CONCEPT DISCOVERY AND DATASET EXPLORATION WITH SINGULAR VALUE DECOMPOSITION

Mara Graziani[1], **An-Phi Nguyen**[2], **Laura O' Mahony**[1,3]*, **Henning Müller**[1,4], **Vincent Andrearczyk**[1]
[1] Haute école spécialisée de Suisse occidentale, Hes-so Valais, Sierre, Switzerland
[2] Biognosys AG, Schlieren, Zürich, Switzerland
[3] University of Limerick, Limerick, Ireland
[4] The Sense Research and Innovation Center, Sion, Lausanne, Switzerland

ABSTRACT

Providing reliable and trustworthy predictions as the outcome of deep learning models is a major challenge, particularly in supervised settings that include misleading training annotations. Concept-based explanations clarify the relevance of high-level concepts to the model predictions, although this may be biased by the user's expectations of the concepts. Here we propose a post-hoc unsupervised method that automatically discovers high-level concepts learned by intermediate layers of vision models. By the singular value decomposition of the latent space of a layer, we discover concept vectors that correspond to orthogonal directions of high variance and are relevant to the model prediction. Most of the identified concepts are human-understandable, coherent and relevant to the task. Moreover, by using the discovered concepts we identify training samples with confounding factors that emerge as outliers. Our method is straightforward to implement, and it can be easily adapted to interpret multiple architectures and identify anomalies in the data collection.

1 INTRODUCTION

The supervised pre-training of a model's parameters on large scale datasets such as ImageNet (Russakovsky et al., 2015) is a de-facto standard in multiple applications. Little attention has been given, however, to labeling mistakes that occur in the training dataset, and to long-tail errors that are counter-intuitive even to domain-experts Vasudevan et al. (2022). It is no secret that labelling errors and noise would impact the quality and evaluation of the model, at times introducing undesired and harmful biases. Unfortunately, the larger the dataset sizes, the harder it becomes to properly assess the quality of the supervised labels. Deep learning models were shown to be overconfident Guo et al. (2017), unreliable Nguyen et al. (2015), and biased towards simple first order statistics features such as texture Geirhos et al. (2018). In this work, we show how matrix factorization can be used to analyze the patterns learned by a deep learning model. Our framework, based on Singular Value Decomposition (SVD) automatically identifies vectors at intermediate representations that can be associated with high-level, human-understandable concepts. These directions are then used to explore the training dataset and identify inputs presenting artefacts, confounding factors, or wrong labels.

Empirical evidence has already demonstrated that high-level concepts such as objects, shapes and texture detectors are learned by intermediate model representations Bau et al. (2017); Kim et al. (2018); Graziani et al. (2018); Raghu et al. (2017). The work on Concept Activation Vectors (CAVs), in particular, identified unit vectors in the latent space that discriminate the learned representations from those presenting a high-level concepts from those without. Yet, the existing work on *concept discovery* is limited, with only a few methods that automatically analyze a layer's activations without requiring user-defined directives and questions Ghorbani et al. (2019); McGrath et al. (2022); Raghu et al. (2017); McGrath et al. (2022). Different from the work on CAVs, discovery methods do not require user-defined queries and are, as such, *unbiased* towards the user's knowledge and expectations Rosenthal & Fode (1963). These methods may force us to pay attention to some concepts, or patterns that were neglected in the model analysis. The work in Ghorbani et al. (2019),

*Work performed at Hes-so Valais.

for example, identifies image segments that cluster in the latent space, and uses CAVs to determine if the direction of a cluster is a relevant concept for the model. Black and white striped textures emerge as the main concept for recognizing zebras, and vehicle parts emerge as the main concept to detect police vans as in Ghorbani et al. (2019). On a separate line, factorization approaches are based on the interpretation of neurons as the basis vectors of a space, that is the latent space of a layer. McGrath et al. (2022), for example, propose non-negative matrix factorization to decompose the activations in AlphaZero and identify principal vectors that describe the model's playing mechanics in chess. The vectors identified with this method, however, were not directly associable with high-level concepts and no analysis was performed on more intuitive tasks such as natural image classification.

We propose a general framework for concept discovery that can be applied to multiple architectures, tasks and data types. Our method is a novel approach to automatically discover concept vectors through the decomposition of a layer's latent space into matrices consisting of singular values and vectors. We identify concepts that are human-understandable and consistent with the results of previous research, even if being computed only on 1% of the original training dataset. Similarly to Bau et al. (2017); Ghorbani et al. (2019), our concept vectors point at precise concepts, such as textures and object parts. Our user evaluation studies demonstrate that the discovered concepts are easy to understand, allowing the user to inspect a model's decision process. Extending our work in Graziani et al. (Under Review.), here we offer a novel analysis of how the discovered concept vectors can be used to retrieve outlier images from the dataset with incorrect or confounding labels.

