Do deep neural networks generate visual categories like humans? The case of flower classification

Abstract

Deep learning algorithms successfully perform image classification with high accuracy. While the abilities of such networks approach human performance, the principles which underlie this success are not completely clear. Specifically, in comparison to human image-classification, an open question is whether the same success humans and machines share in *function* also rely on similar *mechanisms*. Here we report on the assembly of a novel dataset of labeled flower images. Flower classification bares cultural, and until recently also medicinal and agricultural significance, and humans are able to accurately cluster together flowers belonging to the same class. We trained a tabula-rasa network on the flower dataset and report a testing accuracy of 91%. Testing the potency of transfer learning, we used a different network, pre-trained on the ImageNet dataset but agnostic to the flower dataset, as a feature extractor and trained a multiclass SVM classifier on the activations of the last fully-connected layer of the network. With slightly diminished performances, the SVM classifier still classified the flower dataset with high accuracy (79%). Finally, applying unsupervised learning, we clustered the flower-images using their extracted features and compared these "machine" clusters to the human clusters, namely the flower classes. We find that the two data-partitioning, as implied by the two clustering methods, are to a large extent disparate. The similarity between the machine-induced portioning and the human categorization is equivalent to the similarity with a random reshuffling of more than half of the data. This result is consistent with the hypothesis that while resembling in function, deep-networks and humans do not in fact share similar underling mechanisms in image categorization. We discuss limitations, implications, and further extensions.

Introduction

The flowering plants, Magnoliophyta[1], are the most diverse group of land plants, with approximately 369,000 known species[2]. Plant taxonomy, specifically known as Angiosperm* for the system of flower classification, is the process of matching a specimen plant to a known taxon. Due to the importance of flower classification to human society, e.g. for agricultural and medicinal purposes, systematic classification methods and taxonomical nomenclature were early to develop by the scientific community[3,4]. Computational methods for flower classification were also early to develop, utilizing hand-crafted features (most notably Fisher's Iris[5], see also[6]). This classic approach of machine-learning is useful but difficult to scale, since acquiring the values for the crafted feature (e.g. measuring petal length) is costly (see however more data-driven approaches[7,8], though these approaches still define the classification features manually). In contrast, an automated procedure for flower classification which relies solely on image is a more scalable solution.

The problem of classifying an image showing a flower, to the category of flower shown in the image is not different from other problems of image classification (see for example problems of face recognition[9] and broad semantic categories[10]). Nevertheless, a few challenges that are noteworthy for the task are (Fig. 1) the variability in image configuration which results from the photography angle, the multiple morphologies characterizing single species, photography distance from the classification target and the abundance of semantic categories which should be grouped into a single classification category (a single, two, a few, and a field of poppies, should all be classified as a poppy). A robust classification solution should be illumination-, angle- and distance- (size) invariant and classify flower images correctly in cases of partial occlusion. A prominent candidate to solve such problem is deep learning, which in other domains was shown to demonstrate outstanding performances in solving classification problems[9,10].

The following is composed of two main sections. In the first section, we ask whether deep learning methods would be equally capable, as demonstrated in other domains, in classifying flower-images. To this end, we report here on the assembly of a novel

---

* Angiosperm, from the Ancient Greek αγγείον, angeíon (bottle, vessel) and σπέρμα, (seed), was coined in the form Angiospermae by the early botanist Paul Hermann in 1690

dataset of labeled flower images (for comprehensive details of the dataset, see supplementary table 1, also Fig. 2 and 3 bottom). In addition, to optimize the learning process a few learning strategies are implemented, analyzed and compared. Specifically: (1) We ask whether the classification procedure may gain accuracy by sampling *down* input images and what are the consequences of increasing image resolution when learning a limited dataset; (2) We compare the performance and generalizability of two learning strategies, first training a *tabula-rasa* network on the relatively small but ultimately relevant dataset of flowers; and second, training a network on a very large but less relevant dataset (imageNet[11,12]) and then using the network as a feature extractor for classification of the flower dataset, i.e. the potency of transfer learning[13,14] for the respective datasets.

