Interactive Mars Image Content-Based Search with Interpretable Machine Learning

Bhavan Vasu[1, 2], **Steven Lu**[1], **Emily Dunkel**[1], **Kiri L. Wagstaff**[1, 2]
Kevin Grimes[1], **Michael McAuley**[1]

[1] Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109-8099, USA
[2] Oregon State University, Corvallis, OR 97331, USA
vasub@oregonstate.edu, {you.lu, emily.dunkel, kevin.m.grimes, michael.mcauley}@jpl.nasa.gov, wkiri@wkiri.com

Abstract

The NASA Planetary Data System (PDS) hosts millions of images of planets, moons, and other bodies collected throughout many missions. The ever-expanding nature of data and user engagement demands an interpretable content classification system to support scientific discovery and individual curiosity. In this paper, we leverage a prototype-based architecture to enable users to understand and validate the evidence used by a classifier trained on images from the Mars Science Laboratory (MSL) Curiosity rover mission. In addition to providing explanations, we investigate the diversity and correctness of evidence used by the content-based classifier. The work presented in this paper will be deployed on the PDS Image Atlas, replacing its non-interpretable counterpart.

Introduction

The PDS Cartography and Imaging Sciences Node is the curator of NASA's primary digital image collection spanning past, present, and future planetary missions. The PDS Imaging Node provides access to this data archive via the PDS Image Atlas. Due to the data archive's ever-growing nature, manually searching through tens of millions of images to find data products of interest is infeasible. With an ever-growing list of missions, future releases of the Atlas will accommodate a reliable, interpretable content-based classification system that aims to provide a three-fold benefit.

Firstly, an interpretable content-based classification system will validate the evidence used by the content-based classifier and ensure the right visual cues are being used by the classifier. Secondly, such an interpretable model will help bridge the gap between the mental model of Atlas users and planetary image classifiers. Finally, identifying erroneous evidence through user adjudications can establish a feedback loop from Atlas users to data scientists regarding the quality of evidence used by the planetary image classifiers. Users searching Atlas will have the ability to interactively identify relevant images. Establishing such a feedback loop is vital for improving classifier performance while enabling users to play an active role, increasing user engagement and understanding.

In this paper, we employ explanations from case-based reasoning approaches that identify the evidence from the training set used to classify a test image.

We demonstrate an interpretable content-based image search system by leveraging the prototypical architecture proposed by Chen et al. (2019). In addition to building upon the existing work, we extend it by evaluating the diversity and correctness of prototypes resulting from a classifier trained on imagery from Mars. Based on observations we made during the evaluation, our contribution was the incorporation of a diversity-enhancing term to the original work by Chen et al. (2019), which notably amplified the diversity of evidence utilized and subsequently enhanced performance. We describe our plan to deploy the system by replacing the non-interpretable counterpart currently hosted on the Atlas with the interpretable content-based classifier proposed in this paper. The MSL surface dataset [1] used in this paper was published by Wagstaff et al. (2021).

Related Work

Interpretable content-based classification has appeared in the literature multiple times based on the classifier under investigation for content classification (Nauck and Kruse 1999; Rui et al. 1998; Vasu et al. 2021). Further improvements in the wider field of interpreting machine learning decisions were achieved with the introduction of increasingly complicated and opaque classifiers. Explanations generally appear under different taxonomies such as white-box vs. black-box, inherently interpretable vs. post-hoc, and neuron vs. primary vs. layer attribution methods (Lucas et al. 2022). White vs. black-box categorizes methods based on whether they leverage internal classifier structure to generate explanations. Inherently interpretable vs. post-hoc categorizes them based on the model naturally providing explanations vs. using an explanation method after model development.

The introduction of deep models sacrificed interpretability in favor of improved performance and automatic feature extraction. Interpreting deep models has gained traction in recent years due to the large-scale deployment of deep models. One of the first works to interpret deep convolution neural network (CNN), proposed by Zeiler and Fergus (2014), deals with understanding activations and the internal operations of CNNs by looking for patches that maximize a neu-

[1] https://zenodo.org/record/4033453

ron activation. Zhou et al. (2016) leveraged the presence of a global average pooling layer to backtrack and combine activations with the strongest connection to a particular class to produce local explanations in the form of saliency maps.

Several works were also inspired by the gradient-based approach known as GradCAM proposed by Selvaraju et al. (2017). More recently Khorram, Lawson, and Fuxin (2021) used integrated gradient and an optimization paradigm to obtain explanations. Interpretability can also come in the form of attention mechanisms. Zheng et al. (2017); Zhang et al. (2014); Petsiuk, Das, and Saenko (2018) offer some insight into regions of the image attended, indicating the "where" but fails to address "why" a region of the image was paid attention to.