2 Concept discovery in latent spaces

Based on the observations of Kim et al. (2018), we assume that directions in the latent space of a layer can identify the concepts used by a model for inference. However, while Kim et al. (2018) rely on user-defined concept examples to find the concept vectors, here we aim to reverse-engineer the problem. Our objective is to identify concept vectors that directly emerge as relevant directions impacting the model decision function. We further assume, as suggested by Chen et al. (2020), that concept vectors should be orthogonal to each other to maximize the separability of the concepts. the objective of our method is to identify the set of orthonormal vectors in the latent space that carry most of the information, i.e. that most impact the model's prediction.

The method consists of three phases, namely: (i) identification of orthogonal vectors via the singular value decomposition (SVD) of the activations of a layer; (ii) ranking of the singular vectors based on high gradients and activation values; and (iii) selection of the top vectors in the new ranking and their assignment to human-understandable concepts through visualization.

2.1 Notation

We consider the prediction function of a neural network $f : \mathbb{R}^m \to \mathbb{R}^n$ from an m-dimensional input vector to an n-dimensional output vector. We assume the model was already trained using a dataset consisting of N labeled pairs of input data points and labels $\{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^m \times \mathbb{R}^n$. Given an arbitrary layer l, the neural network can be seen as a composition of a feature extractor $\phi^l : \mathbb{R}^m \to \mathbb{R}^d$ and a downstream predictor $\psi^l : \mathbb{R}^d \to \mathbb{R}^n$, i.e. $f(x) = \psi^l(\phi^l(x))$. In the following, to simplify notation, and since we consider a single layer at a time, we drop the superscript identifying the layer l. Furthermore, for a given input x_i, we use the shorthand $\phi_i = \phi(x_i)$. We are interested in finding M orthonormal vectors $u_1, .., u_M \in \mathbb{R}^d$.

For the sake of generality, we present the method for $n = 1$ and for densely connected layers. The method can be extended to $n > 1$, for example multi-class classification by applying the same construction to each of the outputs. Similarly, for convolutional layers, a pooling operation is introduced to obtain a d-dimensional representation. Details about both extensions are given in the Appendix B.

2.2 Step 1: SVD

Step one takes N input images from the train dataset, and it identifyies d orthonormal vectors that best summarize the encoding of the dataset in the d-dimensional latent space of a layer l. We apply SVD to the matrix of the entire layer's response to the input dataset $\Phi \in \mathbb{R}^{d \times N}$, which is obtained by column-stacking the latent representations ϕ_i, with $i = 1, \ldots, N$. We obtain:

$$\Phi = U \Sigma V^T \tag{1}$$

Note that $U \in \mathbb{R}^{d \times d}$, $V^T \in \mathbb{R}^{N \times N}$, and Σ is a diagonal matrix of singular values $\sigma_1, \ldots, \sigma_d$. The left singular vectors are the columns of the matrix U, and they align with the variance in Φ. The singular values rank the directions from the largest to the lowest observed variance. In convolutional neural networks (CNNs), the feature extraction is $\phi^l : \mathbb{R}^{h' \times w' \times c} \to \mathbb{R}^{h \times w \times d}$ and it maps an input image x_i to d feature maps of width w and height h. We compute the SVD after the aggregation of the spatial information by a global average pooling operation, hence $\Phi \in \mathbb{R}^{d \times N}$, where d is the number of channels. More precisely, by pooling, we reduce each $\phi(x_i) \in \mathbb{R}^{h \times w \times d}$ to a d-dimensional vector.

2.3 Step 2: Gradient-informed ranking of the singular vectors

At this point, we are only making use of the feature extractor ϕ, and we are not considering whether the singular vectors are actually used to solve the downstream task. This means that the vectors found in the previous step might not be particularly relevant to the downstream predictive task. This second step evaluates the relation of the singular vectors with respect to the output function. More specifically, we evaluate the perturbation impact of moving the feature representation of each input along the direction of the singular vector. As in Kim et al. (2018), this is done by considering the directional derivative of the model output along the singular vectors. This operation is computed for all the singular vectors in U.

We consider the gradient of the downstream predictor ψ (Section 2.1) with respect to the latent space, which we denote $\nabla_\phi \psi_i$ for input ϕ_i. To compute the directional derivatives, we project the gradient onto the singular vectors in U: $\tilde{\nabla}_\phi \psi_i = U^T \nabla_\phi \psi_i$.