In the second section, we use the trained networks in an unsupervised-learning paradigm and ask whether the flower categories (namely the different species) as defined by humans, naturally emerge from the activation patterns used to classify the flowers by the deep-networks. The motivation for such comparison is the question of whether human and machine categorization share some underlying principles. Both humans and deep-network are, after proper training, very capable of classifying images. An active field of research, both in computer science and neuroscience, is whether these comparable *functions* also rely on similar underlying computational or architectural *mechanisms*. One line of reasoning, regarding human and machine classification competence, argues that if a complex optimization task is well accomplished by two independent agents the resulting solutions should share some characteristics of their implementation. In contrast, an opposing hypothesis may be that the two classifying entities may reach similar performance while utilizing very different approaches. In these lines, here we ask whether the deep networks, which learn to classify flowers according to human categorization, learn and utilize some representation of the same features that humans use for classification by comparing machine and human clustering patterns in the flower dataset.

Figure 1. Classification challenges, demonstrated by poppy (Papaver subpiriforme) images from the flower-dataset: (**A**) varying angles, (**B**) different morphologies, (**C**) changing image percentage (object size) and (**D**) varying number of objects, all should be classified to the same class.

Results

Data

A novel dataset of images depicting flowers of different species at their natural habitat was assembled and labeled by a human expert (see Materials and Methods). The dataset included 5,470 labeled images manually classified into 58 categories (species) (Fig 2, supplementary table 1). All images include at least one but often several flowers (see Materials and Methods). The median number of images per category is 76 (min=7, max=277).

