Prototype Generation: Robust Feature Visualisation for Data Independent Interpretability

Arush Tagade, Jessica Rumbelow
Leap Laboratories
{arush,jessica}@leap-labs.com

Abstract

We introduce *Prototype Generation*, a stricter and more robust form of feature visualisation for model-agnostic, data-independent interpretability of image classification models. We demonstrate its ability to generate inputs that result in natural activation paths, countering previous claims that feature visualisation algorithms are untrustworthy due to the unnatural internal activations. We substantiate these claims by quantitatively measuring similarity between the internal activations of our generated prototypes and natural images. We also demonstrate how the interpretation of generated prototypes yields important insights, highlighting spurious correlations and biases learned by models which quantitative methods over test-sets cannot identify.

1 Introduction

Interpretability techniques have become crucial in the era of increasingly powerful artificial intelligence (AI) systems [OpenAI, 2023, Anthropic, 2021, Anil et al., 2023, Touvron et al., 2023]. As AI models continue to outperform human benchmarks across numerous tasks and domains, the importance of understanding their decision-making processes has never been more pressing. Apart from safety concern in high-stakes domains such as healthcare[Bohr and Memarzadeh, 2020] and autonomous vehicles[Elallid et al., 2022], EU law requires certain AI systems to comply with the 'right to explanation' [Goodman and Flaxman, 2017] making interpretability crucial for business operations.

Over the past decade, various methods have been developed to improve human understanding of complex AI models. Techniques such as LIME [Ribeiro et al., 2016], SHAP [Lundberg and Lee, 2017], CAM [Oquab et al., 2015] and Network Dissection [Bau et al., 2017, Zhou et al., 2018] have targeted local interpretability, offering explanations of model decisions for individual data points. However, these techniques cannot provide a global understanding of *what a model has learned overall*, which is necessary for comprehensive analysis and trust in automated systems.

In this work, we focus on feature visualisation [Olah et al., 2017] as a powerful interpretability tool able to extract such holistic insights from arbitrary neural networks. Despite its promise, feature visualisations have not been without criticism. Past research has pointed out the disparity between internal processing of feature visualisations as compared to other natural images [Geirhos et al., 2023] by observing path similarity. We discuss these criticisms further in Section 2.

Addressing these limitations, we introduce *Prototype Generation* in Section 3, a robust visualisation tool that not only contains determining features for any given class but also maintains equal or better path similarity with natural images. Our experiments using Resnet-18[He et al., 2015] and InceptionV1[Szegedy et al., 2014] show that prototypes generated using our method are highly similar to natural images in terms of internal path activations.

ATTRIB: Workshop on Attributing Model Behavior at Scale at NeurIPS 2023.

Understanding the model at a global level helps in identifying systemic biases, uncovering spurious correlations, and potentially refining the model for better performance and fairness. We use prototype generation to discover undesired correlations and identify failure modes on unseen data in Section 4, demonstrating how our method provides *data-independent* insights. Our method helps us directly attribute model behaviour and performance to dataset characteristics. Through this, our contribution serves the broader goal of enhancing global data-independent interpretability in deep learning models, thereby making them more transparent, accountable, and trustworthy.

2 Related Work

Feature visualisation is a method to extract information from a model about what it has learned [Molnar, 2021, Mordvintsev et al., 2015, Erhan et al., 2009, Olah et al., 2017, Nguyen et al., 2016]. Unlike local interpretability methods that focus on individual predictions, feature visualisation is a *global* interpretability method that aims to visualise the features that different neurons inside a model have learned to respond to. Observing feature visualisations to understand model behaviour is a data-independent approach to interpretability, allowing for qualitative assessment of a model's internal logic irrespective of any test dataset – and so, can be used to find failure modes irrespective of whether examples of those failures exist in a test set. This technique works by generating an input $\hat{x}$ that maximises a chosen output logit or internal activation (in this case, output logit c with respect to model h): $\hat{x} = \arg\max_x h_c(x)$. Feature visualisation has been used for a number of purposes, such as identifying specialised circuits within neural networks [Olah et al., 2020] and understanding the learned features of individual nodes or groups of nodes [Olah et al., 2018].

