Constellations: A novel dataset for studying iterative inference in humans and AI

Tarun Khajuria Kadi Tulver Taavi Luik Jaan Aru

Institute of Computer Science
University of Tartu, Estonia
{tarun.khajuria,kadi.tulver,taavi.luik,jaan.aru}@ut.ee

Abstract

Under complex viewing conditions, human perception relies on generating hypotheses and revising them in an iterative fashion. We developed novel visual stimuli to study such iterative inference in humans and AI. In these stimuli, called "constellations", all local information about the object has been removed and it can only be recognized when taking into account the global pattern. We here describe the dataset and demonstrate that humans indeed use an iterative process of generating hypotheses and refining them to solve these images. We also provide code that allows researchers to create their own constellation images. The constellation dataset allows researchers to develop sketching algorithms for guessing the hidden object. As such algorithms used by humans appear to be iterative in nature, this dataset will facilitate the study of iterative inference in minds and machines.

1. Introduction

In the past decade, artificial neural networks (ANNs), particularly convolutional neural networks (CNNs) have made major advances in reaching human level performance in certain visual recognition tasks. However, it is still clear that human visual capabilities go beyond these object detection and classification tasks where the ANN models excel.

Consider looking at the night sky to find star constellations while being an amateur at it. In this case, one cannot immediately see objects formed from stars, but can nevertheless generate many original hypotheses about what could be there ("a coffee machine", "a baby dolphin"). One is also able to test whether these hypotheses explain patterns of stars in the night sky or not in an iterative fashion. Eventually, one might discover a warrior (Orion) or the great bear (Ursa Major) as a solution. This process of going back and forth, revising the hypotheses until finding a solution has been called 'iterative inference' [64].

While traditional ANN solutions model relationships in the scene [18, 32, 55, 58], they do so in a single pass over the image, relying on a deep stack of convolution layers to model relationships between the image components. Such a bias prevents generating a rich understanding of the image as it does not allow for an iterative refinement of hypotheses about its content and meaning.

On the other hand, in human vision, understanding a scene has been shown to involve a much more recursive evaluation of the visual input [10, 35, 37, 53, 64] as different possible hypotheses or interpretations are considered. Such a process makes human vision more robust to noise, context changes and helps adapt to new scenarios [16, 53].

To investigate the full scope of iterative inference, tasks are required where the participant has the possibility to generate many hypotheses that are refined in an iterative fashion.

Here we develop a minimal experimental setting that would help us study iterative inference. We were inspired by the example brought above about the identification of constellations in the night sky. For finding constellations humans cannot directly identify the object but have to consider in their mind different ways in which the stars can be connected and evaluate which one of them relates to a familiar concept. This process has a generative component, imagining potential solutions and iteratively refining or changing the solutions according to the given constellation image. When it comes to ANNs we can think of these potential solutions as sketches of objects made by connecting the dots or just passing through the relevant dots in the image.

We synthetically generated a dataset of images with common objects hidden as constellations to promote the probing of iterative inference while controlling for various factors. This dataset complements the already existing datasets like Imagenet [9], COCO [46], Things Dataset [28], and Ecoset [52] as these previous datasets were not developed for this particular goal of studying the iterative process of inference. (However, in principle one could generate constellation images from any image dataset with the

Figure 1. Six examples of the constellation objects. Depicted are the original image, the dotted outline and the constellation image with low local signal. The constellation dataset allows researchers to train and evaluate sketching or other generative solutions for inferring the object hidden in dots.

code we provide). The images are made compatible for use in experiments for psychology, neuroscience, and AI. We illustrate how one can use this task to better understand the process of iterative inference in humans.

Our contributions are the following:

- we present a new dataset that allows one to study iterative inference of perceptual input
- we show data of how humans perform iterative inference on these constellation images
- we provide the code to generate the constellation images
- we show that the constellations can be used to study sketching in humans and machine vision algorithms

2. Background and related works

In this section, we will discuss the current state of Deep Learning architectures for vision and the motivation for developing this dataset.

2.1. Role of datasets in advancing computer vision

Methods and models for computer vision have evolved over time from hand-engineered algorithms and signal processing methods that extract useful features to the neural networks approach, with feed-forward neural networks achieving the state of the art in many tasks [39,43]. Datasets such as Imagenet played a huge role in the development and evaluation of various state-of-the-art ANNs like Alexnet [39], VGG [58] and ResNet [26] that pushed the boundaries of computer vision on the object recognition task. Going beyond image categorisation, datasets like COCO [46] proved to be highly influential in the generation of solutions for tasks like image segmentation, captioning, and scene description. Architectures like Mask R-CNN [25], Yolo [55], and DenseCap [32] are the results of this exploration.