At this point, it is important to note that an input data point may have large gradients for low-activation features, and vice-versa, high activation values may be annihilated by close-to-zero gradients. Therefore, we consider the joint impact of gradients and activations together. Let

$$\tilde{\phi}_i = U^T \phi_i \tag{2}$$

denote the projection coefficients of ϕ_i on the singular vectors. We consider the (element-wise) product between the projection coefficients of the activations $\tilde{\phi}_i$ and of the gradients $\tilde{\nabla}_\phi \psi_i$:

$$\tilde{g}_i = \tilde{\nabla}_\phi \psi_i \odot \tilde{\phi}_i \tag{3}$$

Finally, to compute the ranking, we consider the overall importance of a singular vector to the prediction. This is simply given by the sample mean of the g_i over all inputs i in the dataset: $\tilde{g} = \frac{1}{N} \sum_i \tilde{g}_i$. Note, $\tilde{g} \in \mathbb{R}^d$. For simplicity of the notation, we drop the tilde in the rest of the paper. Simply put, this step replaces the conventional ranking of the singular vectors given by the values in Σ with a new ranking based on the value of the projected gradients. In classification tasks with $n = K$ classes, we compute a separate $g_k \in \mathbb{R}^d$ for each class k. To obtain the ranking, we further evaluate how each g_k compares to the values obtained for the other classes. For instance, we compare g_k to the distribution of the g_k^- obtained for all the input data points that are not of class k, hence for all the K classes except k. Please refer to the Appendix B for additional details.

2.4 Step 3: Candidate directions for discovery

Finally, the third step uses the values of $\tilde{g}$ to identify the first M directions in the ranking as the candidate vectors for discovery. As there is no guarantee that the discovered concepts will point to human-understandable patterns or features, the candidate vectors need to be confirmed as discovered concept vectors through human interfacing. The simplest way to obtain insights about a candidate vector is by data analysis. We can project some input images, provided that they match the distribution of the data used to train the model, along the direction of the vector and retrieve input samples with increasing projection values. In dense models, the candidate vectors can also be used as feature importance estimates. The elements of a candidate vector u are used as weights that are multiplied element-wise to the feature importance values obtained by back-propagating the model's gradients all the way to the input. Section 2.5 gives more detail on how to obtain concept visualizations and interpretations.

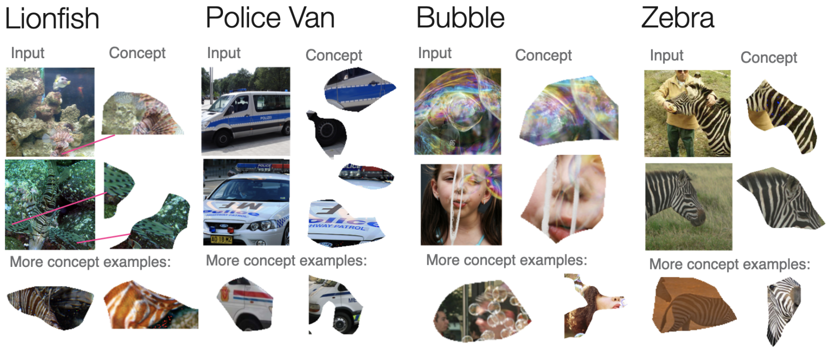


Figure 1: Visualization of the discovered concept vectors for ImageNet classes. In the first two rows, the input image is shown together with a zoomed-in version of the automatically segmented concept. The last row shows the input images with largest projection on the concept vectors and the relative concept segmentation masks. Our results are in line with previously discovered concepts for these classes.

2.5 Concept visualization

Visualization is a useful tool to verify the human-understandability of the discovered concepts. We define concept activation maps as the visualization of the model's response, for a given concept, to a certain input. The computation of such maps is straightforward for convolutional networks. For a discovered concept direction $u \in \mathbb{R}^d$, and an input image x_i, we can visualize the layer's activation responding to the discovered concept as the sum of the d feature maps for each channel $\phi_{i,1}, \ldots, \phi_{i,d}$ weighed by the concept vector coefficients. In other words, one can take the feature maps of each channel and weight them by the coefficient values of the concept vector[1]. Note, if we were to consider a concept that is fully aligned with the n-th feature map with $n < d$, e.g. $u = (0, ..., 0, 1, 0, ..., 0)$, that is non-zero only for u_n, the visualization for x_i would correspond to the n-th feature map itself. Concept segmentation masks can be directly derived from the concept maps by retaining the input pixel values with concept activation higher than the 80-th percentile in the concept maps as in Bau et al. (2017).

2.6 Dataset Exploration

Concept discovery can be used for dataset exploration, for example, to find input samples that are mislabeled or that contain confounding factors. Given the representation of an input sample in the latent space, we project it onto the concept vectors as in Eq. 2. Anomalous samples are then identified based on the statistical dispersion of the data projected onto the concept vectors. For instance, we isolate 10% of the representations that fall outside of the interquartile range of the projection coefficients for the training data. The inputs corresponding to these representations are flagged for further inspection, as they may contain artefacts or potentially misleading confounding factors.