Despite its utility, feature visualisation is not without its detractors. One prominent line of criticism comes from Geirhos et al. [2023], arguing that the visualisations may not truly represent what the model has learned, and so cannot be reliably used to predict its behaviour on unseen data in the future. These criticisms are substantiated by experiments that manipulate feature visualisations to produce misleading or contradictory representations without changing the model's decision-making process. They also introduce the path similarity metric to quantify this. This metric measures the similarity between internal activation 'paths' caused by two different inputs across the layers of a neural network. If two inputs excite similar neurons, this leads to a high path similarity between these two inputs. The measure of similarity chosen by Geirhos et al. [2023] is Spearman's rank order correlation (referred to as spearman similarity (SS) in the rest of this paper). This path similarity metric is used to show the disparity between internal activations in response to natural images versus feature visualisations of the same class.

In the sections that follow, we show that feature visualisations of a specific kind – *prototypes* – generated using our method contain key features for the class they represent, and maintain a consistent path similarity with natural images. By doing so, we overcome some of the limitations previously associated with feature visualisation.

3 Prototype Generation

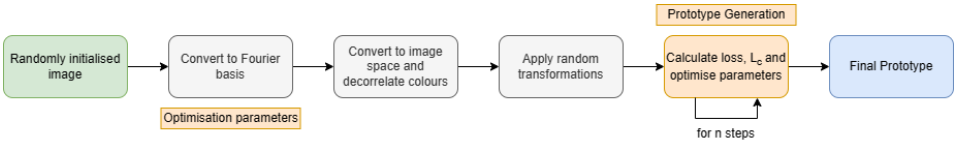
For a given model M, we define a prototype P as an input that maximally activates the logit corresponding to c, while keeping the model's internal activations in response to that input close to the distribution of 'natural' inputs. Let $\mathbb{I}$ represent the set of all possible natural inputs that would be classified by model M as belonging to class c. We aim to generate a prototype P such that it aggregates the representative features of a majority of inputs in $\mathbb{I}$. Formally, we posit that the activations $\mathbf{A}_P$ of P are 'closer' to the mean activations $\mathbf{A}_{\mathbb{I}}$ of all $I \in \mathbb{I}$ than any individual natural image I across all layers $\mathbb{L}$ in M. We measure 'closeness' between $\mathbf{A}_P$ and $\mathbf{A}_{\mathbb{I}}$ using spearman correlation.

We use spearman similarity as per Geirhos et al. [2023] to allow for direct comparison of our methods with their published work. Denoting spearman similarity as SS, our formal assertion is that:

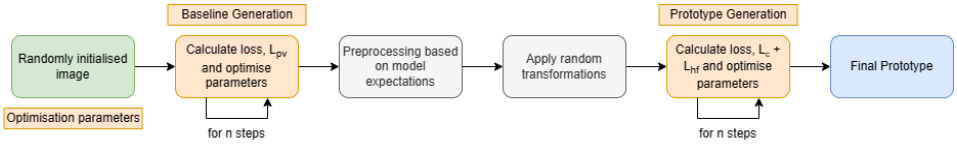
$$SS(\mathbf{A}_{\mathbb{I}_l}, \mathbf{A}_{\boldsymbol{P}_{c_l}}) \geq SS(\mathbf{A}_{\mathbb{I}_l}, \mathbf{A}_{\boldsymbol{I}_l}), \forall l \in L, \forall \boldsymbol{I} \in \mathbb{I} \tag{1}$$

If this conditions is satisfied, we can confidently assert that prototype P shows prototypical qualities of the class c, and contains features representative of the model's understanding of that class.

3.1 Our Method



(a) Flowchart for methodology used by Olah et al. [2017]



(b) Flowchart of our method

Figure 1: Comparison between our method and feature visualisation method proposed by Olah et al. [2017]

Existing feature visualisation methods aim to generate an input that maximally activates a selected neuron inside a neural network. Prototype generation is similarly a technique that generates an input, but with the objective of maximally activating a selected output logit (rather than an internal activation). This positions prototype generation as a specialised form of feature visualisation, distinguished by its focus on class-specific logits rather than internal activations. Our approach differs from the existing feature visualisation methodology in a number of ways, as shown in Figure 1. We compare our implementation with the publicly available *Lucent* [Greentfrapp, 2018] library – the PyTorch implementation of the methodology proposed by Olah et al. [2017].