2.2. Inspiration from natural vision

Deep Learning (DL) based vision solutions often fail to perform well on a few instances that are typically considered easy for humans [27]. One of the features of human vision that makes it robust against such failures is its ability to process the inputs in a recurrent manner [17, 56]. Research has also shown that recurrence is necessary for neural networks to account for the activation patterns in data recorded from human visual cortex [35, 37]. More importantly, recurrence added to these DL networks explains the trade-off between accuracy and speed as seen in human vision [59]. Due to these, it makes sense to make the architectures recurrent and in general add iterative processing to DL algorithms for certain vision tasks.

2.3. Iterative and recurrent processing in DL

Many attempts have been made to include the iterative capabilities of natural computations in DL architectures for vision. Recurrent capability was added to CNNs by adding a recurrent connection within each CNN layer [45]. CornetR [40] added local recurrent connections to the existing feedforward pipelines for object recognition by implementing biologically plausible unrolling.

Methods used in image restoration have increasingly

been using the principles of recurrent modelling to identify self-dependencies between different regions. Recurrent processing especially helps to model long-range self-dependencies [49] in images that the convolutions cannot easily model. Whereas architectures for Image denoising mainly make use of CNN-based networks [36, 50], recurrent connections are increasingly being added especially when dealing with more difficult real and blind noise [19, 61, 73]. DL architectures for super-resolution also follow the trend of using recurrent networks [60, 67]. For instance, a step forward from cascading upsampling based solutions [41, 42, 66], iterative upsampling [23, 24, 30] has become an important framework in the field.

Applications requiring inference between different parts of the image are increasingly developing using an iterative method. For object detection, [7] makes initial labels using CNN based methods, but models the interaction between the object labels using graph neural networks while iteratively correcting the labels. Fields like scene representation or scene decomposition increasingly make use of iterative DL methods. For example, MONet [3] is a model where the scene is recurrently decomposed into its constituent parts by learning to attend to the constituents object-wise while regenerating the scene. IODINE [20] makes parallel individual interacting cascading passes to model the constituent objects but refines the representation iteratively. Genesis [12] models the iteration explicitly between the constituents during the generative process of the scene re-composition. [51] introduces slot attention to model object-based representations in scene. This is an iterative method to bind CNN-based perceptual representations of objects in the scene to a fixed number of learnt representation variables called slots.

2.4. Datasets for visual reasoning

There are various other tasks intended to test the visual understanding capabilities of ANNs which may indirectly promote the use of iterative inference in DL. Corresponding datasets like visual question answering datasets [1, 29] make the system answer questions based on the image given. The complexity of relations that the visual system may be challenged to model in such datasets can be dependent on the complexity of the questions. Other datasets like the Multi-object dataset [34] with sets like CLEVR with stripes (based on [31]), or Multi-3d stripes, promote scene modelling tasks requiring multi-object modelling and masked generation of each object in the image. Such generation often requires both iterative and generative capabilities as many objects in the scene are hidden or overlapping [3, 20].

There are several other datasets such as the Pathfinder [48] and the cluttered ABC [38] for studying visual inference where the task cannot rely on local information. [38] shows that increasing the complexity of the vision task hinders learning in ANNs relying solely on low level information. [48] also introduces a task requiring networks to learn long range dependencies and introduce a horizontal gated recurrent units, that help in modelling such dependencies.

2.5. Sketching for visual reasoning

In a recent trend, the use of AI for creating sketches has gained momentum [13, 14, 57]. Even in earlier papers on drawing, DRAW [21] used iterative attention based modelling of images, using each iteration to refine and generate part of the image. Sketch RNN [22], introduced an RNN model to learn stroke based neural representations for sketches of common objects and could produce conditioned or unconditioned sketches. However, here the iterative use of networks is used to efficiently construct images. Iteration is now also used to iterate over while navigating a conceptual space to find a correct match for a conceptual description. Sketching solutions [14, 57] based on searching the possible space of solutions under the guidance of CLIP [54], a large image-text pre-trained model, have been very effective for guiding description based image generation. Other solutions [13] in the same category iteratively model a hierarchy of commands in the form of a sketch stroke to draw based on an image description. Sketch-based modelling is not limited to model a single object, as sketch-based representations can be used to model a multi-parts scene [68] or a scene with temporal dynamics in a video [8, 72].

Various methods also try to translate sketches into realistic images [6, 65]. A related visual inference problem is sketch-based image retrieval, where user inputs based on sketches are used to infer related original images. Many current solutions in this field use methods that align sketch and image representations through a CNN with various projections [15, 63]. Improvements are also being made on the reverse problem of generating a sketch based on an original image [44, 69]. The current solutions [2], however, are still highly dependent on reading low level original image features for the sketch generation.

3. Creating the constellation dataset

We used images of common objects from the Things Dataset [28] to create the constellation dataset. The Things dataset consists of 26,107 images of 1,854 objects, with each object having 12 or more exemplars.

The steps to generate the constellation images are illustrated in Figure 2 and are the following: 1) Generating outlines for the object from the original image; 2) Manual selection of best outline candidate, followed by manual editing in some cases; 3) Automated generation of dotted version and then constellation version of the image using the selected outline. In the next section, we discuss these steps in detail.