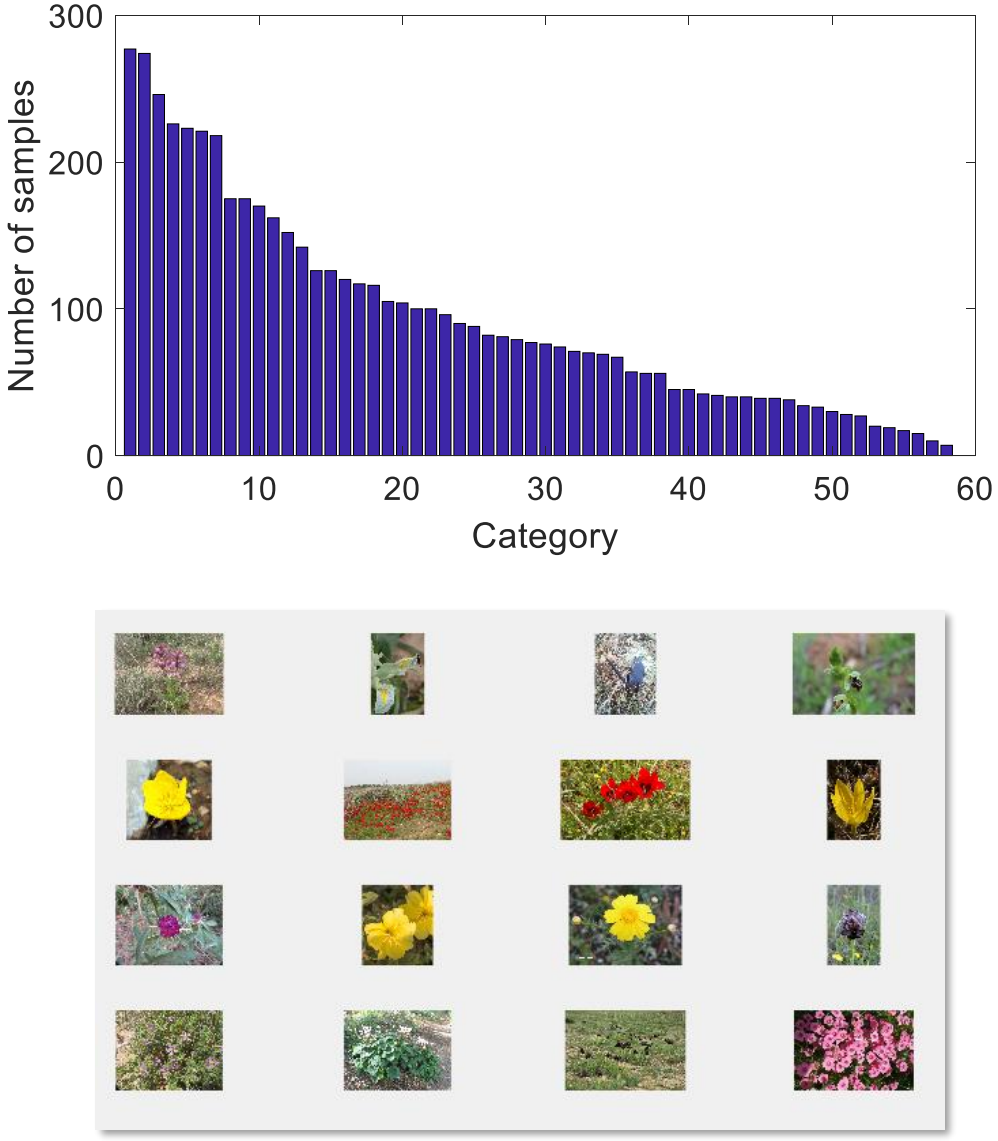

Figure 2 – The novel flower dataset: **Top** - samples per label: different classes have different number of images. **Bottom** - example images: the original images are of different sizes, and different backgrounds.

Flower classification using deep convolutional neural networks

Each of the 58 categories was partitioned into training and testing datasets. 80% of the images of each category were randomly chosen to the training dataset and the remaining 20% of the images were considered as a testing dataset.

A feed-forward convolutional neural network with no prior knowledge (i.e. initialized with random weights) was trained to classify the images in the training dataset into their true labels (see Materials and Methods). The accuracy of the trained network for a given dataset is estimated by the proportion of correctly identified images. Chance

level accuracy on the entire dataset is 1.7% using uniform prediction, 2.7% using proportional prediction, and 5.1% using most-frequent prediction (see Materials and Methods). At the completion of the training, the accuracy on the testing and training dataset using optimal initial-resolution (see below) was 90.63% and 89.62% respectively.

The effect of changing initial image size

As a processing step to both the training and testing phases, all images were resized to a rectangular uniform size (see Materials and Methods). To test the effect of the initial size of the images on the network's accuracy, the training-testing procedure was repeated several times while changing the size of the input layer (see table 1). We note that as the original size of the images was on the order of mega-pixels, each of the initial-resolutions tested constituted a substantial down-sampling. The results of this procedure, as shown in table 1 (see also Fig. 3), suggest that a 14X14px initial-resolution crosses a lower-threshold after which the effectiveness of learning is low, and the network is not able to achieve a good accuracy even on the training dataset (67%). Similarly, a much higher initial resolution of 112X112px is also ineffective in learning (45% accuracy on the training dataset). For a smaller initial size of 56X56 pixels, the network is effective in learning the training dataset (90% accuracy) but poorly generalizes over the testing dataset (44%). Finally, an optimal initial resolution, per the tested conditions, is achieved using an initial-resolution of 28X28 pixels, classifying correctly 90% and 91% of the training and testing datasets respectively.

Initial image size	Accuracy training (%)	Accuracy testing (%)
14 X 14	67.66	43.54
28 X 28	**89.62**	**90.63**
56 X 56	90.42	44.45
112 X 112	45.39	27.96

Table 1: the effect of initial image size on the network's classification accuracy. An optimal initial resolution for the tested conditions is 28X28px (highlighted in bold).