Both implementations begin with a randomly initialised image. Lucent converts this randomly initialised image to the Fourier basis, but we find (as shown later) that this causes the resulting feature visualisations to be unrepresentative of natural class features. In contrast, we do not optimise in the Fourier basis, instead optimising the input pixels directly. We first optimise to minimise what we call probability variance loss, L_{pv} to generate a baseline input. This loss ensures that the output logits for our input image are balanced i.e. the input image has roughly an equal chance of being predicted to be a member of any class. Our preprocessing steps vary depending on the model's expectations; for models trained on ImageNet, this involves a normalisation shift using the mean and standard deviation of the ImageNet training set – of whatever preprocessing the model expects for inference. Additionally we apply random affine transformations to constrain the optimisation process and discourage the generation of out-of-distribution (OOD) adversarial inputs – we further discuss the effect of these transformations in Appendix A. Lucent uses similar random transformations, but does not tune them for path similarity. The difference in the resultant prototypes for *Lucent* and our method can be seen clearly by comparing Figures 2 and 3.

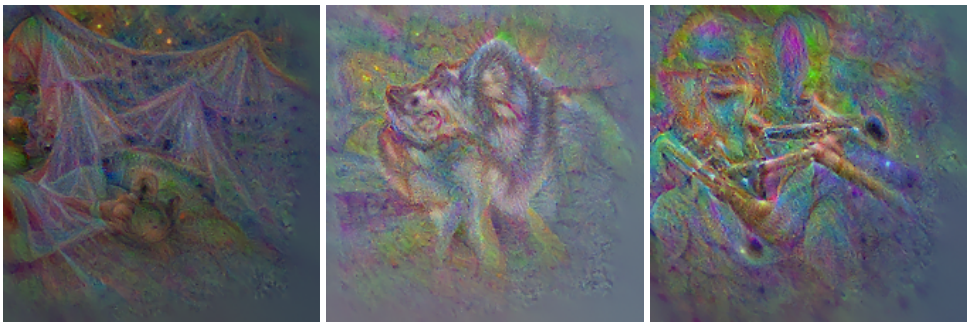
We define two losses: L_c, the negative of the logit for the desired class; and L_{hf}, the high-frequency loss that penalises large differences between adjacent pixel values. We use both L_c and L_{hf} to define our combined loss whereas Olah et al. [2017] employ only L_c.

3.2 Experiments

We assess the prototypes generated by observing how closely the prototype's activations mirror the average activations of natural images in the same class. We quantify closeness between activations by calculating spearman similarity as defined in Section 2. Appendix A contains information about hyperparameters and other implementation details.

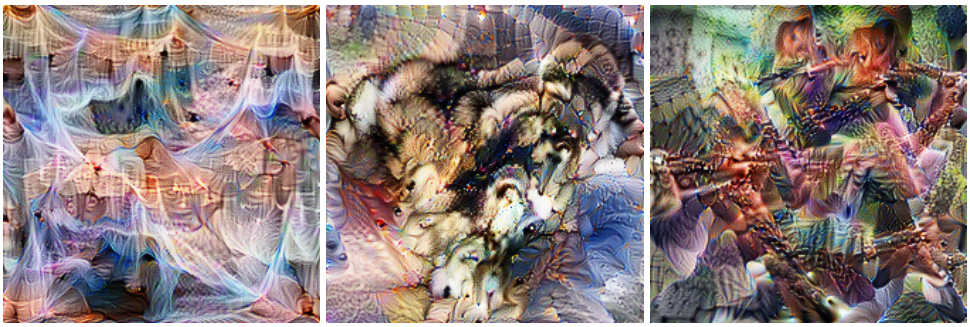
Path similarity per layer. To characterise the path similarity of our generated prototypes, we select 11 random classes from the ImageNet dataset.

We approximate $A_{\mathbb{I}}$ by averaging the activations of 100 randomly selected images from each of the 11 classes. For each class c, we generate a prototype P, capture its activations A_P, and also capture activations from the individual images in $\mathbb{I}$. To alow for direct comparison of our results with those reported by Geirhos et al. [2023], we also select 100 random images from other classes and capture



(a) Mosquito Net (0.61) (b) Alaskan malamute (0.53) (c) Flute (0.54)

Figure 2: Example prototypes generated by our method for the ImageNet classes Mosquito Net, Alaskan malamute and Flute with their average spearman similarity across all layers of Resnet-18 denoted in brackets