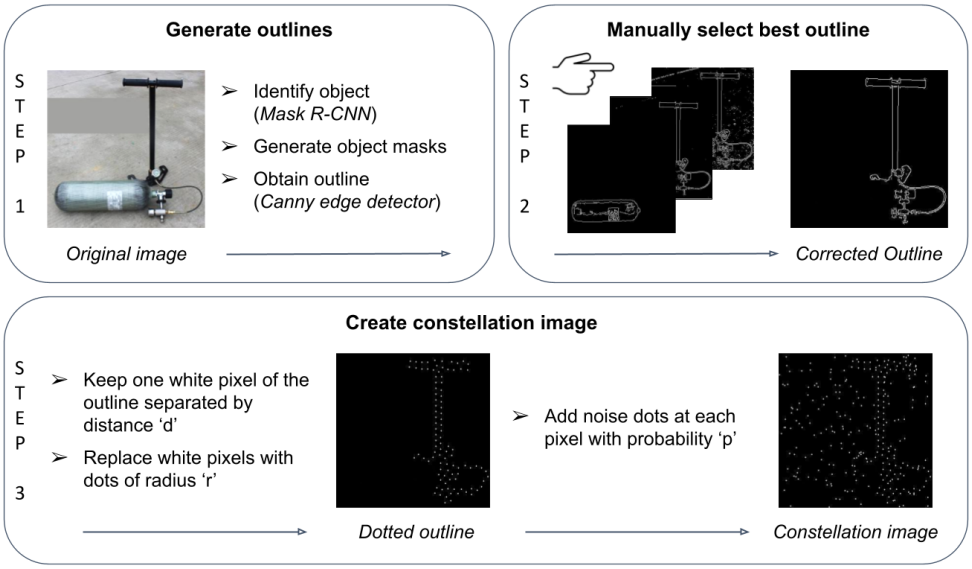

Figure 2. The image generation process with various automated and manual steps. The original image is first transformed into outlines using a serial application of image segmentation and then canny edge detector operation. The best outline is manually selected. Some corrections (additions or erasing some portions) are made manually to make sure the key feature of the object is visible in the outline. Finally, the automated pipeline is run on the final outline to generate various versions of dotted and constellation images for the experiment.

3.1. Automated generation of an outline

We first used multiple ways (described below) to obtain an outline for the object in the image.

First, we use Mask R-CNN [25] to identify the region of the image containing the object. We obtain the binary mask output from Mask R-CNN, indicating the pixels belonging to that object in the image. We multiply this binary mask with the original image to get the image with only the detected object. In particular, we use Mask R-CNN pre-trained on the COCO dataset [46], which contains many categories overlapping with the objects in the Things dataset. Still, the segmentation performance is sub-optimal on the missing categories. It is impossible to train the network on the Things dataset due to the non-availability of ground-truth object masks.

Second, to compensate for inconsistency from Mask R-CNN outputs, we repeated step 1 with multiple mask settings to obtain multiple masked images. We generated multiple masked versions by 1) capturing only the principal object, 2) capturing the first two prominent objects, 3) an unmasked full image. Having these multiple versions allows for a simple manual selection later, making the final outputs more appropriate.

Finally, we use the Canny edge detector [5] to obtain the outlines from these images. We use the canny edge detector with a threshold of (100,200) for all masked images. For the unmasked images, an additional blurring mask with a radius of 5 pixels is used. The image obtained after these steps can be seen in step 2 of Figure 2.

3.2. Manual selection and editing

We perform a round of manual selection to choose the best outline image per original image. We may reject an image exemplar here if the outline obtained does not represent the object very well. Many objects, such as liquids, moss, and foam, are more represented by their texture and colour than their shape. Hence they may not be fit to be represented with just an outline. Some manual editing of the outline may be done at this point if the object can be made perfectly visible by blackening some of the remaining outlines from background objects.

3.3. Generating constellation images from outlines

On the selected outlines, we run two algorithms to convert them into constellation images. First, we convert the outlines into dotted images. The objective of this step is to generate an image representing the outlines with dispersed dots. These dots are separated by a regular pixel distance 'd'. This pixel distance 'd' is a parameter that we can control as an input to the function. We first traverse the image pixel by pixel to find any white pixel on the outline image. We draw a black circle around the found pixel with radius 'd', leaving the original (central) pixel as white. When a traversal over the image with this procedure is complete, we obtain an outline represented by dots at regular intervals. Now we can increase the radius of these dots by doing

