Pattern Analysis and Applications manuscript No.
(will be inserted by the editor)

Solving the Same-Different Task with Convolutional Neural Networks

Nicola Messina · Giuseppe Amato · Fabio Carrara · Fabrizio Falchi · Claudio Gennaro

Received: date / Accepted: date

Abstract Deep learning demonstrated major abilities in solving many kinds of different real-world problems in the computer-vision literature. However, they are still strained by simple reasoning tasks that humans consider easy to solve. In this work, we probe current state-of-the-art convolutional neural networks on a difficult set of tasks known as the *same-different* problems. All the problems require the same prerequisite to be solved correctly: understanding if two random shapes inside the same image are the same or not. With the experiments carried out in this work, we demonstrate that residual connections, and more generally the skip connections, seem to have a negligible impact on the learning of the proposed problems. In particular, we experiment with DenseNets, and we examine the contribution of residual and recurrent connections in already tested architectures, ResNet-18 and CorNet-S respectively. Our experiments show that older feed-forward networks, AlexNet and VGG, are almost unable to learn the proposed problems, except in some specific scenarios. We show that recently introduced architectures can converge even in the cases where the important parts of their architecture are removed. We finally carry out some zero-shot generalization tests and we discover that in these scenarios residual and recurrent connections can have a stronger impact on the overall test accuracy. On the four difficult problems we take in consideration from the SVRT dataset, we can reach state-of-the-art results with respect to the previous approaches, obtaining super-human performances on three of the four problems.

Keywords AI · Deep Learning · Abstract Reasoning · Relational Reasoning · Convolutional Neural Networks

This work was partially supported by AI4CHSites, CNR4C program, CUP B15J19001040004 and by the AI4EU project, funded by the EC (H2020 - Contract n. 825619).

Nicola Messina
via Moruzzi, 1 - 56124 Pisa, Italy
Tel.: +39-050-6213054
E-mail: nicola.messina@isti.cnr.it

1 Introduction

With the advent of deep learning, the computer-vision world gained a huge boost in almost all its fields. One of the major innovations in this field was the introduction of Convolutional Neural Networks (CNNs) [13]. CNNs constitute now a standard approach to transform a raw matrix of pixels into some higher-level representation, and they are used in many downstream tasks.

In particular, with the work carried out on the ImageNet challenge, Krizhevsky et al. [11] demonstrated the major capabilities of deep neural networks, and CNNs in particular, in the task of image classification, which is one of the most important computer vision problems.

Recently, deep convolutional architectures have defined the state-of-the-art on multiple computer vision tasks, such as image classification [6, 14], object detection and segmentation [5, 18], multimedia and cross-media information retrieval [2, 23], and detection of adversarial examples [3].

Despite their success, there are many open problems with current deep architectures, and in particular with their abstract reasoning abilities. In fact, they are still unable to distill high-level general concepts that can be transferred to different domains. This brings to low generalization abilities and often to an overfit to the specific domain on which the network is trained. Humans can recognize some shape patterns never seen before, and they can deduce some general properties of a never seen shape (e.g., is it a closed shape? is it the same shape as another one but rotated?).

In this work, we try to probe modern state-of-the-art vision architectures to understand their ability to draw abstract conclusions on the objects contained in images. We tackle the *same-different* tasks, which consist of a set of problems having a common underlying difficulty: predicting if two shapes inside the same image are the same or not. It is a challenging set of tasks for convolutional architectures since they are required not to learn specific shape patterns to solve the problem; instead, they require to grow some abstract internal representation which is powerful enough to draw a logical conclusion on a fact hidden in the image (e.g., the shapes in the images are the same even if they are orientated in different ways).

The same-different problems are really interesting challenges, although they are not important on their own. They serve as guidance for solving a whole set of more specific tasks, having a real-world application in many contexts and disciplines. For example, these neural network abilities may have a great impact on classification problems, on the research of particular patterns in cultural heritage, and the detection of patterns defining aesthetic beauty in images and even music. In fact, the world is often perceived by humans as a set of recurrent structures composited together, such as the human eyes in a face or the repeating chorus in a song; understanding the underlying relationships between these repeating patterns is a must for developing a human-like intelligence.

This paper is an extension of our previous work [1], which developed upon the studies carried out by Kim et al. [10]. In [1], we tested a variety of state-of-

the-art deep convolutional architectures, originally designed for image classification, on some challenging *same-different* tasks. The analyzed problems are drawn from the SVRT dataset [4], a collection of images specifically generated to test the reasoning abilities of machine learning models on the same-different tasks. Preliminary experimental results suggested that residual connections could have been the most important architectural detail for solving this task.

In this work, instead, we show through an improved set of experiments that ResNet-like skip connections are probably not the only essential architectural details triggering the convergence. In this regard, out intention is to probe an extended set of state-of-the-art models. In particular, we try to remove the key distinctive architectural features from some of the previously probed models, such as residual connections from ResNets or residual and/or recurrent connections from the CorNet-S architectures.

We also demonstrate that even DenseNets can reach excellent results on the same-different problems; we show that the older VGG-19 and AlexNet architectures, in some specific cases, are able to move away from pure chance accuracy on the test set, although they cannot reach state-of-the-art results on the presented problems.

Instead, by targeting zero-shot generalization tests on converged architectures, we show that residual and recurrent connections can have possibly strong impacts on the final test accuracies.

Despite the underlying difficulty in discerning what are the key architectural factors driving convergence and generalization abilities of these networks on the analyzed problems, we think that this work evidences in a systematic way the current weaknesses of actual vision models as far as the same-different tasks are concerned.