(a) Mosquito Net (0.33) (b) Alaskan Malamute (0.22) (c) Flute (0.31)

Figure 3: Example prototypes generated by Olah et al. [2017]'s method for the ImageNet classes Mosquito Net, Alaskan malamute and Flute with their average spearman similarity across all layers of Resnet-18 denoted in brackets

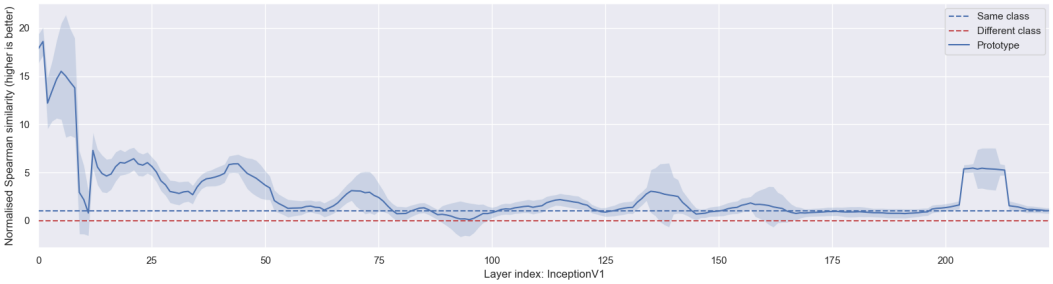
their activations denoted by $A_{\boldsymbol{I}_{dc}}$. Our raw results consist of three sets of spearman similarity scores between,

- Approximated $A_{\mathbb{I}}$ and A_P
- Approximated $A_{\mathbb{I}}$ and $A_{\boldsymbol{I}}$, averaged across all $\boldsymbol{I} \in \mathbb{I}$
- Approximated $A_{\mathbb{I}}$ and $A_{\boldsymbol{I}_{dc}}$, averaged across all $\boldsymbol{I}_{dc} \in \mathbb{I}_{dc}$

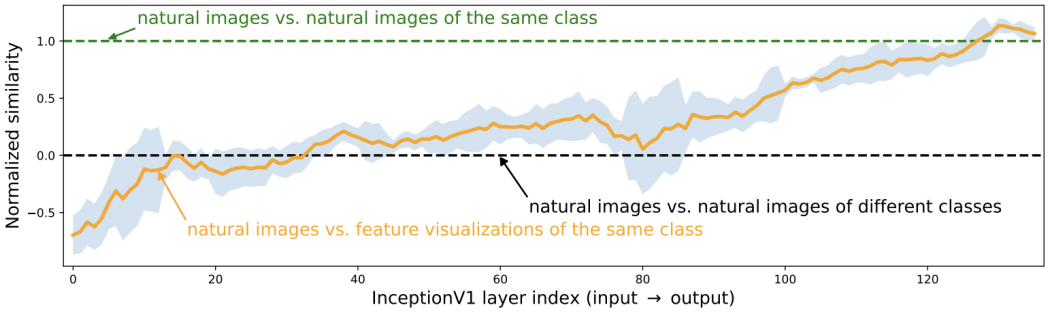
This raw data is normalised such that 1 corresponds to the spearman similarity obtained by comparing natural images of the same class, and 0 corresponds to the spearman similarity obtained by comparing images of one class against images of different classes. Figure 4 shows the normalised path similarity obtained by making the above comparisons. Our experimental results show that these prototypes support our formal assertion for Spearman similarity specified in Equation 1 holds for 147/224 i.e 65.6% of all layers in InceptionV1 averaged across the same 11 classes.

4 Prototype Insights

Since our method generates prototypes that have high path similarity with natural images, 3 we might expect to be able to better understand *what* models have learned about given classes simply by observing their generated prototypes for those classes. Here follows a case study to test whether information present in our prototypes can reliably predict model behaviour on unseen data. We focus on prototypes for two Imagenet classes: the academic gown prototype and the mortarboard prototype, generated by Resnet-18, as shown in Figure 5.