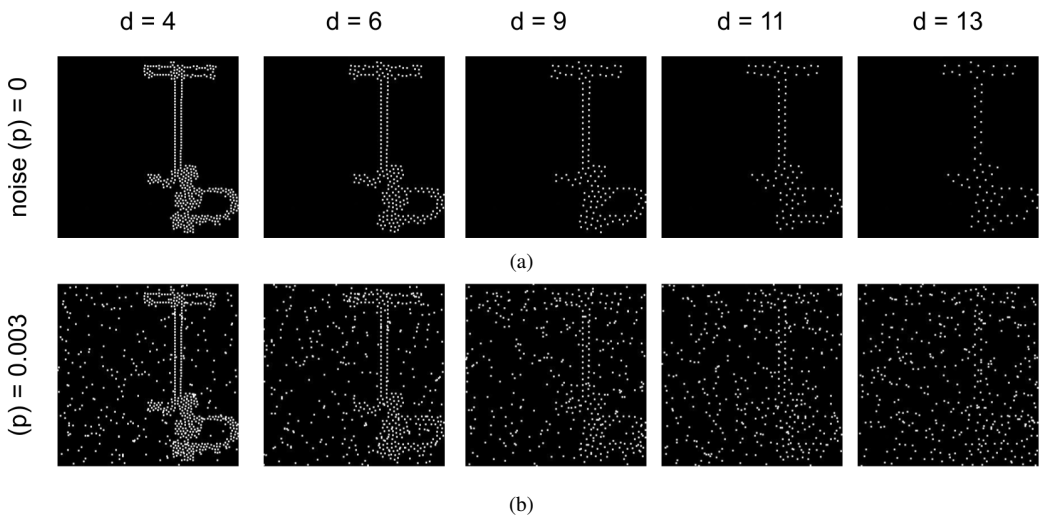

Figure 3. Dotted and constellation images with different difficulty levels. A combination of 'd' and 'p' is chosen to get the optimal difficulty level for images to be used in experiments. 3a: Dotted version with different distance 'd' between dots.3b The respective constellation images with noise (p = 0.003).

another pass and making a white circle of the required radius 'r' at each point when you encounter a white pixel in the image. The variations of image obtained after this step can be seen in Figure 3a.

Second, we add noise to the generated dotted image. In this step, the image is traversed again to randomly generate circles of radius 'r' with probability 'p' at each pixel, generating the noisy version. Figure 3b shows the final constellation images obtained after this step.

4. Dataset

The final dataset consists of 3533 image sets from a total of 1215 objects. The dataset consists of common objects of many types from the Things dataset, such as animals, kitchenware, appliances, furniture, vehicles, tools, musical instruments, food etc. Only the objects for which texture is an important cue to recognition (e.g., grass, ice, sauce) were removed as they are not suitable for the constellations dataset. Mostly 2 or 3 exemplars for a given object are provided in this final set.

The images released in the dataset are of size 320 $\times$ 320 pixels and represent the object in 4 modes: original image, outline image, dotted image, and final constellation image. Along with these variations in modality, the dotted and constellation images are also provided at different signal/noise ratios. This offers the benefit of a wide range of stimuli that can be chosen to fit different individual tasks. The signal/noise ratio is varied in one of two ways. First, it is possible to change the distance between two dots on the original image which corresponds to a change in signal, i.e., the farther the dots, the lower the signal. The other way is to directly increase the amount of noise added in the last step. Distances ranging from 4 to 17-pixel length with noise levels of 0.002 to 0.003 are used. Other than the main dataset used for various experiments we provide an additional constellations dataset obtained from 20,000 sketch images as drawn by humans covering 250 objects [11]. More generally, one can create more constellations images from other large sketch datasets like Sketchy and Quick, Draw! [4, 33]

The positions of dots in the dotted (ground truth positions) and constellation images can be easily extracted using the script that we release with the code. Other than for evaluating the sketches of the solutions, having the positions of the dots also allows one to effectively re-generate the constellation images in other resolutions. In addition, we release code for a set of tools that can be used to evaluate the sketches made by AI algorithms on the constellation images. This allows counting the numbers of dots belonging to the original outline of the object that the sketch passes through or nearby.

Finally, we also release a smaller set of 481 "top" images that have been hand-picked for using in experiments with humans. These were selected based on criteria extracted from pilot experiments that make constellation images more likely to be solved by human subjects while allowing for some alternative hypotheses, and that are not too easy or too difficult to solve. These include objects with a distinctive shape, presented in full and at an angle that is characteristic to the object. We also excluded very generic shapes that could belong to a large number of objects, objects that are rarely encountered, or contours with very straight lines that are likely to pop out immediately.