To sum up, in this paper, we extend the previous work [1] with the following:

- we train a new set of state-of-the-art deep image classification networks on the same-different tasks;
- we probe in more detail already examined models, by removing some key architectural elements to understand to what extent they are able to converge.
- we perform an extensive evaluation of the newly trained architectures both in terms of convergence and generalization abilities, reporting training and validation curves.

The rest of the paper is organized as follows: in Section 2, we review some of the related work concerning abstract reasoning capabilities of deep neural networks; in Section 3, we briefly describe the same-different problems of interest from the SVRT dataset, and the models that we will probe to solve them; in Section 4, we introduce the models that we will probe on the presented problems, together with the overall general architecture that we will use for training them; in Section 5, we will present our experimental setup and discuss the obtained results; finally, in Section 6 we draw some conclusions on the sys-

tematic study performed in this work, and we will propose future directions for this research.

2 Related Work

It has been proven that deep neural networks can obtain remarkable performance in a wide variety of computer vision fields. However, relatively few works have tackled the cognitive and abstract reasoning capabilities of state-of-the-art deep neural networks.

Recently, the introduction of on-purpose generated benchmarks such as CLEVR and Sort-of-CLEVR [7] has paved the way towards fine-grained studies on relational capabilities of neural networks. CLEVR is a synthetic dataset composed of 3D rendered scenes, and it has been designed for the Relational Visual Question Answering (R-VQA) task, which consists in answering questions about complex dispositions of simple 3D shapes. Sort-of-CLEVR is a simpler version of CLEVR dealing with 2D shapes and a mix of relational and non-relational questions.

In particular, Santoro et al. [20] developed a relational reasoning module able to solve the CLEVR R-VQA task by augmenting a standard convolutional network with a reasoning module able to process all possible couples of objects. Slight variations of this network, the 2S-RN and the AVF-RN [16, 17], have been used for relational content-based image retrieval (R-CBIR), to extract relationship-aware visual features for indexing purposes.

Transparency and explainability are considered key objectives for understanding what kind of reasoning neural networks are internally performing. To this aim, on the CLEVR dataset, Johnson et al. [8] used explicit module composition to build an explicit reasoning pipeline, while Mascharka et al. [15] proposed a set of primitives which, when composed, manifest as a model capable of performing complex reasoning tasks in an explicitly-interpretable manner. In particular, they reached more than 99% accuracy on the CLEVR test set.

Other than CLEVR, other difficult tasks, such as Raven's Progressive Matrices (RPM), have been proposed to test in great detail, and under fully-controlled environments, reasoning capabilities of a given architecture. RPMs consist of visual geometric designs with a missing part. The test taker is given a small number of different choices to pick from and fill in the missing piece.

In particular, Zhang et al. [25] worked on RPMs and tried to establish a semantic link between vision and reasoning by providing structured representation. Similarly, Barret et al. [19] used an on-purpose created dataset similar to RPM, called Procedurally Generated Matrices (PGM). They demonstrated that popular models such as ResNets perform poorly on this benchmark, and they presented a novel architecture demonstrating quite effective reasoning capabilities. Differently from [19], our work concentrates on a very simple yet challenging problem, the *same-different* challenge.

The work by Yang et al. [24] introduced a synthetic dataset composed of sequences of 2D images to test memorization capabilities of neural networks. Fleuret et al. [4] introduced SVRT, a simple dataset composed of 2D shapes developed to test the comparison and the relational capabilities of neural networks. Stabinger et al. [21] first showed that problems involving comparisons between SVRT shapes were difficult for convolutional architectures like LeNet and GoogLeNet [22]. By contrast, instead, the traditional boosting method used by [4] was able to reach very good results even on comparison problems.

Even Kim et al. [10], in their recent work, found that the *same-different* problem strains simple feed-forward convolutional networks. Recently, Messina et al. [1] demonstrated that some state-of-the-art deep learning architectures for classifying images, in particular ResNet models, can learn this task, generalizing to some extent.

3 Review of the Same-different Problems

Every same-different problem comes with a certain rule that should be discovered by solely looking at some images. Despite having different rules, the same-different problems have a common underlying difficulty: discerning if two objects in the same image are the same or not, under different sets of geometric transformations.

A wide collection of same-different problems has been produced in [4], in the so-called SVRT dataset. SVRT consists of simple 2D images containing simple black closed curves on a white background. Every visual problem in SVRT is divided into two classes: the set of positive examples, which are the images that satisfy the specific rule, and the set of negative examples, which collect the images not satisfying the rule.

SVRT collects a total of 23 different visual problems, that can be further divided into two clusters: problems related to the spatial arrangement of shapes, and problems regarding comparisons between shapes. In this work, we are not interested in the spatial relational understanding capabilities of the probed networks. Therefore, we concentrate only on the problems requiring shapes comparisons.

Kim et al. [10] probed simple CNNs upon all the problems in the SVRT dataset, and they found some of these problems particularly straining for their simple setup. Following the findings by Kim et al., the same-different problems number 1, 5, 20, 21, are among the most difficult to solve for a convolutional neural network. For this reason, in this work, we concentrate in-depth on these four problems.

In particular, to solve the enumerated problems we are requested to handle the following challenges:

– **Problem 1 (P.1)**: Detecting the very same shapes, randomly placed in the image, but having the same rotation and scale.