3 Results

This section gathers the experimental results obtained by applying concept discovery to multiple tasks and models. The experiments are run on de-facto standard models for which pretrained weights are available online and interpretability research has already given multiple insights. Here we focus on Inception V3 (IV3) with pretrained weights Szegedy et al. (2016) on the ImageNet ILSVRC2012 dataset Russakovsky et al. (2015). We first show with empirical evidence that our concept discovery method can be used to automatically identify and recognize concepts of natural image categories in Section 3.1. We then assess our method with human evaluation and quantitative studies in Section 3.2. Finally, we analyse the results of dataset exploration in Section 3.3.

[1] Being the concept vector a singular vector, it has norm of one.

Note that concept discovery is applied only to 1% of the entire ImageNet training dataset to limit the memory requirements and make the computation accessible to our infrastructure. We undersample, for instance, by retaining only one of every hundred training images. Where not stated otherwise, we consider the concatenation layer *Mixed 7b*, which is a deep layer close to the end of the model and where we expect to identify complex concepts. Similar analyzes can be performed at multiple depths, layers and even on entirely different architectures, as demonstrated in the Appendix D.

3.1 Discovered Concepts in IV3

For comparative purposes, we focus on classes that were already analyzed in related works Ghorbani et al. (2019); Bau et al. (2017); Kim et al. (2018), namely *lionfish, police-van, bubble and zebra*. We identify (with $M = 1$) one concept vector for each class, four orthogonal vectors in total. Figure 1 depicts the visualization outputs, which illustrate different types of concepts, ranging from patterns (as in the *lionfish* fins and the *zebra* coat), to graphics and tires (*police van*), and to glossy reflections (*bubble*). The concept segmentation masks are obtained in a completely unsupervised way as explained in Section 2.5. The concepts are automatically segmented at different scales, and the segmentation is robust to repetitions of a concept within the same image. Additional concept visualizations for other classes than the selected ones and for ResNet50 are provided in the Appendix D, and a complete analysis for all classes will be made accessible online.

3.2 Human Evaluation and Quantitative Assessment

Following the literature Ghorbani et al. (2019), we incorporated a human evaluation test with 30 participants[2]. The test was comprised of experiments on (i) intruder detection (ii) concept meaningfulness and (iii) inter-user agreement. The users were introduced to the test by an exemplar question and the relative correct answer (Appendix Figure 7). The intruder detection experiment (i) was designed to evaluate the coherency of ten discovered concepts for ten different classes. Each question showed a visualization of four concept segmentation masks obtained from the most relevant concept vector, and one random segmentation mask obtained for a different (non-relevant) concept vector. In each question, they were asked to identify the outlier image, namely the image being conceptually different from the rest. On average the participants were able to identify the intruder with an accuracy of 0.88. To evaluate the concept meaningfulness (ii), we asked participants to label the concepts based on the visualization of the concept segmentation masks for three images. Despite the small image segments illustrated by the concept segmentation masks, all participants correctly labeled all the images, showing that the concept segmentation masks were easy to interpret and associate with a concept. Finally, we evaluate inter-user agreement (iii) by asking each user to agree or disagree with the concept labels defined by other participants. Out of 30 users, the inter-user agreement was above 91% for all questions.

As a quantitative evaluation of the importance of the concept vectors, we compute the impact of removing the discovered concepts as in Ghorbani et al. (2019). Occlusion is performed on the input pixels with high values in the concept map of each concept, for instance, the input pixels with concept map values higher than the 80-th percentile of the values. We quantify the smallest destroying concepts (SDC) as the smallest number of concepts to remove in order to see a performance degradation on at least 80% of the dataset. Such degradation is observed on approximatively 400 classes when the first most relevant concept is removed, and it extends to almost all ImageNet classes (962 out of 1000) when we remove the first five concepts (as illustrated in the Appendix Figure 9).

3.3 Dataset Exploration

We visualize some of the inputs that are flagged by the dataset exploration with concept discovery. Our method flags 23% (0.2% of ImageNet) of the training set for which the representation are distributed as outliers. For 184 of these images (1.8% of the training set) the model prediction is wrong, differing from the ground-truth label. The flagged images belong to the training dataset, hence they represent hard training examples that may be confusing the model rather than making it more robust. This is confirmed by the lower top-1 accuracy on the flagged images at 0.90, against the accuracy on the rest of the training images at 0.93. The accuracy on the images flagged by our

[2] The evaluation test can be accessed at `https://forms.gle/MJ63G984ERvozuF38`

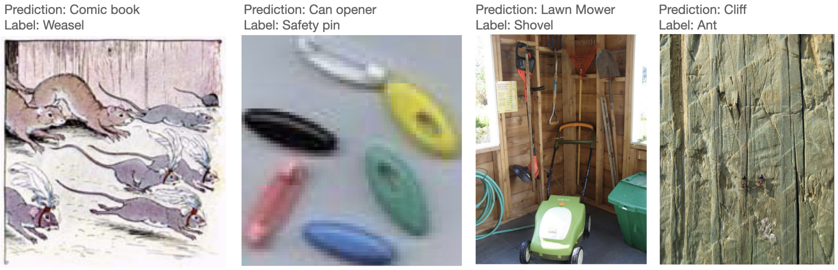


Figure 2: Results of dataset exploration with concept discovery. The method identifies training images with particular issues. The first example presents a strong style shift from real images to drawings. Extremely poor resolution affects the quality of the second input. The last two images present confounding factors. Multiple labels are equally correct for the third image, and the last image shows an optical illusion. Where there seems to be a cliff, there is actually a high resolution detail of two ants on a wooden surface.