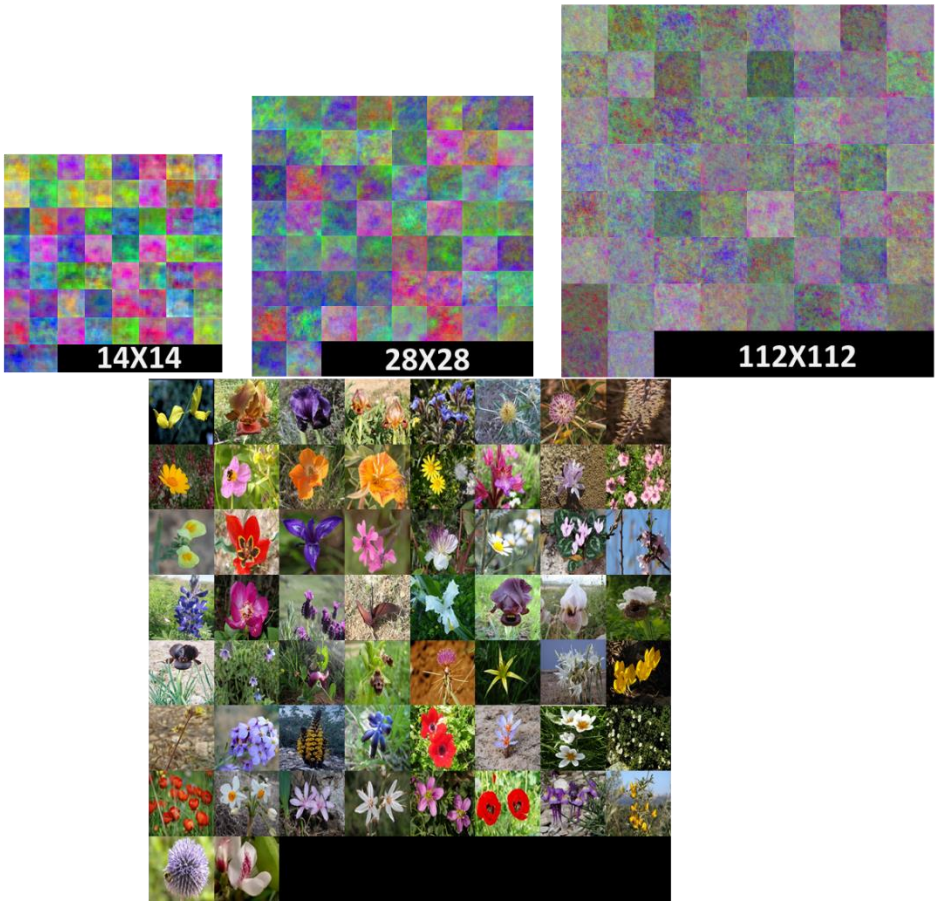

Figure 3. Visualization of the trained networks. Top – learned feature visualization with deep dream[15]. For each label of each trained network a strongly activating image is synthesized. From left to right, the images correspond to the same network trained with an initial image size of 14X14 pixels (**left**), 28X28 pixels (**middle**), and 112X112 pixels (**right**). **Bottom** – representative image of each label.

Using a pre-trained CNN as a feature extractor

An alternative method to training a tabula-rasa neural network, a costly procedure[16,17], is to use a pre-trained network as a feature extractor and then classify the images with more "classic" techniques based on these extracted features. Here, we use the 23-layer deep AlexNet[10], pre-trained on the 1.2 Million images and 1000-categories of ImageNet[11] dataset. As a feature extractor, we choose the last fully-connected layer of the network, which comprises of 4096 neurons. Similar to the previous section, the dataset was split into training and testing datasets by choosing at random 80% of the images per each label for the training dataset and the remaining 20% for the testing dataset. Per the classification learning, for each of the images in the training dataset, we calculate a feature-vector: the resulting activation of all the neurons in the last fully-connected layer when the images are given as input to the network. We then use

the feature-vectors of all the images to train a multiclass SVM classifier with K (number of classes) binary learners, in a one-vs-all design[18]. To test the classifier, we perform the same procedure - calculate the feature-vectors of all the images in the testing dataset and then use the trained classifier to classify them. Using this method, the SVM classifier successfully classifies 79% of the testing dataset. To examine the errors that the classifier performs, its confusion matrix over the testing dataset is calculated (Fig 3) but does not reveal any systematic errors. Overall, while the classification is less accurate than the network trained tabula-rasa, the performance of the SVM classifier is comparable to that of the network. This suggests that the features extracted by the network, when trained on an independent dataset, are indeed relevant for the classification of the current, previously unobserved, dataset.