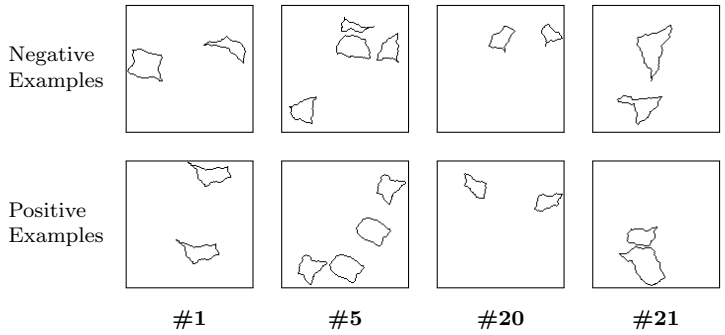

Fig. 1: Positive and negative examples from the four considered SVRT problems.

- **Problem 5 (P.5)**: Detecting two pairs of identical shapes, randomly placed in the image. The two images inside every pair have the same orientation and scale.
- **Problem 20 (P.20)**: Detecting the same shape, translated and flipped along a randomly chosen axis.
- **Problem 21 (P.21)**: Detecting the same shape, randomly translated, rotated, and scaled.

All the images from the proposed problems contain two shapes except problem 5, where images contain two pairs of shapes.

In Figure 1 we report some positive and negative examples from each one of the above-listed problems.

In all of these benchmarks, shapes are well separated and never overlapping. Brute-force memorization cannot solve the task since the shape of the closed curves is randomly generated. Despite being a quite straightforward dataset, built of simple shapes and with no color information, SVRT can help to highlight some intrinsic limitations of current neural network models, as far as abstract reasoning on shapes is concerned.

4 Method

This work is aimed at probing state-of-the-art architectures forming the basis of modern computer-vision models, and at measuring their abilities to intrinsically perform some abstract reasoning on images.

In our previous work [1], we probed the following state-of-the-art architectures for image classification: AlexNet [11], VGG-19 [14], three variants of the Resnet [6]: in order of increasing complexity ResNet-18, ResNet-34 and ResNet-101, and a recently introduced biologically inspired network called CorNet-S [12].

In this paper, our aim is trying to draw some better conclusions on the architectural factors that contribute to solving the same-different problems.

For this reason, we probe also two versions of the DenseNet architecture, DenseNet-121 and DenseNet-201, which implement non-residual skip connections, and we explored the Batch Normalized version of the VGG-19. Furthermore, we try to remove important architectural building-blocks from previously converging architectures to try to isolate the important architectural factors triggering the convergence. In this regard, we try ResNet-18 and ResNet-34 without residual connections, and 3 variations of the CorNet-S obtained by removing recurrent and/or residual connections.

Following, we describe in more detail the architectures that we will probe on the same-different problems.

VGGs VGG-19 contains a simple convolutional architecture comparable with the AlexNet structure, but it is significantly deeper. The original VGG-19, however, does not include in its convolutional modules a batch normalization layer. This could be quite an important detail for reaching network convergence and better stability, especially if the input image is non-normalized. For this reason, we experimented also with the *VGG-19-BN* architecture, which is the Batch Normalized version of the VGG-19. It simply includes a `BatchNorm` layer after each `Conv2D`, before the ReLU activation.

ResNets ResNets introduce residual connections. The skip connections used in the residual architecture force the model to learn incremental differences in the stored representations, refining the information multiple times until it can be used for the downstream task. The experiments on residual-connection removal from previously converging ResNet-18 and ResNet-34 (namely ResNet-18-WS and ResNet-34-WS, where WS = *Without Skip-connections*) can spot out to what extent the residual connections are a key detail for solving the *same-different* task.

DenseNets The novel experiments conducted on DenseNets can spot the differences between a residual network and a network based on generic skip connections. In fact, the DenseNets, differently from ResNets, introduce multiple non-residual skip connections moving lower-level information to each one of the higher-level layers.

CorNet-S Kar et al. [9] introduced this biologically-inspired network which evolves the ResNet architecture by introducing recurrent connections. CorNet-S is inspired by some experimental evidence reported by the authors on primates brain, claiming that the visual cortex could be comprised of recurrent connections. CorNet-S is composed of four blocks, mimicking different brain cortical areas involved with vision; each one of these four blocks contains a recurrent connection together with a residual skip connection so that ResNets are unrolled versions of CorNet-S.

Compared to ResNet-101, which has a comparable number of weights with CorNet-S, this model is also lighter to train since, differently from ResNets,

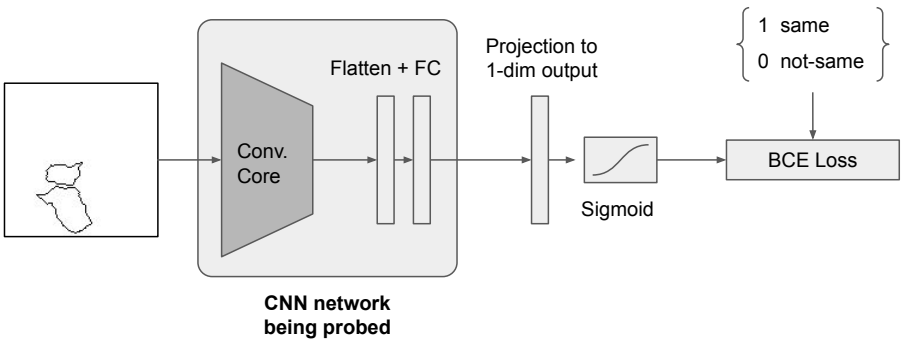

Fig. 2: Overview of the network for training on the same-different problems. The architecture of the network in the large light-gray box depends on the specific convolutional network being probed. It is usually composed of a core built of CNN layers plus final FC layers with ReLU activations outputting a fixed-sized vector. We remove the classification head, and instead, we linearly project the output to a single scalar value using a single FC layer. We then normalize this value in the range $[0, 1]$ with a sigmoid activation function before computing the Binary Cross-Entropy (BCE) loss.

recurrent connections make most of neurons weights shared among different timesteps.