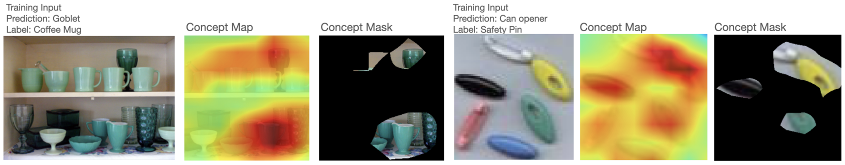


Figure 3: Visual explanations for two training image examples that were flagged as outliers. Multiple equally valid labels exist for the first example, and here the model is focusing on both the coffee mug and the goblet in the image. Because of the low resolution of the safety pin example, the silver reflection on the top right is misunderstood by the model as a can opener.

method is also lower than that obtained on images flagged by using a random neuron or a random direction, both of which have an accuracy of 0.92. This confirms that the discovered concept vectors identify directions in the latent space that are more informative for dataset exploration than random neurons or vectors.

Figure 2 shows some examples flagged by our method. The flagged images present either strong variations in the style and image resolution (e.g., the first two images from the left) or contain confounding factors. In the third image from the left, for example, multiple labels are equally correct, as both a shovel and a lawn mower are present in the image. Additional examples are shown in the Appendix. To interpret the result, as in Figure 3 we visualize the concept maps and the segmentation masks for some of the training images that are flagged as outliers. The model is deceived by multiple object categories in the same image, and by optical illusions that are enhanced by poor resolution as in the case of the *safety pin* image being wrongly classified as *can opener*.

We evaluate the utility of dataset exploration for concept discovery on 1000 randomly selected validation images. The method flags 22% of the validation images for further inspection. The top-1 accuracy on these images is 0.73, against 0.78 on the rest of the validation images. To verify if these images are actually harder samples for the model to classify, we perform uncertainty estimation by approximating an ensemble of five models with Monte Carlo Dropout Gal & Ghahramani (2016). The aleatoric uncertainty [3] is higher on the flagged images (at 2.13) than on the rest of the validation set (at 1.96). Figure 10 of the Appendix illustrates that the flagged validation images show similar artefacts to those in Figure 3, such as the shift from real images to drawings, poor resolution and multiple object categories in the same image.

[3] Assessed by the expected entropy as in Malinin et al. (2022)

4 Conclusions

We proposed a simple yet effective method that can be used to identify concept vectors representing informative signal in the representation of a layer. From the empirical results and user studies we can observe that most of the identified concepts are understandable and aligned with human reasoning. At the same time, it is also true that the automatic discovery of the latent concepts used by complex models is a challenging research direction. There is no theoretical guarantee that human-understandable concepts will always emerge when interpreting deep learning models. In most scenarios, intelligible behaviour and uninterpretable signals are likely to coexist, with models sharing both similarity and differences in the processing of the input signals. Even in such cases, we believe that concept discovery with SVD is a tool for unbiased interpretability, since it does not require any pre-defined preference on the type of concepts being analysed. We demonstrate that such type of analsyis can be useful to identify novel patterns and artefacts that affect the training of deep learning models. This can be particularly beneficial in limited data regimes or in fields where deep learning models are accelerating knowledge discovery such as chemistry and biology.

5 Acknowledgements

This work was supported by AI4Media of the European Union's Horizon 2020 research and innovation program under grant agreement No 951911.

References

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

Mara Graziani, Vincent Andrearczyk, and Henning Müller. Regression concept vectors for bidirectional explanations in histopathology. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pp. 124–132. Springer, 2018.

Mara Graziani, An-Phi Nguyen, Henning Müller, and Vincent Andrearczyk. Concept discovery in latent spaces with singular value decomposition. Under Review.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Andrey Malinin, Andreas Athanasopoulos, Muhamed Barakovic, Meritxell Bach Cuadra, Mark JF Gales, Cristina Granziera, Mara Graziani, Nikolay Kartashev, Konstantinos Kyriakopoulos, Po-Jui Lu, et al. Shifts 2.0: Extending the dataset of real distributional shifts. *arXiv preprint arXiv:2206.15407*, 2022.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Robert Rosenthal and Kermit L. Fode. The effect of experimenter bias on the performance of the albino rat. *Behavioral Science*, 8(3):183–189, 1963. doi: https://doi.org/10.1002/bs.3830080302. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/bs.3830080302.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Vijay Vasudevan, Benjamin Caine, Raphael Gontijo-Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. When does dough become a bagel? analyzing the remaining mistakes on imagenet. *arXiv preprint arXiv:2205.04596*, 2022.