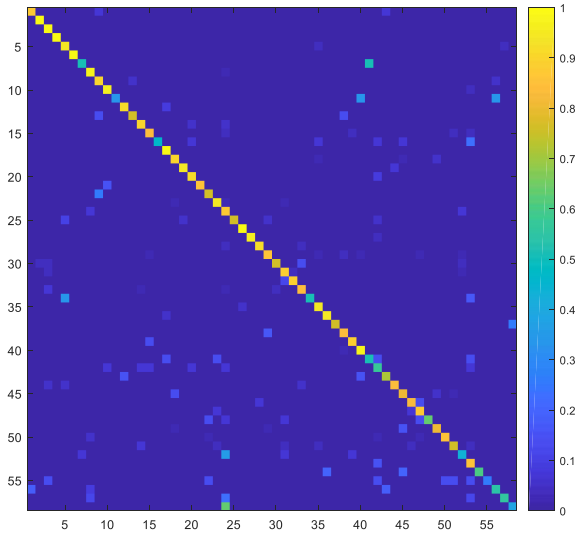

Figure 4 – SVM classifier confusion matrix. The classifier achieves relatively high accuracy and does not display any systematic errors.

Comparing human and machine natural clustering solutions

To compare the natural clustering of the full dataset, we compare the human categorization (i.e. the botanical categories as classified by the human expert) with clustering solutions of the image features extracted by the network. As an input for the clustering of the machine-extracted features, we considered the activations of all images in the last fully-connected layer of AlexNet, trained on the ImageNet dataset, when the images are flowed through the network. The resulting matrix has 5,470 columns (one per image) and 4,096 rows (the resulting activation after passing the

image through the network in each of the neurons of the last fully connected layer). This matrix is then clustered with an input of the true number of clusters (58, species of flowers) by unsupervised techniques, i.e. without an additional input of the labels of each image. Specifically, we use the k-means algorithm[19] applying several distance matrices (additional clustering methods were applied but are not discussed here, see Fig. 6).

Comparing the human and machine clustering solutions, a first global property which seem reserved in both solutions is a comparable distribution of clusters sizes (Fig 5). Next, we ask whether the partitioning of the dataset by the human and machine clusters is comparable. Since the unsupervised solution is agnostic with regards to the original categories there is no notion of identity to the clusters (there is no "poppy cluster"). Hence, the question of similarity between the clustering solutions is translated to comparison of the different portioning of the data, i.e. a normalized estimation of the probability that two flowers which shared a category in one of the solutions will also share the same category implied by the other solution. To estimate this probability, we use the adjusted rand index (ARI)[20–22]. Briefly, ARI is a symmetric index which indicates the similarity between two partitioning of a given dataset. The ARI quantify the probability that two instances that belong to the same cluster in one partitioning method, also belong to the same partition in the second partitioning, accounting for false positives and false negatives (instances that were not clustered together in one of the methods but are clustered together in the second) and chance level. ARI values range between 0 (the two data partitioning do not agree on any pair of points)* to 1 (indicating that the two partitioning are identical).

Furthermore, to estimate the quality of the match between the two clustering solutions, we randomly reshuffle an increasing proportion of the data and calculate for each of the reshufflings the ARI when compared with the original clusters. The resulting ARI for the unsupervised clustered activations and the original clustering is 0.225 which is equivalent to the ARI resulting from reshuffling 53% of the data (Fig. 6).

---

* The range $0 \leq ARI \leq 1$ is stated for simplicity, note however that though this range holds for the Rand Index (RI), the Adjusted RI may yield negative values.

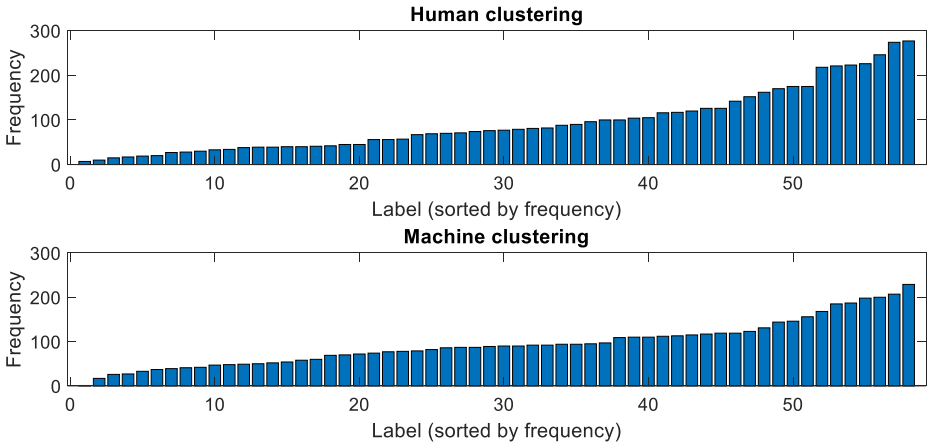

Figure 5 – human and machine cluster sizes. The two clustering solutions are similar in their cluster-size distribution.