The introduced architecture CorNet-S-WR, where WR = *Without Recurrent-connections*, is identical to the CorNet-S model but has recurrent connections removed. This modified network is aimed at spotting out what is the influence of recurrent connections as far as the same-different problems are concerned. Similarly, CorNet-S-WS removes the residual connections from the basic CorNet-S model. We try also to remove both the recurrent and the residual links, giving rise to the CorNet-S-WR-WS model.

Note that CorNet-S-WR, when unrolled, constitutes a very shallow network. To conserve the original depth, we stack in sequence the internal modules a number of times equal to the original timesteps proposed by the authors. In this way, CorNet-S-WR is effectively an unrolled version of CorNet-S, with non-shared weights among timesteps.

4.1 Training

All the previously described architectures come with a final classification head since they have been mostly used for the task of image classification. In our scenario, instead, the output must be a binary value indicating if the shapes in the figure correctly satisfy the rules of the specific SVRT problem or not.

For this reason, we replace the final classification head with a fully-connected layer outputting a single scalar value, normalized in the range $[0, 1]$ using a fi-

nal sigmoid activation function. The whole network is then trained end-to-end using a binary cross-entropy loss (Figure 2).

5 Experimental setup

We train the networks discussed in Section 4 on all the four same-different problems from the SVRT dataset. Concerning AlexNet, ResNets, DenseNets, and VGG-19, we use the models provided by the PyTorch framework. Instead, for CorNet-S we employ the publicly available implementation provided by the authors.

For each one of the four same-different benchmarks, we use 400k images for training, 100k for validation, and 100k for testing. Differently from our previous work where we reported the final accuracy values on the validation set, here we use a separate test set. All the generated images in the SVRT dataset have a size of 128×128 pixels.

Using 400k training examples could in principle bring to overfitting since the variability of synthetic datasets is often limited. Nevertheless, we claim that being the figures randomly generated in a 128×128 pixels grid, the probability of generating the same image twice is very low. Furthermore, humans can indeed learn the proposed problems using only a few samples. However, humans use a lot of pre-learned priors to solve these tasks, like the rotation/scale invariance, the concept of shape and closed shape or the mirroring invariance. Humans acquire this knowledge from the experience acquired on a multitude of other tasks, usually by performing transfer-learning. In this scenario, instead, this knowledge must be internally constructed from scratch, and this is achieved by using a relatively big and non-overlapping training set.

All the positive and negative examples are perfectly balanced in all the training, validation, and test sets. For all the probed models, we use SGD as the optimization algorithm, with a momentum of 0.9, weight decay of 1e-4, and a learning rate of 0.1. We do not use pre-trained weights if they are available. In fact, since these networks are trained on image classification from real-world images, the pre-trained weights cannot deal with the very different distributions of the synthetic SVRT dataset.

The input images are resized to the standard 224×224 for ResNets and DenseNets architectures. We perform more experimentation on the VGG-19 and AlexNets models since they are the ones demonstrating major strain. In particular, for these architectures, we try both to resize the input images to 224×224 and to keep the original 128×128 dimensions. Furthermore, since AlexNet is designed with no batch-normalization layers, we tried to normalize the input images by subtracting the mean and dividing by the standard deviation of the SVRT dataset.

We present two different experimental setups: in the first, we are interested in understanding if the different networks can converge, and with which strain, on the four same-different problems; in the second, instead, we want to tackle the generalization abilities of the trained models by testing their performance

using the test set of a different problem. Following, we in-depth describe these two experiments.

5.1 Experiment 1: Convergence

In this first experiment, we aim to understand if the explored networks can converge on the four same-different problems. Thus, in this setup, we train the various models, measuring their accuracy on the test set of the same same-different problem. In this experimental scenario, we are also interested in measuring what is the strain perceived by the network during the training phase. For this reason, we desire to capture fine-grained insights during the training phase, for understanding what is the effect of different architectures (or small modifications of the same) on the training curves. Reporting the training curves for all the explored networks is infeasible; however, we can extract some relevant information from the training curves and summarize them under the form of a simple index.

To this aim, similarly to [10], we extract from the training curves the point (expressed in epochs or fraction of epochs) in which the validation accuracy reaches 90%. We assume that the more the network is strained, the more examples it needs during the training phase to reach a good accuracy. From this moment on, we will refer to this particular point as the *convergence epoch* (CE). For performance reasons we validate the model every half epoch, so we can provide the CE with a resolution of 0.5 epochs. We think that this resolution is enough to capture substantial differences in the training curves.

Together with our measurements on deep convolutional networks, we also report the values as measured by [4, 21] on LeNet, GoogLeNet, and AdaBoost (using feature group 3). Table 1 summarizes the accuracy reached by all the architectures, over the test sets of the respective problems. For the architectures trained by us, we report also the measured CEs, only when the architecture converged.

5.1.1 Discussion

Table 1 shows that, among all the probed architectures, ResNets, DenseNets, and CorNet-S can learn all the four problems correctly and with a very few error rate, defeating all the state-of-the-art results previously reached with AdaBoost [4].