A Appendix

B Extension to Vision Classification Tasks

Our approach can be easily extended to fit a variety of data types, latent spaces and model tasks. Let us consider a convolutional neural network (CNN) $f : \mathbb{R}^{h' \times w' \times c'} \to \mathbb{R}^K$ classifying an input image in $n = K$ classes. Particularly, $f_{i,k}$ is the predicted probability of input x_i belonging to class k , namely $p(y = k|x = x_i)$. N_k is the number of samples in class k, with $N = \sum_{k=1}^{K} N_k$. For convolutional layers, the feature extraction is $\phi^l : \mathbb{R}^{h' \times w' \times c} \to \mathbb{R}^{h \times w \times d}$ and it maps an input image x_i to d feature maps of width w and height h.

We compute the SVD in the latent space of a convolutional layer, after the aggregation of the spatial information by a global average pooling operation, hence $\Phi \in \mathbb{R}^{d \times N}$, where d is the number of channels. More precisely, by pooling, we reduce each $\phi(x_i) \in \mathbb{R}^{h \times w \times d}$ to a d-dimensional vector.

Step 2 is also modified to consider K-classification tasks with CNNs. For each class k, we compute a separate $g_k \in \mathbb{R}^d$. To obtain the ranking, we further evaluate how each g_k compares to the values obtained for the other classes. For instance, we compare g_k to the distribution of the g_k^- obtained for all the input data points that are not of class k, hence for all the K classes except k.

Formally, for each class k we compute the average of the projection of g_k on the singular vectors:

$$g_k = \frac{1}{N_k} \sum_{i, y_i = k} U^T g_{i,k}, \tag{4}$$

where $g_{i,k} \in \mathbb{R}^d$ is the global average pooling of $\phi(x_i) \odot \nabla_\phi \psi_{i,k}$[4]. We then compute the sample mean and standard deviation of the values obtained for all the rest of the data, namely for all classes except k. The mean is obtained, for instance, by averaging over all the samples x_i such that $y_i \neq k$:

$$g_k^- = \frac{1}{N - N_k} \sum_{i, y_i \neq k} U^T g_{i,k'}. \tag{5}$$

Similarly, the variance is computed as:

$$\sigma_k^{-\,2} = \frac{1}{N - N_k} \sum_{i, y_i \neq k} (U^T g_{i,k'} - g_{k'}^-)^2. \tag{6}$$

Finally, the importance scores for each singular vector in U are given by:

$$z_k = \frac{g_k - g_k^-}{\sigma_k^-} \tag{7}$$

where $z_{k,j}$ is the importance score of the j-th column in U. This measure is then used to obtain a class-specific ranking, from which we retain the first M positions to identify M concept vectors.

C Additional Results on IV3 and ResNet50

We provide additional results for 16 classes, namely *airliner, clock, corkscrew, albatross, border collie, road sign, flamingo, mushroom, artichoke, hammerhead shark, screwdriver, iPod, tench fish, suspension bridge, umbrella* and cucumber. The concept segmentation masks for the first most important concept are shown in Figure 5. The concepts were resized to fit in the square, but they originally appear at multiple scales in the input images.

Results obtained on ResNet50 (at *layer_4.0.add*) are shown in Figures 6 and 4. The visualizations point to concepts that are similar to those highlighted in IV3 visualizations, such as the fish fins, the zebra stripes, the glass-like reflections and van tires and logos. Figure 4 shows the input images in the dataset that have the largest projection value on the concept vector for the four analysed classes. We can see the striped pattern emerging again for the class zebra, although in this case the color of the stripes seems to be given less importance. For the class *bubble*, the shape of the bird belly and their repeated presence may share a similarity with the images in the bubble class.

[4] Note, the pooling is here applied after the element-wise product of the features maps and the gradients

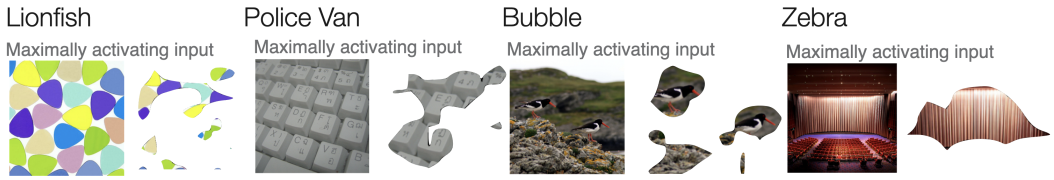

Lionfish
Maximally activating input
Police Van
Maximally activating input
Bubble
Maximally activating input
Zebra
Maximally activating input

Figure 4: Results on ResNet50. Visualization of the inputs with the largest projection value on the concept vectors for the classes *lionfish, police van, bubble and zebra*. Next to the input images we visualize the concept segmentation masks.