The code and link to the full dataset can be found

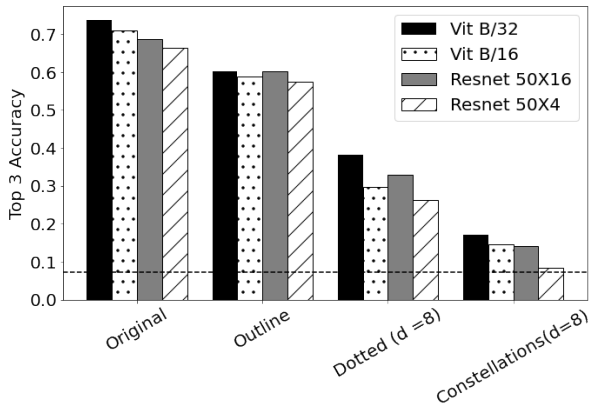

Figure 4. Model accuracy for CLIP versions Vit B/32,Vit B/16, Resnet 50X16, Resnet 50X4 models on different modalities of images in the constellation dataset. Dotted line shows the baseline top 3 accuracy for random prediction.

here: https://github.com/tarunkhajuria42/Constellations-Dataset

5. Evaluating pre-trained CLIP for Inference

CLIP is a joint image-text model that has been pre-trained on large datasets and works on a wide variety of text and images [54]. It has been used in many sketching solutions to guide the generation process [14, 57]. As the model is very actively used in designing sketching solutions, we evaluate its direct applicability to the constellation dataset. Note that in our work we are not so much interested in fine-tuning CLIP directly on constellation images, as human subjects in our experiments (see next section) are also not trained on constellations.

We evaluate four variants of CLIP (Vit B/32, Vit B/16, Resnet 50X16, Resnet 50X4) on various modalities of our dataset images by setting up a classification task based on the categories provided with the Things dataset [28] (setup details in supplementary materials). We find that the pre-trained model's performance drops drastically from the original image to the constellation image (see Figure 4). The performance with constellation images is at near random guess levels.

6. Human experiments show iterative inference

We conducted a study where participants were able to view the constellation images for an unspecified amount of time with the task to identify the object. They used a touchscreen monitor and stylus to trace the outline of the object. This process of sketching the outline of the object was included to examine the various hypotheses the participant may consider and allow them to specifically test their fit with the dots. Furthermore, sketching the outline

Figure 5. Examples of sketches made by different human participants on three constellation images. The top row (A) depicts correct guesses for the constellation images of a seahorse, a dog, and glasses. In the bottom row (B) participants have sketched alternative object guesses (a face, an acorn, and a car) for the same constellation images. The original images and dotted outlines of these stimuli can be seen in Figure 1.

of the objects ensures that participants do not provide random guesses without considering the evidence. Some examples from sketches made by human participants can be seen in Figure 5. Once the participant had identified an object, they had to write down their guess, as well as estimate how confident they were in their response on a 7 point scale. If the image was deemed too difficult, they were able to see a slightly easier version of the image with a reduced noise level. To gather more information about the process of solving the image we also recorded the participant's voice as they were prompted to explain in detail their thought process from forming a hypothesis to arriving at a solution. From the transcripts of these recordings, we were able to trace the iterative inference process in many instances, as participants considered alternative hypotheses before settling on a final answer. An illustration of a participant solving a constellation image in an iterative fashion is depicted in Figure 6.

Participants guessed the correct object on an average of 83.4% of the stimuli (SD=37.3%; range 68.6% - 94.3%), and made valid guesses on 97.1% of the trials (SD=16.7%; range between 94.3% and 100%).

The time it took participants to arrive at a solution varied from only a few seconds to over two minutes (min=2.15 s, max=143.8 s). The average time it took to arrive at a guess was 24.38 s (SD=28.79 s). Shorter guess times were linked to higher confidence ratings, as trials which were guessed under the average 24.38 seconds were rated higher in confidence (M=5.63, SD=1.22) than those which took longer to guess (M=4.0, SD=1.45). In future experiments, it is possible to select constellation images with the suitable difficulty level for a specific task based on the average reaction times it takes to solve the image.