On the other hand, most of the configurations derived from AlexNet and VGG-19 architectures are unable to learn or are particularly strained. More in detail, almost all the VGG-19 configurations remain on the chance level accuracy of 50%, apart from the VGG-19-BN resized to 224x224. Nevertheless, this configuration can converge on only two out of four problems, and with accuracies far below the state-of-the-art reached with ResNets and DenseNets. This is the case even for the AlexNet with normalized input images.

Table 1: Accuracy values (%) measured on the test set of the probed architectures, for each of the four SVRT problems. Experiments reaching a perfect chance accuracy are reported in gray. The values from the first experiments are reported as they are from [4, 21]. They did not report any convergence information (CE is n.a.).

	Problem 1		**Problem 5**		**Problem 20**		**Problem 21**	
Model	Acc.	CE	Acc.	CE	Acc.	CE	Acc.	CE
LeNet [21]	57.0	n.a.	54.0	n.a.	55.0	n.a.	51.0	n.a.
GoogLeNet [21]	50.0	n.a.	50.0	n.a.	50.0	n.a.	51.0	n.a.
AdaBoost [4]	98.0	n.a.	87.0	n.a.	70.0	n.a.	50.0	n.a.
Human [4]	98.0	n.a.	90.0	n.a.	98.0	n.a.	83.0	n.a.
AlexNet	50.0	-	50.0	-	50.0	-	50.0	-
AlexNet 224x224	50.0	-	50.0	-	50.0	-	50.0	-
AlexNet norm.input	80.1	-	50.0	-	76.1	-	84.1	-
VGG-19	50.0	-	50.0	-	50.0	-	50.0	-
VGG-19 224x224	50.0	-	50.0	-	50.0	-	50.0	-
VGG-19-BN	50.0	-	50.0	-	50.0	-	50.0	-
VGG-19-BN 224x224	93.8	1.5	93.1	6.0	50.0	-	50.0	-
ResNet-18	99.2	0.5	**99.9**	2.5	95.5	2.0	96.2	17.5
ResNet-18-WS	98.9	0.5	99.5	2.0	95.7	1.0	96.7	8.5
ResNet-34	98.2	4.5	98.7	1.5	93.8	6.5	96.9	13.0
ResNet-34-WS	98.6	1.0	97.6	1.5	93.6	1.0	90.8	17.5
ResNet-101	99.1	3.5	96.0	3.5	**95.8**	4.0	91.1	20.5
CorNet-S	96.9	1.0	96.8	2.0	95.0	2.0	96.9	17.0
CorNet-S-WS	95.6	1.5	97.1	2.0	92.7	3.0	90.7	18.5
CorNet-S-WR	94.2	1.5	91.0	7.5	91.5	4.0	88.3	-
CorNet-S-WS-WR	93.5	1.5	92.7	8.0	91.3	7.5	86.5	-
DenseNet-121	**99.6**	1.0	98.2	2.5	94.2	1.5	95.1	7.0
DenseNet-201	99.5	0.5	99.3	1.5	94.3	1.5	**97.5**	17.0

By contrast, residual networks (ResNet-18, ResNet-34, ResNet-101, and CorNet-S) and DenseNets obtain the best performances on all the problems. Even the small ResNet-18 behaves well, both in terms of reached accuracies and CEs. Considering P.1, all the ResNets and the DenseNets reach almost perfect performance on the validation set, approaching 98-99% accuracy with pretty fast convergence. Only P.21 seems to cause some strain in the training process of the residual networks since more epochs are required for the models to converge.

Overall, residual and dense networks defeat humans on three of the four tasks.

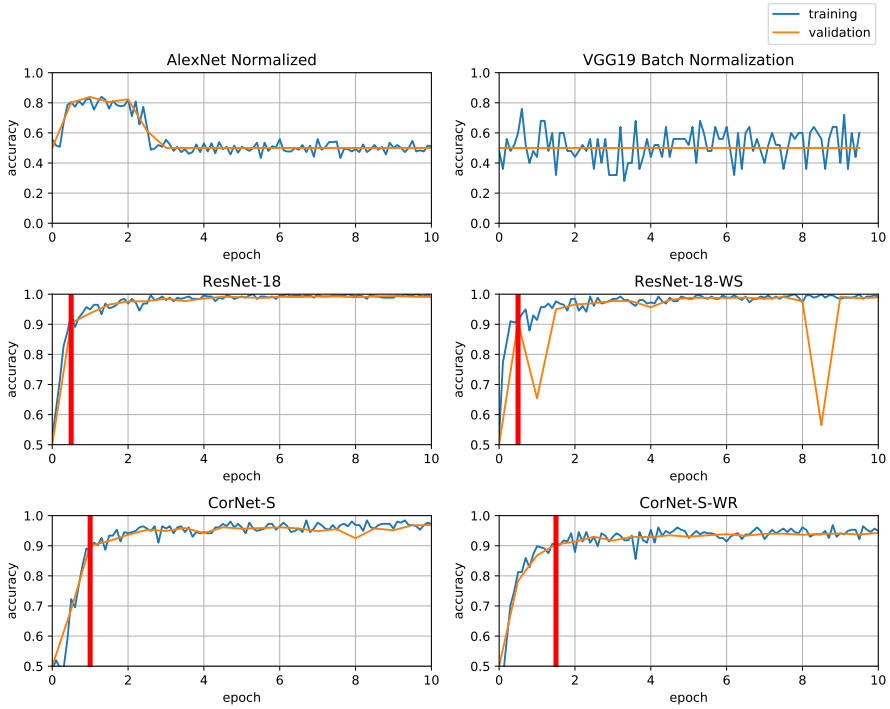

Fig. 3: Training curves for some of the networks trained on P.1. The red vertical line is placed in the correspondence of the convergence epoch (CE). Note that it is not present for the methods that do not reach at least 90% accuracy on the validation set.