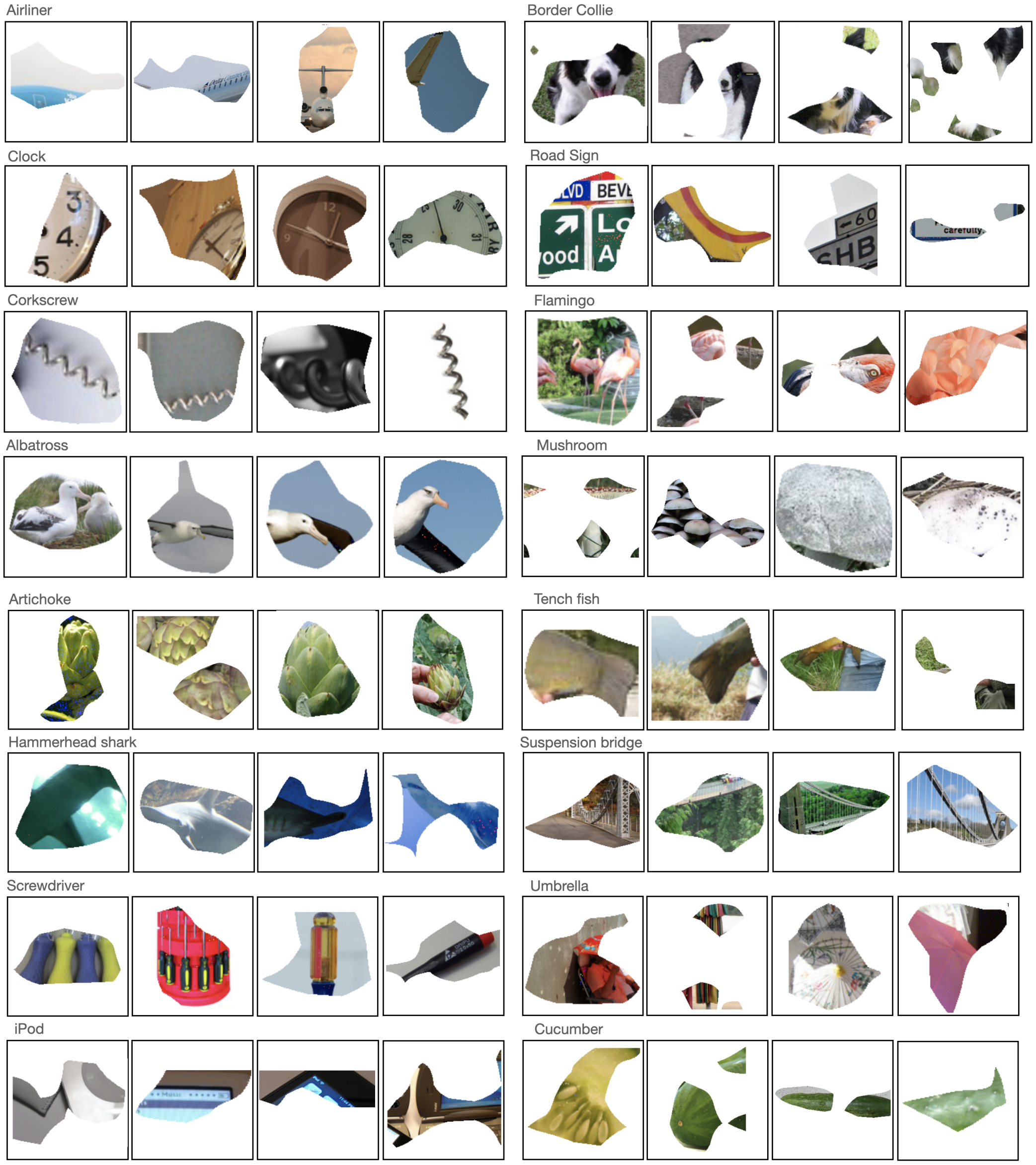


Figure 5: IV3 concepts for additional classes. We show the resized segmentation masks.

