. On a general note, it seems reasonable to assume that in a similar manner could it be applied to determine differences between other image features (e.g. the number of figure edges or the number of elements in multi-figure images). Verification of this claim is one of our current research goals.

ACKNOWLEDGMENT

This work was supported by the National Science Centre, Poland, grant number 2017/25/B/ST6/02061.

REFERENCES

[1] A. M. Turing, "Computing machinery and intelligence," in *Parsing the Turing Test*. Springer, 2009, pp. 23–65.

[2] S. Bringsjord and B. Schimanski, "What is artificial intelligence? Psychometric AI as an answer," in *IJCAI*, 2003, pp. 887–893.

[3] J. Hernández-Orallo, F. Martínez-Plumed, U. Schmid, M. Siebers, and D. L. Dowe, "Computer models solving intelligence test problems: Progress and implications," *Artificial Intelligence*, vol. 230, pp. 74 – 107, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370215001538

[4] T. G. Evans, "A heuristic program to solve geometric-analogy problems," in *Proceedings of the April 21-23, 1964, spring joint computer conference*. ACM, 1964, pp. 327–338.

[5] C. Strannegård, M. Amirghasemi, and S. Ulfsbäcker, "An anthropomorphic method for number sequence problems," *Cognitive Systems Research*, vol. 22-23, pp. 27 – 34, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389041712000356

[6] P. Sanghi and D. L. Dowe, "A computer program capable of passing IQ tests," in *4th Intl. Conf. on Cognitive Science (ICCS'03), Sydney*, 2003, pp. 570–575.

[7] S. Ohlsson, R. H. Sloan, G. Turán, and A. Urasky, "Verbal IQ of a four-year old achieved by an AI system," *Age*, vol. 4, no. 5, p. 6, 2013.

[8] M. Klenk, K. Forbus, E. Tomai, and H. Kim, "Using analogical model formulation with sketches to solve Bennett Mechanical Comprehension Test problems," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 23, no. 3, pp. 299–327, 2011.

[9] M. Ragni and A. Klein, "Solving number series-architectural properties of successful artificial neural networks." in *IJCCI (NCTA)*, 2011, pp. 224–229.

[10] F. Martínez-Plumed, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana, "A computational analysis of general intelligence tests for evaluating cognitive development," *Cognitive Systems Research*, vol. 43, pp. 100–118, 2017.

[11] J. C. Raven and J. H. Court, *Raven's progressive matrices and vocabulary scales*. Oxford Psychologists Press Oxford, UK, 1998.

[12] J. Raven, "The Raven's progressive matrices: change and stability over culture and time," *Cognitive psychology*, vol. 41, no. 1, pp. 1–48, 2000.

[13] C. Strannegård, S. Cirillo, and V. Ström, "An anthropomorphic method for progressive matrix problems," *Cognitive Systems Research*, vol. 22-23, pp. 35 – 46, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389041712000435

[14] M. Ragni and S. Neubert, "Analyzing Raven's Intelligence Test: Cognitive model, demand, and complexity," in *Computational Approaches to Analogical Reasoning: Current Trends*. Springer, 2014, pp. 351–370.

[15] D. Barrett, F. Hill, A. Santoro, A. Morcos, and T. Lillicrap, "Measuring abstract reasoning in neural networks," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 511–520. [Online]. Available: http://proceedings.mlr.press/v80/barrett18a.html

[16] D. Hoshen and M. Werman, "Iq of neural networks," *arXiv preprint arXiv:1710.01692*, 2017.

[17] C. S. Mekik, R. Sun, and D. Y. Dai, "Deep learning of Raven's Matrices," *Advances in Cognitive Systems*, 2017. [Online]. Available: http://www.cogsys.org/papers/ACS2017/ACS_2017_paper_23_Mekik.pdf

[18] S. Thrun and L. Pratt, Eds., *Learning to Learn*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[20] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 77–82.

[21] B. Quanz and J. Huan, "Large margin transductive transfer learning," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1327–1336.

[22] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *Advances in neural information processing systems*, 2013, pp. 46–54.

[23] P. A. Carpenter, M. A. Just, and P. Shell, "What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test." *Psychological review*, vol. 97, no. 3, p. 404, 1990.

[24] J. C. Raven, "Mental tests used in genetic studies: The performances of related individuals in tests mainly educative and mainly reproductive," Master's thesis, University of London, 1936.

[25] B. Mensa. (2017) What is an IQ Test? http://www.mensa.org.uk/what-is-an-iq-test. [Online]. Available: http://www.mensa.org.uk/what-is-an-iq-test

[26] L. E. Matzen, Z. O. Benz, K. R. Dixon, J. Posey, J. K. Kroger, and A. E. Speed, "Recreating Raven's: Software for systematically generating large numbers of Raven-like matrix problems with normed properties," *Behavior research methods*, vol. 42, no. 2, pp. 525–541, 2010.

[27] "Traing and testing RM and O1O instances." 2018, url: github.com/deepiq/deepiq.

[28] M. Gardner and D. Richards, *The colossal book of short puzzles and problems*. Norton, 2006.

[29] J. Sinapov and A. Stoytchev, "The odd one out task: Toward an intelligence test for robots," in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*. IEEE, 2010, pp. 126–131.

[30] K. Tanya, "Odd One Out," Department of Mathematics, MIT, Tech. Rep., 2010. [Online]. Available: https://arxiv.org/pdf/1005.2700.pdf