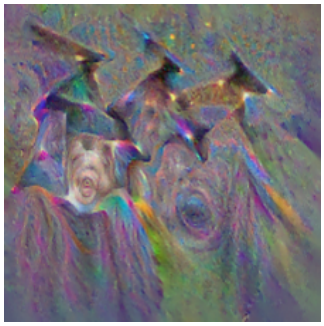


(a) InceptionV1

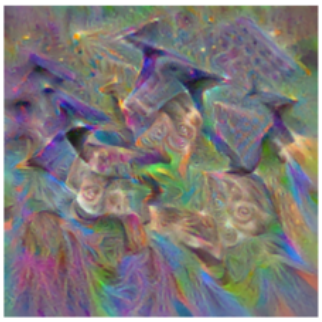


(b) InceptionV1, results from Geirhos et al. [2023].

Figure 4: **Comparison of Spearman Similarity**. Spearman similarities are normalised such that 1 corresponds to the spearman similarity between natural image activations of the same class and 0 corresponds to the spearman similarity between natural image activations of different classes. Here we show examples on two different networks, and for comparative purposes also provide the results of the same experiment from Geirhos et al. [2023]. Note that our method produces super-normal results, with early layer activations from prototypes being closer to the mean natural activation than any natural input.



(a) Academic Gown Prototype



(b) Mortarboard Prototype

Figure 5: Example prototypes from Resnet-18

Hypothesis 1: Resnet-18 will have higher accuracy classifying lighter skinned people wearing academic gowns than darker skinned people wearing academic gowns.

This hypothesis emerges from the observation that the academic gown prototype in Figure 5a shows a lighter-skinned face prominently. We test this hypothesis by observing the performance of Resnet-18 on two different sets of 50 images, one containing lighter skinned people wearing academic gowns and the other containing darker skinned people wearing academic gowns. As shown in Table 1, lighter-skinned people wearing academic gowns are more likely to be classified as the academic gown class than darker-skinned people.

Table 1: Comparison of Resnet-18 performance on light and dark skinned people wearing academic gowns along with the probability of prediction for academic gowns(AG)

	Accuracy	**Probability(AG)**
Lighter-skinned people	72.5%	0.62
Darker-skinned people	60%	0.54

Hypothesis 2: Resnet-18 is likely to misclassify images of mortarboards as academic gowns, if the images have a mortarboard and a face in them.

By observing differences in the mortarboard and academic gown prototypes, we see that the mortarboard prototype has a much weaker representation of a face, compared to the academic prototype. This leads us to hypothesise that an image containing both a mortarboard and a face is likely to be misclassified as an academic gown. To test this hypothesis we again observe the performance of Resnet-18 on a set of 50 images containing mortarboards with no faces and a set of images containing mortarboards and a face. We ensure that the mortarboards with faces have no presence of academic gowns. Results were again as expected, Table 2 shows that mortarboards with faces are more likely to be misclassified as academic gowns.

Table 2: Comparison of Resnet-18 performance on mortarboards with and without faces along with the probability of prediction for mortarboards(MB) and academic gowns(AG)

	Accuracy	**Probability(MB)**	**Probability(AG)**
Mortarboard without face	92.3%	0.77	0.05
Mortarboard with face	70.5%	0.67	0.25

5 Discussion

In both cases presented in Section 4, we didn't have to comb through the entire dataset but rather observing the prototypes provided us with a data-independent way of understanding Resnet-18's behaviour on unseen data. While metrics on a test set can provide a broad overview of a model's performance *with respect to that test set*, they often don't provide the granularity needed to understand *why* a model might be likely to fail on future, unseen data. This also helps us compare quantitative model performance to qualitative features seen in generated prototypes. Prototypes can meaningfully augment existing evaluation metrics in a number of ways:

- **Identifying dataset bias.** If a model shows bias, as in the lighter-skinned academic gown prototype, this points at bias in the data. Armed with this knowledge, the dataset can be modified or augmented to remove this bias, and the model retrained to improve performance on underrepresented classes or features.
- **Spotting spurious correlations.** By comparing prototypes for closely related classes, one can discern which features are given undue importance, enabling deeper understanding of model failures due to the presence of potentially misleading features.
- **Rapid Iteration.** Model developers can generate prototypes during training, spotting issues like biases or potential misclassifications early in the process. These insights also enable targeted data augmentation, necessitating the collection and preprocessing of data samples specific to correcting a problem in the model, rather than just throwing more (potentially also biased) data at the problem. This means more rapid iteration and correction, saving both time and resources.