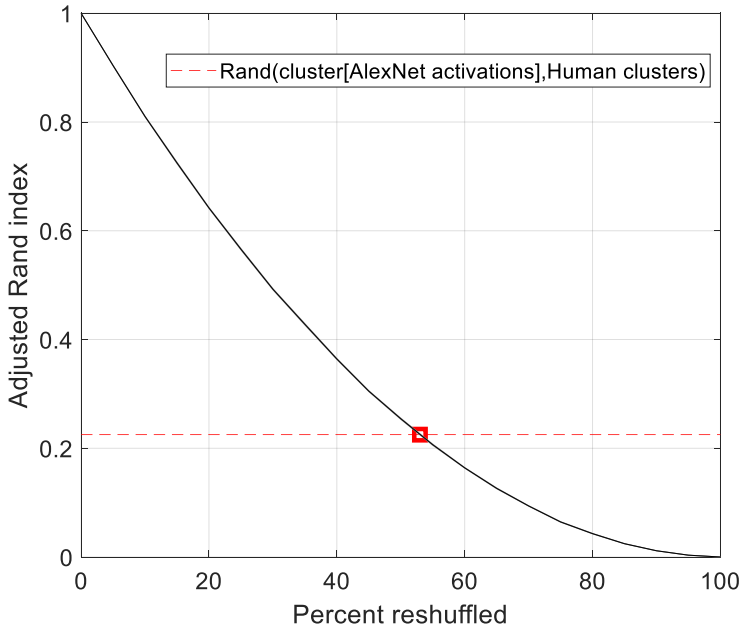

Figure 6 – Adjusted Rand Index (ARI) for the similarity between human clusters and machine unsupervised clusters. As shown in the figure, an ARI of 0.225 resulted in when applying for k-means clustering with Euclidean distances as a distance metrix. An ARI of 0.21 was achieved when applying cityblock as a distance metric. Each data point of the black curve was calculated by shuffling a certain percentage of the data and calculating the ARI with the unshuffled data. These reshuffling were implemented multiple times and standard errors are plotted but not clearly visible due to their very small magnitude (a typical error bar is of magnitude 0.0005). Additional clustering methods (clustering using general mixed models and hierarchical clustering) all yielded an ARI smaller than 0.05 (results not shown).

Discussion

Assembling a novel dataset of thousands of flower images labeled by the flower appearing in the image, we compared the effectiveness of training a tabula-rasa network with a pre-trained network used as a feature extractor for classification of novel data. We found that resizing the images to a too-low initial resolution (14X14 pixels) led to poor learning while utilizing to a too high resolution (56X56 pixels and 112X112 pixels) led to poor generalizability. In line with current well-established results of deep-learning models, we found that a tabula-rasa network trained with a relatively small initial resolution (28X28 pixels) successfully learns the training dataset and achieves robust generalization (91% accuracy on testing dataset). While yielding a lower accuracy, a different network (AlexNet[10]) trained on much larger dataset (imageNet[11,12]) still achieved high accuracy when used as a feature extractor in the classification of the flower dataset. Considering the activations of the last fully connected layer of the network as features extracted from each image, we compared the unsupervised clustering of these features with the data-portioning induced by human-defined categories. While the two partitioning of the data resembled some similarity, the similarity between the emerging unsupervised clusters and the human categories, were equivalent to comparison with a dataset in which more than half of the labels were reshuffled.

These findings do not offer strong evidence that humans and deep neural networks indeed use the same mechanisms to form visual categories. This result may be explained in several frameworks. First, while the unsupervised clustering method we chose, which we considered "natural", did not yield a good match between the machine and human portioning of the data, there exists, trivially, an equivalent method which will indeed generate an accurate mapping between the two partitioning of the data. In this sense, the chosen clustering algorithms are not unique and there is no a-priori justification to choose the ones described here and not another. Second, both the architecture and the training of the neural network determine the input for the unsupervised clustering. It is possible that for a different architecture or a different training protocol, the resemblance between the two clustering solutions would be significantly different. Future research may address this point by repeating the comparison between human and machine clustering solutions, using different architectures and training protocols.