The fact that both residual and dense networks obtain state-of-the-art results on these tasks suggests that there is no significant difference between the residual connections and the DenseNet-like skip connections in this scenario.

Furthermore, the ResNets without recurrent connections, (ResNet-18-WS and ResNet-34-WS) reach almost the same accuracies of the full ResNets, except in the P.21 where ResNet-34-WS obtains a lower accuracy with a higher CE, indicating a little strain with respect to the full ResNet-34. Overall, these results suggest that residual connections may be architectural building-blocks with little impact on the final test accuracy.

Concerning the CorNet-S network, the comparison among the CorNet-S-WS and CorNet-S-WR experiments shows that the lack of recurrent connections has a stronger impact than the lack of the residual ones. This in part confirms the previously formulated hypothesis about residual connections and indicates that recurrent connections can be important architectural details in CorNet-S. Without recurrent connections, CorNet-S cannot reach 90% validation accuracy on P.21, leaving the CE for this experiment uncharted.

Furthermore, the CorNet-S-WS-WR experiment, which lacks both residual and recurrent connections, reaches accuracies on the test set perfectly comparable with the CorNet-S-WR network, suggesting that the removal of residual connections has not a strong impact on the overall accuracies.

We claim that recurrent connections, which implies sharing the weights among timesteps, help in regularizing and stabilizing the network, and they impact considerably on the overall test accuracy when removed.

Training curves on P.1 In Figure 3 we report also some detailed training insights for some of the architectures present in Table 1 and trained on P.1. The strain for AlexNet and VGG19 can be observed from the first two graphs on the top. In particular, VGG19 cannot learn, as accuracy remains on chance values both in the training and validation sets; AlexNet can reach around 80% accuracy on the validation set, although it is destabilized after few training epochs.

On the other hand, ResNet and CorNet-S can converge easily to almost perfect accuracy.

Note that the training and validation accuracy curves are always perfect comparable, meaning that no overfit occurs in any of the reported scenarios.

Training curves on P.21 In Figure 4 we report the training insights for some of the architectures present in Table 1 and trained on P.21. The P.21 looks immediately more difficult by watching at the training curves. The convergence epochs are very noisy and visibly shifted to the right.

Differently from the P.1 training scenario, in this setup, especially in CorNet-S networks, slight overfitting becomes visible.

5.2 Experiment 2: Generalization

Description In this scenario, we test the generalization abilities of the converged models by measuring their performance on the test set of other problems.

We probe the models trained on P.21 and P.1. In particular, P.21 is a problem that should force the models to learn most of the required invariances needed to solve all the other problems (translation, scale, rotation). It is interesting to understand if the networks trained on P.21 are also able to perform well on P.1 and P.5 and to measure their generalization abilities to shapes mirroring (P.20).

On the other hand, P.1 only requires the networks to learn translation invariance; therefore, it is interesting to understand how well networks trained on this task can deal with rotation, scale, or mirroring invariance (P.20, P.21). It is also worth trying the P.1 generalization abilities to multiple shapes (P.5).

Table 2 reports the accuracies for the most promising models trained on P.1 on the test sets of P.5, P.20, and P.21. Instead, Table 3 provides the accuracies for the most prominent networks trained on P.21 on the test sets of P.1, P.5, P.20.

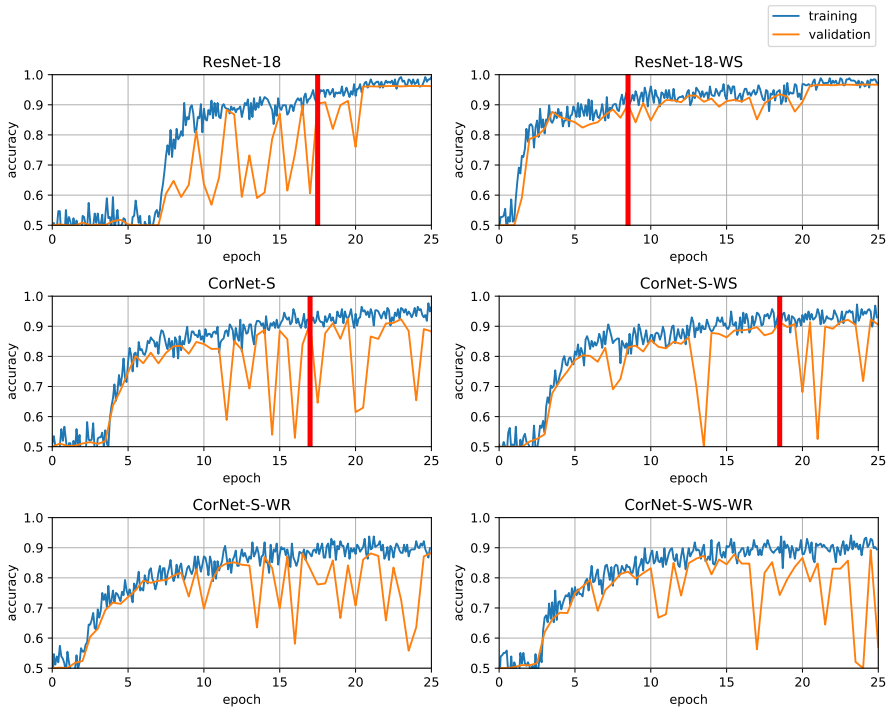

Fig. 4: Training curves for some of the networks trained on P.21. The red vertical line is placed in the correspondence of the convergence epoch (CE). Note that it is not present for the methods that do not reach at least 90% accuracy on the validation set.