Despite the vast body of work in post-hoc explanations (Selvaraju et al. 2017; Petsiuk, Das, and Saenko 2018; Zheng et al. 2017), we primarily focus on leveraging an inherently interpretable deep model such as ProtoPNet (Chen et al. 2019) due to shortcomings of post-hoc approaches highlighted by Adebayo et al. (2018), Rudin (2019), and Lakkaraju and Bastani (2020). Methods by Selvaraju et al. (2017) are unreliable in the presence of repetitive patterns spread across a larger portion of the image, such as rocks or terrain on Mars. More importantly, we chose ProtoPNet because we expected its explanations to be more intuitive for users of the Atlas who may not be ML experts. Case-based reasoning work by Kolodner (1992) uses previously created prototypes, limiting adaptivity, while the work by Aamodt and Plaza (1994) enhances adaptability through a four-phase approach involving both past and new prototypes. In contrast, the Prototypical Part Network (ProtoPNet) proposed by Chen et al. (2019) creates new prototypes by optimizing network weights, focusing on minimizing the distance between prototypes and class instances in a high-dimensional space. This paper leverages the architecture and training routine proposed by Chen et al. (2019). We quantify the evidence learned by the deep models in terms of diversity and correctness of evidence as opposed to quantifying them through agreement (Bau et al. 2017, 2019) with a region of the image as the latter requires a large volume of fine-grained annotations.

Prototypical Part Network

The ProtoPNet architecture proposed by Chen et al. (2019) uses a standard deep network for feature extraction and introduces a prototypical layer that learns a pre-defined number of prototypes or exemplars that best represent each image class. The fully connected layer after the prototypical layer represents the contribution of each learned prototype to the final decision. Therefore, the network avoids problems such as unreliability due to inaccuracies in understanding a model's decision-making process, described by Rudin (2019) by only allowing information to flow to the final classification layer through the prototypical layer. Once regions of images most similar to the learned prototypes are found through similarity matching, they can be visualized as a bounding box by using a threshold at 95% of maximum similarity. The similarity score represents the strength of the prototype match, while the weights of the fully connected

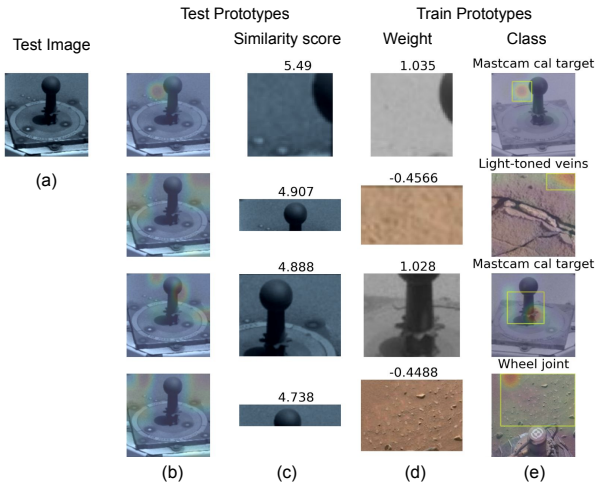

Figure 1: Qualitative example of the top-4 most visually similar prototypes for class *Mastcam cal target* from the MSL surface dataset. Column (a) is the test image, (b) shows the same image overlayed with a heatmap showing regions most activated by the prototype learned during training followed by (c) showing a cropped version of the heatmap after threshold with the similarity score. (d) shows the cropped regions of heatmaps from Column (e). Column (e) shows the training images overlayed with regions obtained after prototypes projection on the training set and . The evidence looks coherent across both training and testing prototypes i.e., column (c) and (d) when the weights are positive.

layer represent its contribution to a class during training. Above all the highlighted benefits, ProtoPNet learns part-based associations with no additional region-based annotation, which allows the developer to define a finer subclass.

In the following sections, we present results for both a VGG19 and ResNet18 backbone on the MSL surface dataset. The MSL surface dataset contains visual features observed on the surface of Mars by the Curiosity rover. Figure 1 (discussed in more detail later) shows an example ProtoPNet output for the MSL surface dataset image belonging to class *Mastcam cal target* used for calibrating the Mastcam instrument on board the rover. We use ten prototypes per class for all experiments in the rest of the paper. We use the weights pre-trained from ImageNet (Russakovsky et al. 2015) data set to initialize all our feature extractor backbones.