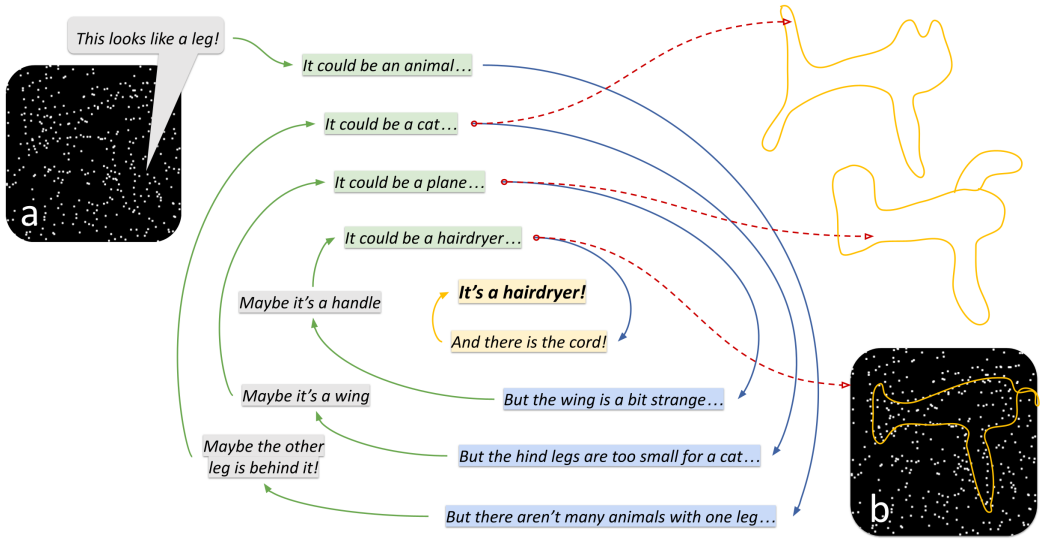

Figure 6. Illustration of iterative inference of a participant when solving the constellation image of a hairdryer (a), as extracted from the transcripts of the verbalized solving process. He first identifies a coherent pattern in the dots and proposes his initial object category hypothesis ("an animal"). He then continues to iteratively refine the hypotheses (green arrows) and test them against the data (blue arrows), until arriving at a solution that he deems the best match for the data at hand (b).

7. Discussion

Typical vision datasets like Imagenet [39], COCO [46], Things Dataset [28], and Ecoset [52] cater to tasks such as image recognition or object detection. However, under more complex circumstances, the human brain does not rely on a single pass of feature extraction from the image. Rather, in a process of iterative inference [64] it may also refine the feature extraction based on various hypotheses and accumulate supporting and contradicting features from an image before coming to a conclusion.

Here we have presented a dataset called constellations to aid the investigation of iterative inference in humans and AI algorithms. We have demonstrated in human subjects that humans are indeed depending on iterative inference to solve these images. We instructed the human participants to sketch their solutions, thus showing that the constellation images are also a tool for studying sketching under such circumstances.

The dataset consists of 3533 image sets from a total of 1215 objects, thus making it possible to study a large variety of concepts. For this particular dataset we used original images from the Things database [28], because these images only contained one object. We generated an additional set of 20,000 constellation image sets using sketches made by humans in [11]. However, in principle the code we provide can be applied to any image from any dataset, with sketch datasets needing only the fully automated pipeline. In the future, we will augment this dataset with more data from human subjects, describing their performance and solution process on these constellation images.

DL architectures have used recurrent connections and interactive inference for many tasks, but exploiting its true potential still faces problems in training [47]. We know from tasks such as image restoration that convolution as an operator can be used in image denoising when the denoising can be done using local information. Recurrent connections or iterative processing help model long-range self-information [49] relationships. In scene modeling tasks from the Multi-object dataset [34], where the objects are occluded, the relationship between the scene and the object is modeled by passing the information using the iterative loop [3, 12, 20] while the convolution pipeline models the objects locally.

Our dataset further promotes more iterative and generative solutions due to the lack of local information in the scene. We remove the local information from the images and add additional noise dots to further obscure local shape information. As the constellation images have no local information to rely on, the inference process is forced to look for information cues at various scales. Iterative hypothesis testing becomes an important part of the search as the solutions cannot rely on composition of bottom up (local information) information to learn possible global shape solutions as is done by most single-pass ANNs on images. For example, we observed that the classification performance of CLIP on these tasks is quite low (see Figure 4), so using CLIP for direct inference of labels or providing meaning-

ful initial conditioning of search on dotted or constellations images will be very difficult. However, it remains possible that some of this performance can be improved by training or fine tuning on contours, dotted images or constellations.

One possible direction for solving constellations includes using a GAN trained on sketches or outlines and then searching in the GAN's latent space for a solution contour that passes through the maximum number of dots in the constellation image. A bottom-up approach could be to model the search using primitives such as lines or curves connecting the dots in the image while searching for a combination giving a high concept score in a network like CLIP. Many CLIP-guided visual search [14, 57] methods already exist even combined with evolutionary search methods [62]. However, there are two important differences that make the constellation images different from previous work. First, the solution must be connected to the underlying constellation image instead of being drawn on an empty canvas. In other words, any solution needs to respect the dots on the constellation image. Second, the search is not guided by one, but by thousands of competing text prompts (as in principle there could be anything hidden in the constellation). Hence, the search space can be fairly large both in terms of sketches drawn and potential object labels.

A sketch by itself represents a fairly dynamic process. Hence, previous work to model sketches provides multiple ways in which the solution of constellation images might be approached. Works like [70] model sketches by not only their visual features, but also capture the temporal dynamics of sketch strokes using a RNN-based pipeline. Modelling sketches as motor strokes as done in [22] provides another way to align the representation of sketches to human representations, where the sketching process represents natural iterative inference by conditioning next generation steps on drawn parts. In [71] the authors introduce a multi-graph based representation of sketches and model them using GNNs. [69] proposes a method to extract self-supervised sketch specific representations, proposing transformations that preserve the inherent identity in object sketches. All these methods provide important building blocks to solve constellation images. Stroke-based representations in particular inherit a natural element of iterative inference conditioned on the already generated sketch. However, to solve constellation images, the sketching process has to be further conditioned on the underlying constellation image.