5.2.1 Discussion: Training on P.1 and Testing on the others

Looking at Table 2 it turns out that none of the models obtaining almost-perfect test accuracy on P.1 can understand P.21. This is reasonable since P.21 requires invariance to rotation and scale, which are difficult concepts to acquire for an architecture trained solely on P.1.

On the other hand, ResNet-34 can generalize quite well to P.5. This is a remarkable result: P.5 requires the models to understand that objects should be clustered into two pairs of possibly identical shapes, and it is not trivial to deduce if both pairs match when the knowledge is constructed only from single pairs figures.

The architectural changes made to ResNets and CorNet-S have a visible impact on this generalization scenario: these networks experience a loss in accuracy when recurrent and residual connections are removed. For example, when testing on P.5, ResNet-34 without residual connections (ResNet-34-WS) loses around 10% with respect to the basic architecture, while CorNet-S-WS and CorNet-S-WR lose 13% when compared to CorNet-S. The lack of both

Table 2: Accuracy values (%) measured on the probed architectures, by training on P.1 and testing on the test sets of the other three problems.

Model	**Test P.5**	**Test P.20**	**Test P.21**
ResNet-18	56.5	55.6	51.6
ResNet-18-WS	56.4	58.4	51.2
ResNet-34	**84.4**	61.6	51.5
ResNet-34-WS	75.4	61.3	51.5
CorNet-S	73.6	**78.7**	52.0
CorNet-S-WS	64.6	76.8	51.7
CorNet-S-WR	63.9	71.3	**52.5**
CorNet-S-WS-WR	60.7	76.2	52.4
DenseNet-121	58.8	55.3	51.2
DenseNet-201	56.2	54.5	51.3

recurrent and residual connections in this scenario brings to a loss in accuracy by as much as 18%. ResNet-18 seems to be an outlier to this trend: on P.20, it obtains a better accuracy when the residual connections are removed.

On P.20, the higher accuracy is reached by CorNet-S. Although approaching a test accuracy far below the optimal one, the clear deviation from chance accuracy suggests that this architecture can partially understand flipped shapes.

Remarkably, both the DenseNet networks cannot generalize very well to any of the three test problems.

5.2.2 Discussion: Training on P.21 and Testing on the others

Table 3 shows how the models trained on P.21 can understand also P.1. This is expected since P.1 is a subset of P.21 that does not deal with scales and orientations of the shapes.

A remarkable result can be observed on the networks tested on P.20. In fact, the great part of the networks trained on P.21 can solve this problem almost perfectly, although P.21 does not carry the concept of shape mirroring. In particular, DenseNet-201 can reach state-of-the-art results on the generalization test on both P.1 and P.20.

In this generalization scenario, CorNet-S and ResNets suffer from the removal of residual and recurrent connections only when addressing P.1 and P.20, reaching the minimum accuracy on these problems with the CorNet-S-WS-WR, where both the residual and recurrent links are missing. ResNet-18-WS defines an exception to this trend since it can perform better than the full ResNet-18 on these two problems.

However, there is an interesting trend when comparing P.1 or P.20, with P.5. In fact, if we concentrate on the various CorNet-S versions, we notice that there is a decreasing accuracy trend on P.1 and P.20 when the recurrent and

Table 3: Accuracy values measured on the probed architectures, by training on P.21 and testing on the test sets of the other three problems.

Model	**Test P.1**	**Test P.5**	**Test P.20**
ResNet-18	97.9	54.2	96.0
ResNet-18-WS	98.3	53.3	96.6
ResNet-34	98.3	59.4	96.6
ResNet-34-WS	94.2	**63.4**	91.7
CorNet-S	98.6	54.2	97.0
CorNet-S-WS	95.6	59.1	91.7
CorNet-S-WR	92.4	61.4	89.9
CorNet-S-WS-WR	91.7	62.4	87.9
DenseNet-121	96.9	55.7	95.1
DenseNet-201	**98.9**	50.8	**97.4**

skip connections are gradually removed. Instead, an increasing trend is visible for P.5, although the absolute values for P.5 remain very low. The same thing happens for ResNets and DenseNets.

The low absolute accuracy values obtained in P.5 suggests that it is not sufficient to be invariant to rotation/scale/translation to understand this problem. P.5, as explained earlier, obeys to different logical rules with respect to the other test problems, and it is therefore difficult to zero-shot generalize on it.

On the other hand, we can notice very different results on P.5 with respect to Table 2. In fact, the models trained on P.21 seems unable to generalize well on P.5. This happens since the distinctive features of P.21 and P.5 problems are very different: not only P.5 is more similar to P.1 as they both lack of scale and orientation differences between shapes, but P.21 also carries a single pair of shapes.

In general, zero-shot generalization to P.5 is difficult since usually it is needed to observe few samples of the target problem to understand the new rules (e.g.: "shapes should match in both the two pairs").

6 Conclusions

In this work, we analyzed to what extent very-deep convolutional neural networks can deal with the *same-different* challenging tasks.