Furthermore, if a model is underperforming on a specific class and the reason is not immediately clear from the validation data, generating a prototype can shed light on the shortcomings in its learning. This proactive approach facilitates the identification of what the model has learned or perhaps, more importantly, what it has failed to learn. Moreover, interpretability techniques of this kind make *knowledge discovery* possible – that is, if we are able to train a model to perform a task that humans cannot, we can use interpretability to understand what patterns it has identified in its training data that we are unaware of, and thereby gain new insights about that data.

References

OpenAI. GPT-4 Technical Report, March 2023. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

Anthropic. Model card and evaluations for claude models, 2021.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. PaLM 2 Technical Report, September 2023. URL http://arxiv.org/abs/2305.10403. arXiv:2305.10403 [cs].

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023. URL http://arxiv.org/abs/2307.09288. arXiv:2307.09288 [cs].

Adam Bohr and Kaveh Memarzadeh. The rise of artificial intelligence in healthcare applications. *Artificial Intelligence in Healthcare*, pages 25–60, 2020. doi: 10.1016/B978-0-12-818438-7.00002-2. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7325854/.

Badr Ben Elallid, Nabil Benamar, Abdelhakim Senhaji Hafid, Tajjeeddine Rachidi, and Nabil Mrani. A Comprehensive Survey on the Application of Deep and Reinforcement Learning Approaches in Autonomous Driving. *Journal of King Saud University - Computer and Information Sciences*, 34(9):7366–7390, October 2022. ISSN 1319-1578. doi: 10.1016/j.jksuci.2022.03.013. URL https://www.sciencedirect.com/science/article/pii/S1319157822000970.

Bryce Goodman and Seth Flaxman. European Union Regulations on Algorithmic Decision Making and a "Right to Explanation". *AI Magazine*, 38(3):50–57, 2017. ISSN 2371-9621. doi: 10.1609/aimag.v38i3.2741. URL https://onlinelibrary.wiley.com/doi/abs/10.1609/aimag.v38i3.2741. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1609/aimag.v38i3.2741.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, August 2016. URL http://arxiv.org/abs/1602.04938. arXiv:1602.04938 [cs, stat] version: 3.

Scott Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions, November 2017. URL http://arxiv.org/abs/1705.07874. arXiv:1705.07874 [cs, stat] version: 2.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is Object Localization for Free? - Weakly-Supervised Learning With Convolutional Neural Networks, 2015. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Oquab_Is_Object_Localization_2015_CVPR_paper.html.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations, April 2017. URL http://arxiv.org/abs/1704.05796. arXiv:1704.05796 [cs].

Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting Deep Visual Representations via Network Dissection, June 2018. URL http://arxiv.org/abs/1711.05611. arXiv:1711.05611 [cs] version: 2.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, 2(11):e7, November 2017. ISSN 2476-0757. doi: 10.23915/distill.00007. URL https://distill.pub/2017/feature-visualization.

Robert Geirhos, Roland S. Zimmermann, Blair Bilodeau, Wieland Brendel, and Been Kim. Don't trust your eyes: on the (un)reliability of feature visualizations, July 2023. URL http://arxiv.org/abs/2306.04719. arXiv:2306.04719 [cs, q-bio].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

C Molnar. 10.1 learned features. https://christophm.github.io/interpretable-ml-book/cnn-features.html, 2021. Accessed: 2021-12-21.

A Mordvintsev, C Olah, and M Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog (2015)*, 2015.

D. Erhan, Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Visualizing Higher-Layer Features of a Deep Network, 2009. URL https://www.semanticscholar.org/paper/Visualizing-Higher-Layer-Features-of-a-Deep-Network-Erhan-Bengio/65d994fb778a8d9e0f632659fb33a082949a50d3.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks, May 2016. URL http://arxiv.org/abs/1602.03616. arXiv:1602.03616 [cs].

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 5(3):e00024.001, March 2020. ISSN 2476-0757. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The Building Blocks of Interpretability. *Distill*, 3(3):e10, March 2018. ISSN 2476-0757. doi: 10.23915/distill.00010. URL https://distill.pub/2018/building-blocks.

Greentfrapp. Lucent, 2018. URL https://github.com/greentfrapp/lucent. original-date: 2020-05-09T18:07:01Z.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, April 2015. URL http://arxiv.org/abs/1412.1897. arXiv:1412.1897 [cs].