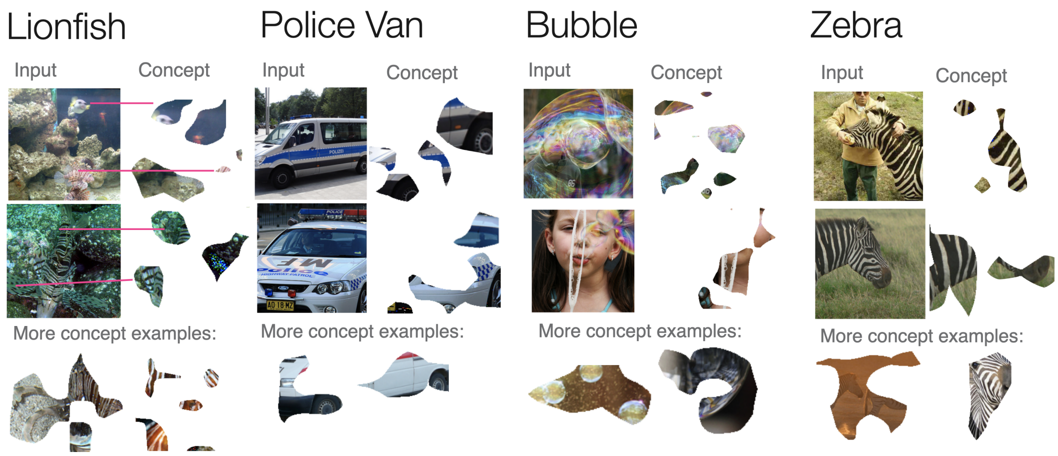
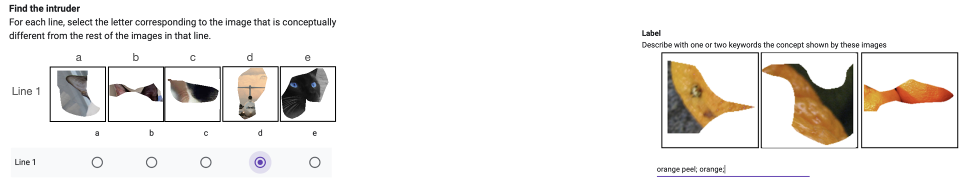


Figure 6: Results on RestNet50. Segmentation masks of the concept vectors in *layer4.add* of ResNet50 for the same classes and input images in Figure 1.

D Additional Evaluation Results

Figure 7 shows two examples proposed during the user study to familiarize the users with the questions. Some of the labels proposed in part (ii) of the studyare shown in Figure 8 in the form of wordclouds. Finally, Figure 9 shows the SDC values for the 1000 ImageNet classes.



Find the intruder

For each line, select the letter corresponding to the image that is conceptually different from the rest of the images in that line.

	a	b	c	d	e
Line 1	○	○	○	●	○

Label

Describe with one or two keywords the concept shown by these images

orange peel; orange;

(a) Task example in the user study part (i) on concept coherency.

(b) Task example in the user study part (ii) on concept understandability.

Figure 7: Examples proposed in the user study to familiarize the participants with the questions.

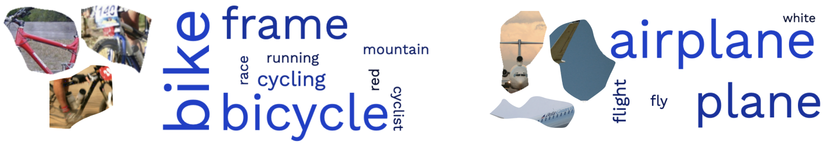


Figure 8: Labels proposed by the users for the visualization of the main concepts for the classes *airliner* and *mountain bike*. The word font indicates the frequency of each word in the user responses.

E Additional Dataset Exploration Results

Figure 10 shows some validation inputs that were flagged by our method for further inspection. The first image shows style shift from real photos to drawings. The second and third images present multiple objects belonging to different categories.

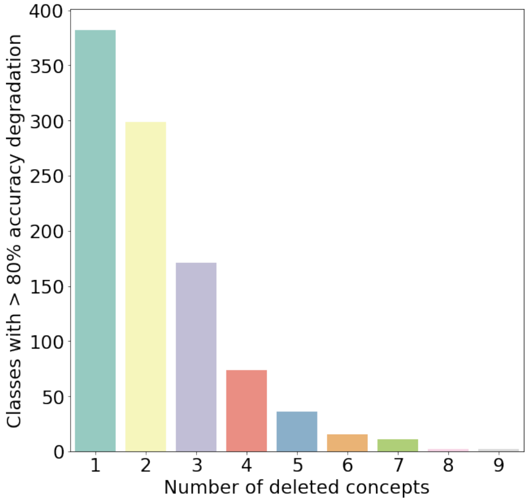
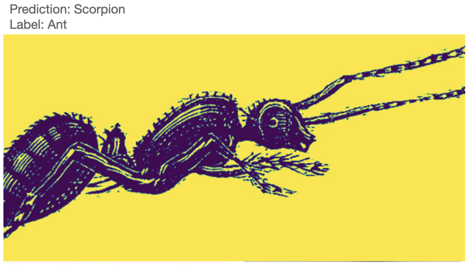
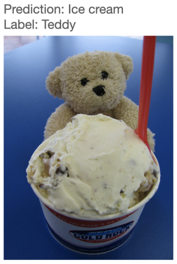


Figure 9: Input space removal





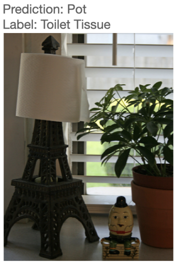


Figure 10: Validation inputs flagged for further inspection