Taking these points together the comparison between the two system of categories (machine and human) still, however, suggest some novel insight. For one, as a thought experiment, if the comparison procedure was repeated, but rather than comparing machine and human, we would have compared between *two humans* (i.e. comparing the independent clustering of the dataset into a predefined number of classes) we expect that the similarity between the human clustering-patterns to be substantially higher. While further investigation is still required, the results of the present study support the hypothesis that while both humans and machine may achieve very high accuracy in classification, this similarity in function indeed rely on different underlying mechanisms. An example for such differences in implementation is reflected in the use of different features of images for classification, which in turn, result different clustering solutions, a hypothesis which is consistent with the present findings.

Materials and Methods

Labeling

The dataset was manually labeled by an expert botanist* (see supplementary table 1). The expert verified that each image portrayed at least one flower. If more than flower was portrayed in the images, the expert verified that all flowers were of the same specimen. Background vegetation, as subjectively judged by the expert, was also allowed. The images were classified into categories such that each category (label) included images portraying flowers of the same specimen, as judged by the expert.

Image Preprocessing

Input images were resized, regardless of their original size, to one of the dimensions $kXkX3, k \in \{14, 28, 56, 112\}$. Resizing was implemented using bicubic interpolation[23] (the output pixel value is a weighted average of pixels in the nearest 4-by-4 neighborhood).

---

* Prof. Avi Shmida, to whom I'm greatly gratefull, is a faculty member at the Ecology, Evolution and Behavior department at the Institute of Life Sciences, The Hebrew University. Shmida has authored multiple books and papers. Amongst these is the guide to Israel's Flora (the dataset included only flowers naturally occurring in Israel). In that sense, Shmida may be considered as the "ground truth" for manual (i.e. "by eye", and not e.g. genetic) flower classification.

Training a simple convolutional neural network

The *input layer* of the network consisted of 3 color channels of the resized (see above) images. Next, a *convolutional* layer[24] with 20 filters (neurons) filtered the data, with additional *bias* layer, through a $[c, c]$ kernel with 0-padding, and a step size (stride) of $[s_C, s_C]$. Following, a *relu* layer[25] performed a threshold operation to each of its inputs such that negative inputs were equated to 0. The next *Max-pooling* layer[26] down sampled its input by considering the maximum value of each region with no overlap, i.e. stride size (s_{MP}) is equal to pool size (p). Then, a *fully connected* layer with a bias, transformed the *relu* layer output to a *softmax* layer[27] which applied a softmax function on its input. Finally, a classification layer assigned a label to the input image by calculating the cross-entropy loss:

$$loss = \sum_{i=1}^{N}\sum_{j=1}^{K} t_{ij} \ln y_{ij}$$

Where N is the total number of images, K the total number of flower categories, t_{ij} is an indicator that the i'th sample is a member of the j'th class, and y_{ij} is the output value of the softmax function for sample i and class j. The parameters chosen for this architecture are: $c = 5, f = 20, s_C = 1, s_{MP} = p = 2$.

The network was trained using stochastic gradient descent with momentum[28]. The initial weights of the network were drawn from a normal distribution with zero mean and standard deviation of σ With a momentum of m, constant learning rate of l and mini-batch size of b, the training was determined to terminate after a maximum of E epochs. An L2 regularization term (weight decay) was added to the loss function such that:

$$E_R(\theta) = E(\theta) + \lambda\Omega(w)$$

Where w is a weight vector, λ is the regularization coefficient and

$$\Omega(w) = \frac{1}{2}w^T w$$

The parameters chosen for the training were: $\sigma = 0.01, \lambda = 0.0005, m = 0.9, l = 0.01, b = 128, E = 30$.