A Experimental Details

Choice of classes. The 11 randomly chosen classes for our experiment in terms of Imagenet class indices were [12, 34, 249, 429, 558, 640, 669, 694, 705, 760, 786].

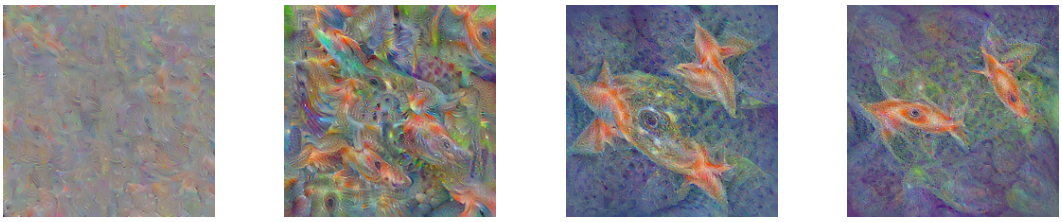
Optimisation parameters. The randomly initialised image is initialised using torch.rand(3, 224, 224) and the optimisation process uses the Adam optimiser with a constant learning rate of 0.05 over 512 optimisation steps.

Selecting regularisation parameters. Generating a feature visualisation that maximises a certain output logit related to class c without any constraints will result in adversarial images that have no representative features of c [Erhan et al., 2009]. To guide the optimisation process towards creating feature visualisations that maintain the natural structure of images belonging to c we need to utilise some constraints in the form of regularisation. The concept of regularisation originates from the broader field of optimisation and is crucial for preventing overfitting.

During the prototype generation process we need to be cautious while implementing regularisation. An unprincipled approach can lead to undesirable outcomes, such as creating visualisations that are aesthetically pleasing but lack meaningful information about what the model has truly learned during training. In contrast, foregoing regularisation may result in visualisations that contain high-frequency noise or adversarial inputs, which are not only hard to interpret but also unlikely to yield useful insights. This brings us to the delicate balance we aim to strike: optimising the feature visualisation in such a way that it remains useful and informative while avoiding misleading or uninterpretable outputs. Figure 6 shows the increasing effect of regularisation on generated prototypes, starting from no regularisation at the left and increasing towards the right, for prototypes generated using Resnet-18 for the goldfish class.

We find that path similarity can work as a reliable measure of features that conform to the natural structure found in images belonging to a certain class in the training set. Regularisation can take many forms [Nguyen et al., 2015, Mordvintsev et al., 2015] but our focus here will be on two specific forms of regularisation: high-frequency penalisation and random affine transformations.

High-frequency penalisation aims to suppress unnecessary details and noise, facilitating a cleaner, more interpretable visualisation. Transformation robustness, on the other hand, ensures that minor alterations in the input space do not result in significantly different visualisations, thus maintaining consistency and reliability. We perform high-frequency penalisation to minimise large variation in magnitude of adjacent pixel values across all channels and apply it by adding the loss L_{hf} described earlier in Section 3 to our overall loss. We also ensure transformation robustness using random affine transforms made up of random scale, random translate and random rotation transforms. The parameters of random scale, random translate and random rotation guide the degree of freedom of the random affine transform and lead to varying visual characteristics of our generated prototype.



(a) (b) (c) (d)

Figure 6: Effect of increasing regularisation (from left to right) for the goldfish prototype

To find regularisation parameters that lead to reliable prototypes, we test the effect of random affine transforms on the spearman similarity between prototypes generated using a particular set of regularisations, and average natural image activations from the training set. We test a set of 125 random affine transforms with varying parameters for random scale, random translate and random rotation transforms,

- Random scale: None, (0.9, 1.1), (0.8, 1.2), (0.7, 1.3), (0.5, 1.5)
- Random rotation: None, 30, 60, 90, 180

- Random translate: None, (0.05, 0.05), (0.1, 0.1), (0.2, 0.2), (0.5, 0.5)

Table 3 shows the top 5 regularisation parameters we discovered, in terms of path similarity calculated using spearman similarity between the A_P (created using these regularisation parameters) and $A_{\mathbb{I}}$, where $\mathbb{I}$ contains 100 natural images belonging to the goldfish class, averaged over all the layers of our model of choice, Resnet-18. The prototype generated using the best regularisation can be seen in Figure 6d.

Table 3: Spearman similarity of top five affine transformation protocols with respect to average natural image activations.