8. Limitations

The applicability of constellation images for problems is limited in certain ways. Firstly, generating good constellation images is dependent on obtaining a clear outline from the original image, but if the Mask R-CNN output is suboptimal or the object contains complex texture, this causes the edge detector to output a very unclear outline. Secondly, the selection of an appropriate difficulty level of the constellation images for experiments with humans or computer vision algorithms is not trivial. A particular combination of distance between dots 'd' and noise probability 'p' used to generate constellation images may not produce the same difficulty level on all objects as objects differ wildly in their surface and conceptual features. Lastly, the scale of objects in the image can vary: a circular shape could be a football or a coin. This may pose a problem for objectively evaluating the solution sketches. Automated evaluation is problematic as single stimuli may have other valid object shapes that pass through its points. Also, avoiding algorithms taking shortcuts and converging to trivial solutions by sketching a circle or a rectangle and labelling them as a clock or box remains a challenge. Due to this, evaluating if such solutions form a valid object sketch is still largely dependent on human evaluation. At first sight it might seem that the validity of our CLIP experiments is limited to show the need for iterative inference, as we have not trained or fine-tuned CLIP on the constellation images. However, note that humans are also not trained directly on constellation images, but rather naturally use iterative inference when first confronted with these images. Nevertheless, we could in principle fine-tune CLIP or other networks on constellation images, as we can expand our dataset using outlines from multiple sketch datasets [4, 11, 33]. This will allow us to properly benchmark the performance of the existing DL architectures on this dataset.

9. Conclusion

With this novel dataset, the task to find an object in the constellation images is a step towards more difficult image recognition which requires hypothesis generation, analysis, and synthesis before arriving at the final solution. This task by itself provides the simplest setting to examine the algorithms that humans use for iterative inference. We have demonstrated that humans indeed require time and several iterations before coming to the solution. Therefore, this dataset provides a way to study and implement human-like iterative inference in machine vision.

10. Ethics statement

The human experiments reported in this work were conducted with permission from the University of Tartu Research Ethics Committee.

11. Acknowledgements

We thank Oriol Corcoll for helpful comments. This research was supported by the European Social Funds through the IT Academy Programme and the Estonian Research Council grant PSG728.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 3

[2] Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Jorma Laaksonen, and Michael Felsberg. Doodleformer: Creative sketch drawing with transformers. *arXiv preprint arXiv:2112.03258*, 2021. 3

[3] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 3, 7

[4] Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, Oleg Sushkov, David Barker, Jonathan Scholz, Misha Denil, Nando de Freitas, and Ziyu Wang. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200*, 2019. 5, 8

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 4

[6] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018. 3

[7] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7239–7248, 2018. 3

[8] John P Collomosse, Graham McNeill, and Yu Qian. Storyboard sketches for content based video retrieval. In *2009 IEEE 12th International Conference on Computer Vision*, pages 245–252. IEEE, 2009. 3

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[10] Jan Drewes, Galina Goren, Weina Zhu, and James H Elder. Recurrent processing in the formation of shape percepts. *Journal of Neuroscience*, 36(1):185–192, 2016. 1

[11] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):44:1–44:10, 2012. 5, 7, 8

[12] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019. 3, 7

[13] Chrisantha Fernando, SM Eslami, Jean-Baptiste Alayrac, Piotr Mirowski, Dylan Banarse, and Simon Osindero. Generative art using neural visual grammars and dual encoders. *arXiv preprint arXiv:2105.00162*, 2021. 3

[14] Kevin Frans, LB Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*, 2021. 3, 6, 8

[15] Anibal Fuentes and Jose M Saavedra. Sketch-qnet: A quadruplet convnet for color sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2134–2141, 2021. 3

[16] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31, 2018. 1

[17] Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350–363, 2013. 2

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[19] Clément Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *Proceedings of the European conference on computer vision (ECCV)*, pages 538–554, 2018. 3

[20] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019. 3, 7

[21] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471. PMLR, 2015. 3

[22] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017. 3, 8

[23] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018. 3

[24] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3897–3906, 2019. 3

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 4

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[27] Douglas Heaven et al. Why deep-learning ais are so easy to fool. *Nature*, 574(7777):163–166, 2019. 2

[28] Martin N Hebart, Adam H Dickter, Alexis Kidder, Wan Y Kwok, Anna Corriveau, Caitlin Van Wicklin, and Chris I

Baker. Things: A database of 1,854 object concepts and more than 26,000 naturalistic object images. *PloS one*, 14(10):e0223792, 2019. 1, 3, 6, 7

[29] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 3

[30] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991. 3

[31] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 3

[32] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016. 1, 2

[33] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The quick, draw!-ai experiment. *Mount View, CA, accessed Feb*, 17(2018):4, 2016. 5, 8