Estimating chance-level accuracy

The number of images of each label are not identical (the dataset is not equipartioned, see Fig. 2). Hence, several statistics for chance-level accuracy are provided: (a) Uniform sampling - Each of the labels is predicted in the same probability, regardless of the number of images per label i.e. the probability p_i of predicting the label $i, i \in [1..N]$, where N is the number of classes, is $p_i = \frac{1}{N}$. (b) Proportional sampling - each of the labels is predicted in probability equal to the proportion of images associated with it ($p_i = \frac{|i|}{N}$, where $|i|$ is the number of images in the i'th class). (c) Most frequent - A constant prediction of the label representing the largest class of images.

References

1. Takhtajan, A. L. Outline of the Classification of Flowering Plants (Magnoliophyta). *Bot. Rev.* **46,**

2. Christenhusz, M. J. M. & Byng, J. W. The number of known plants species in the world and its annual increase. *Phytotaxa* **261,** 201–217 (2016).

3. Linnaeus, C. Hortus Cliffortianus. *Amsterdam* (1738).

4. Bentham, G. & Hooker, J. *Menispermaceae. Genera Plantarum.* (Reeve And Co.; London, 1867).

5. Fisher. The Iris Dataset. *UCI Machine Learning Repository* (1936).

6. Nilsback, M.-E. & Zisserman, A. Automated Flower Classification over a Large Number of Classes. in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing* 722–729 (IEEE, 2008).

7. Nilsback, M.-E. & Zisserman, A. A Visual Vocabulary for Flower Classification. in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)* **2,** 1447–1454 (IEEE).

8. Guru, D. S., Sharath Kumar, Y. H. & Manjunath, S. Textural features in flower classification. *Math. Comput. Model.* **54,** 1030–1036 (2011).

9. Parkhi, O. M., Vedaldi, A. & Zisserman, A. Deep Face Recognition.

10. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. 1097–1105 (2012).

11. Deng, J. D. J. *et al.* ImageNet: A large-scale hierarchical image database. *2009 IEEE Conf. Comput. Vis. Pattern Recognit.* (2009).

12. Russakovsky, O. *et al.* ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **115,** 211–252 (2015).

13. Shin, H.-C. *et al.* Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Trans. Med. Imaging* **35,** 1285–1298 (2016).

14. Oquab, M., Bottou, L., Laptev, I. & Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. in *IEEE Conference on Computer Vision and Pattern Recognition* (2014).

15. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T. & Clune, J. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. 3387–3395 (2016).

16. Glorot, X., conference, Y. B.-P. of the thirteenth international & 2010, undefined. Understanding the difficulty of training deep feedforward neural networks. *jmlr.org*

17. Dahl, G. E., Sainath, T. N. & Hinton, G. E. On the Importance of Initialization and Momentum in Deep Learning. in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013).

18. Fürnkranz, J. Round Robin Classification. Journal of Machine Learning Research **2,** (2002).

19. Arthur, D. & Vassilvitskii, S. K-Means++: the Advantages of Careful Seeding. in *Proc ACM-SIAM symposium on discrete algorithms.* (2007).

20. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2,** 193–218 (1985).

21. Xuan Vinh, N., Epps, J. & Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. Journal of Machine Learning Research **11,** (2010).

22. Santos, J. M. & Embrechts, M. On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification. in 175–184 (Springer, Berlin, Heidelberg, 2009).

23. Carlson, R. E. & Fritsch, F. N. An Algorithm for Monotone Piecewise Bicubic Interpolation. *SIAM J. Numer. Anal.* **26,** 230–238 (1989).

24. Le Cun, Y. *et al.* Handwritten Digit Recognition with a Back-Propagation Network. in *Advances in Neural Information Processing Systems (NIPS)* (1990).

25. Nair, V. & Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proc. 27th Int. Conf. Mach. Learn.* 807–814 (2010).

26. Nagi, J. *et al.* Max-pooling convolutional neural networks for vision-based hand gesture recognition. in *2011 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011* (2011).

27. Bishop, C. M. Pattern Recognition and Machine Learning. Pattern Recognition **4,** (2006).

28. P. Murphy, K. Machine Learning: A Probabilistic Perspective. *Mach. Learn. A Probabilistic Perspect.* (2011).

Acknowledgements

I thank Prof. Avi Shmida for collecting and labeling the flower dataset and supporting this research. I would also like to show my gratitude to Dean Foster and Prof. Tali Tishbi for useful discussions and guidance, although any technical or conceptual errors are solely my own.