In particular, we stuck to the work by Kim et al. [10]. They showed how the same-different problems strain simple feed-forward convolutional neural networks. We extended their study to very-deep state-of-the-art neural networks for image classification.

Considering the SVRT visual challenge, our results show that ResNets, DenseNets, and CorNet-S (a biologically-inspired architecture similar to ResNet architecture) can correctly understand the challenging images and generalize to never seen shapes. We found that all these architectures found P.21 the harder to learn.

By contrast, older models such as AlexNet and VGG are particularly strained by the proposed problems. Their accuracy often settles to pure chance. Some improvements can be observed only when the architecture is augmented with batch-normalization layers, or when the input images are carefully resized and normalized. However, even in these cases, AlexNet and VGG have strong convergence problems.

In our previous work [1], we hypothesized that the residual connections in the ResNet architecture could have been important key points for the successful convergence of these models. In this paper, we conducted additional experiments that demonstrated how residual connections are not the only significant architectural details for successfully solving the same-different task, as far as convergence is addressed. In this case, we showed that there was not a substantial difference between ResNet-like and DenseNet-like skip connections. Furthermore, recurrent connections in CorNet-S seem to have a non-negligible impact on the final test accuracy.

Also, it seems that residual and recurrent connections have important roles in the zero-shot generalization to similar problems: architectures trained on the simpler P.1 seem to obtain lower accuracies on the test problems when residual and/or recurrent connections are gradually removed. In this scenario, DenseNets are not able to generalize. Instead, architectures trained on the difficult P.21 have different trends on different test problems when residual and/or recurrent connections are removed. In this scenario, instead, DenseNets can generalize very well except than in P.5.

In the end, we showed that recent convolutional networks for image classification can reach state-of-the-art results on all these challenging same-different problems from the SVRT dataset. Also, we found that they can reach superhuman performances on P.1, P.5, and P.21.

We think that the development of the abstract and relational abilities of neural networks is an important leap towards achieving some interesting new tasks, such as the discovery of particular models in cultural heritage, or the search for models of aesthetic judgment in images and even in music.

An important future step would be to consider more complex images collecting real-world shapes, possibly seen from different perspective angles. We also think that disentangling perception from reasoning using transfer-learning approaches would be the key to use fewer training examples and to possibly reach higher generalization abilities.

References

1. Blinded: Blinded. In: 2019 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 0–0. IEEE (2019)

2. Carrara, F., Esuli, A., Fagni, T., Falchi, F., Fernández, A.M.: Picture it in your mind: Generating high level visual representations from textual descriptions. Information Retrieval Journal **21**(2-3), 208–229 (2018)
3. Carrara, F., Falchi, F., Caldelli, R., Amato, G., Fumarola, R., Becarelli, R.: Detecting adversarial example attacks to deep neural networks. In: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, pp. 1–7 (2017)
4. Fleuret, F., Li, T., Dubout, C., Wampler, E.K., Yantis, S., Geman, D.: Comparing machines and humans on a visual categorization test. Proceedings of the National Academy of Sciences **108**(43), 17621–17625 (2011)
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
7. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2901–2910 (2017)
8. Johnson, J., Hariharan, B., Van Der Maaten, L., Hoffman, J., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Inferring and executing programs for visual reasoning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2989–2998 (2017)
9. Kar, K., Kubilius, J., Schmidt, K., Issa, E.B., DiCarlo, J.J.: Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. Nature neuroscience **22**(6), 974–983 (2019)
10. Kim, J., Ricci, M., Serre, T.: Not-so-CLEVR: Visual relations strain feedforward neural networks (2018). URL https://openreview.net/forum?id=HymuJz-A-
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
12. Kubilius, J., Schrimpf, M., Nayebi, A., Bear, D., Yamins, D.L., DiCarlo, J.J.: Cornet: Modeling the neural mechanisms of core object recognition. BioRxiv p. 408385 (2018)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
14. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian conference on pattern recognition (ACPR), pp. 730–734. IEEE (2015)
15. Mascharka, D., Tran, P., Soklaski, R., Majumdar, A.: Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4942–4950 (2018)
16. Messina, N., Amato, G., Carrara, F., Falchi, F., Gennaro, C.: Learning relationship-aware visual features. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 486–501 (2019)
17. Messina, N., Amato, G., Carrara, F., Falchi, F., Gennaro, C.: Learning visual features for relational cbir. International Journal of Multimedia Information Retrieval pp. 1–12 (2019)
18. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
19. Santoro, A., Hill, F., Barrett, D., Morcos, A., Lillicrap, T.: Measuring abstract reasoning in neural networks. In: International Conference on Machine Learning, pp. 4477–4486 (2018)
20. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: Advances in neural information processing systems, pp. 4967–4976 (2017)
21. Stabinger, S., Rodríguez-Sánchez, A., Piater, J.: 25 years of cnns: Can we compare to human abstraction capabilities? In: International Conference on Artificial Neural Networks, pp. 380–387. Springer (2016)

22. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
23. Vadicamo, L., Carrara, F., Cimino, A., Cresci, S., Dell'Orletta, F., Falchi, F., Tesconi, M.: Cross-media learning for image sentiment analysis in the wild. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 308–317 (2017)
24. Yang, G.R., Ganichev, I., Wang, X.J., Shlens, J., Sussillo, D.: A dataset and architecture for visual reasoning with a working memory. In: European Conference on Computer Vision, pp. 729–745. Springer (2018)
25. Zhang, C., Gao, F., Jia, B., Zhu, Y., Zhu, S.C.: Raven: A dataset for relational and analogical visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5317–5327 (2019)