[34] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019. 3, 7

[35] Kohitij Kar, Jonas Kubilius, Kailyn Schmidt, Elias B Issa, and James J DiCarlo. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature neuroscience*, 22(6):974–983, 2019. 1, 2

[36] Hui Ying Khaw, Foo Chong Soon, Joon Huang Chuah, and Chee-Onn Chow. Image noise types recognition using convolutional neural network with principal components analysis. *IET Image Processing*, 11(12):1238–1245, 2017. 3

[37] Tim C Kietzmann, Courtney J Spoerer, Lynn KA Sörensen, Radoslaw M Cichy, Olaf Hauk, and Nikolaus Kriegeskorte. Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences*, 116(43):21854–21863, 2019. 1, 2

[38] Junkyung Kim, Drew Linsley, Kalpit Thakkar, and Thomas Serre. Disentangling neural mechanisms for perceptual grouping. *arXiv preprint arXiv:1906.01558*, 2019. 3

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2, 7

[40] Jonas Kubilius, Martin Schrimpf, Aran Nayebi, Daniel Bear, Daniel LK Yamins, and James J DiCarlo. Cornet: modeling the neural mechanisms of core object recognition. *BioRxiv*, page 408385, 2018. 2

[41] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 3

[42] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018. 3

[43] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2

[44] Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1403–1412. IEEE, 2019. 3

[45] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3367–3375, 2015. 2

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 4, 7

[47] Drew Linsley, Alekh Karkada Ashok, Lakshmi Narasimhan Govindarajan, Rex Liu, and Thomas Serre. Stable and expressive recurrent vision models. *Advances in Neural Information Processing Systems*, 33:10456–10467, 2020. 7

[48] Drew Linsley, Junkyung Kim, Vijay Veerabadran, Charles Windolf, and Thomas Serre. Learning long-range spatial dependencies with horizontal gated recurrent units. *Advances in neural information processing systems*, 31, 2018. 3

[49] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. *Advances in neural information processing systems*, 31, 2018. 3, 7

[50] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018. 3

[51] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 3

[52] Johannes Mehrer, Courtney J Spoerer, Emer C Jones, Nikolaus Kriegeskorte, and Tim C Kietzmann. An ecologically motivated image dataset for deep learning yields better models of human vision. *Proceedings of the National Academy of Sciences*, 118(8), 2021. 1, 7

[53] Randall C O'Reilly, Dean Wyatte, Seth Herd, Brian Mingus, and David J Jilk. Recurrent processing during object recognition. *Frontiers in psychology*, 4:124, 2013. 1

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learn-

ing transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 6

[55] J Redmon, S Divvala, R Girshick, and A Farhadi. You only look once: Unified, real-time object detection. arxiv 2015. *arXiv preprint arXiv:1506.02640*, 2015. 1, 2

[56] Pieter R Roelfsema, Victor AF Lamme, and Henk Spekreijse. The implementation of visual routines. *Vision research*, 40(10-12):1385–1411, 2000. 2

[57] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. Styleclipdraw: Coupling content and style in text-to-drawing translation. *arXiv preprint arXiv:2202.12362*, 2022. 3, 6, 8

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[59] Courtney J Spoerer, Tim C Kietzmann, Johannes Mehrer, Ian Charest, and Nikolaus Kriegeskorte. Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. *PLoS computational biology*, 16(10):e1008215, 2020. 2

[60] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017. 3

[61] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020. 3

[62] Yingtao Tian and David Ha. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. *arXiv preprint arXiv:2109.08857*, 2021. 8

[63] Pablo Torres and Jose M Saavedra. Compact and effective representations for sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2115–2123, 2021. 3

[64] Ruben S van Bergen and Nikolaus Kriegeskorte. Going in circles is the way forward: the role of recurrence in visual inference. *Current Opinion in Neurobiology*, 65:176–193, 2020. 1, 7

[65] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14050–14060, 2021. 3

[66] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 864–873, 2018. 3

[67] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020. 3

[68] Yao Xie, Peng Xu, and Zhanyu Ma. Deep zero-shot learning for scene sketch. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3661–3665. IEEE, 2019. 3

[69] Peng Xu, Timothy M Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep learning for free-hand sketch: A survey and a toolbox. *arXiv preprint arXiv:2001.02600*, 2020. 3, 8

[70] Peng Xu, Yongye Huang, Tongtong Yuan, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, and Jun Guo. Sketchmate: Deep hashing for million-scale human sketch retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8090–8098, 2018. 8

[71] Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multigraph transformer for free-hand sketch recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 8

[72] Peng Xu, Kun Liu, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, Jun Guo, and Yi-Zhe Song. Fine-grained instance-level sketch-based video retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1995–2007, 2020. 3

[73] Di Zhao, Lan Ma, Songnan Li, and Dahai Yu. End-to-end denoising of dark burst images using recurrent fully convolutional networks. *arXiv preprint arXiv:1904.07483*, 2019. 3