Scale	Rotation	Translate	Average path similarity
(0.7, 1.3)	180	(0.5, 0.5)	0.564
(0.5, 1.5)	30	(0.5, 0.5)	0.563
(0.8, 1.2)	30	(0.5, 0.5)	0.562
(0.5, 1.5)	60	(0.1, 0.1)	0.561
(0.7, 1.3)	30	(0.5, 0.5)	0.560

We choose the regularisation parameters that lead to the best average path similarity with the parameters of scale, rotations and translation set at (0.7, 1.3), 180 and (0.5, 0.5) respectively.

B More Experimental Results

Figure 4 show the normalised plots for our experimental results Spearman Similarity of our generated prototypes with natural images. In this section, we want to show normalised spearman similarity results for Resnet-18 in Figure 7 and show the raw results before normalisation for our path similarity experiments. Figure 8 shows how close the similarity values are to each other for most part of the network before diverging heavily at the end of the network.

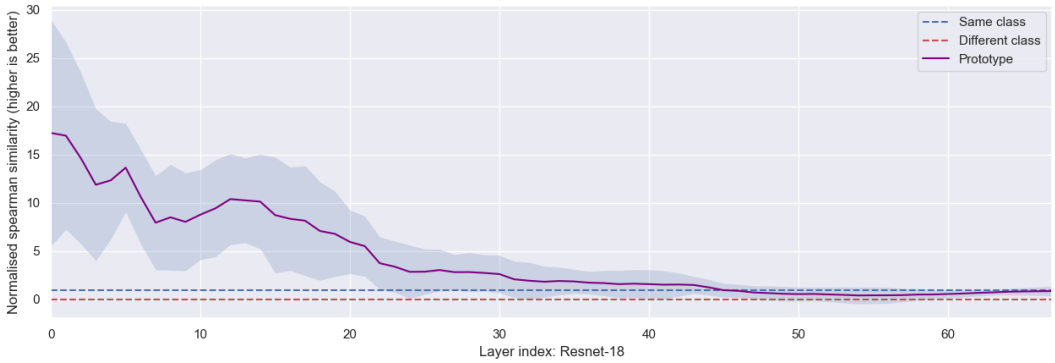
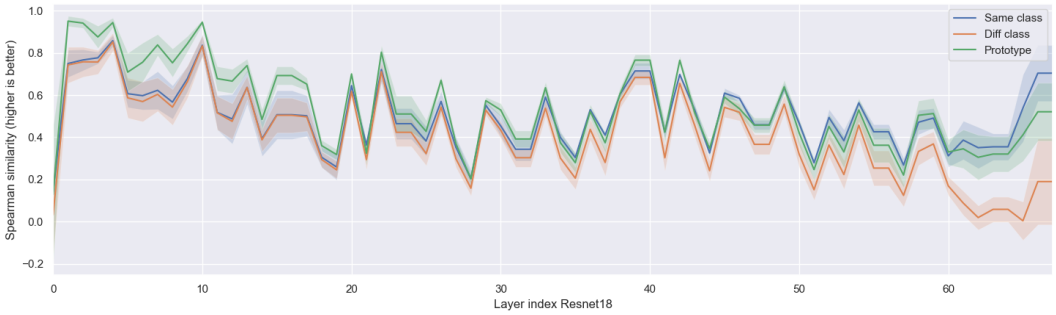


Figure 7: Comparison of normalised spearman similarity for prototypes generated from Resnet-18

Using our raw results, we also quantify the mean spearman similarity across all layers of Resnet-18 and InceptionV1 in Table 4. We can see that the average spearman similarity of our generated prototypes is higher than other natural images belonging to the same class on average, for both Resnet-18 and InceptionV1.



(a) Resnet-18

(b) InceptionV1 (Googlenet)

Figure 8: Raw value plots for spearman similarity comparisons

Table 4: Comparison of average spearman similarity

	Average spearman similarity
Prototype P	0.54 ± 0.06
Same class images $\mathbb{I}_c$	0.50 ± 0.05
Diff class images $\mathbb{I}_{dc}$	0.41 ± 0.06

(a) Resnet-18

	Average spearman similarity
Prototype P	0.56 ± 0.07
Same class images $\mathbb{I}_c$	0.50 ± 0.05
Diff class images $\mathbb{I}_{dc}$	0.40 ± 0.06

(b) InceptionV1