Proto-MSLNet

In this section, we elucidate the training process of the content-based classifier utilizing the MSL surface dataset, herein referred to as MSLNet. The currently deployed version of MSLNet on the Atlas classifies imagery received from the MSL Curiosity rover. We plan to replace MSLNet with an interpretable version we call Proto-MSLNet, which is constructed by training a prototypical and fully connected layer on top of an off-the-shelf feature extractor backbone. Due to the benefits of interpretable models outlined in the

earlier sections, we plan to deploy Proto-MSLNet in spite of a slight drop in test accuracy. Based on our observations while using current approaches, we noticed a lack of diversity among prototypes, i.e., prototypes often look visually similar or come from the same training image. Therefore, we modified the ProtoPNet training procedure to incorporate a diversity factor:

$$\begin{aligned} \text{Div} \ &= -\frac{1}{n}\sum_{x=1}^{n} \underset{z}{\text{minimize}} \max(z - p_j - \text{margin}, 0) \\ &\text{subject to } z \in \text{patches}(f(x_i)) \\ &j : p_j \in P_{y_i} \end{aligned}$$

where n represents the total number of images under consideration. The image x_i denotes the i^{th} image in the dataset, and its corresponding feature representation is captured by $f(x_i)$ while patches$(f(x_i))$ constitutes all potential patches derived from $f(x_i)$. p_j represents learned feature prototype corresponding to the class y_i and P_{y_i} is the complete set of prototypes for class y_i. Lastly, the margin term is a predefined threshold to eliminate trivial differences among prototypes. The total loss L used to train Proto-MSLNet is

$$\begin{aligned} L = \underset{\mathbf{P}, w_{conv}}{\text{minimize}} \frac{1}{n}\sum_{i=1}^{n} &\text{CrsEnt}(h \circ g_{\mathbf{P}} \circ f(x_i), y_i) \\ &+\lambda_1 \text{Clst} + \lambda_2 \text{Sep} + \lambda_3 \text{Div} \end{aligned}$$

where the cross entropy loss (CrsEnt) penalizes misclassification on the training data. The clustering cost (Clst) promotes the presence of a latent patch in each training image that is proximate to at least one prototype from its class. Conversely, by minimizing the separation cost (Sep), it is encouraged that every latent patch of a training image remains distant from prototypes that do not belong to its class. While Clst brings together prototypes of the same class and Sep promotes inter-class separation in prototypes, there is no condition promoting intra-class diversity in prototypes. To address this issue, we introduce the diversity cost (Div). Note, λ_1, λ_2, and λ_3 are hyperparameters used to control the influence of each cost term, $g_{\mathbf{P}}$ is the prototype layer, and h is a fully connected layer. To visualize the prototypes as images we project the prototypes onto the training set as shown below:

$$p_j \leftarrow arg \min_{z \in \mathcal{Z}_j} \|z - p_j\|_2$$

Where $\mathcal{Z}_j$ represents the set of all training patches and p_j is the prototypes found while computing L. Similarly, we can also visualize the test prototypes by replacing $\mathcal{Z}_j$ with the test set.

Data Set

The MSL surface dataset was collected by the Mast Camera (Mastcam) and Mars Hand Lens Imager (MAHLI) instruments on the MSL Curiosity rover spanning 19 classes of interest such as Dust Removal Tool (DRT), Sun, Night Sky, Wheels, Wheel joints, and Wheel tracks. We augment the dataset according to the realistic variations for each instrument: the images from the rotatable platform MAHLI are rotated by 90, 180, and 270 degrees with horizontal and vertical flipping, and the images from the fixed platform Mastcam are only horizontally flipped. Figure 2 shows representative images from eight different classes along with the result of data augmentation on an image from class *Wheel* acquired using MAHLI.

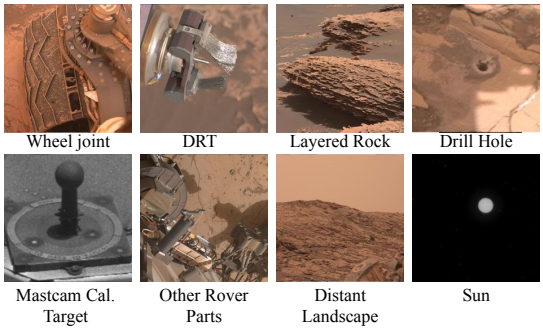

Figure 2: Figure showing representative examples from eight classes of the MSL surface Data Set. DRT refers to the Dust Removal Tool aboard the Curiosity rover.

Experimental Setup

During training, we use a learning rate of 1e-4 for the first 100 epochs followed by a learning rate of 1e-5 for 100 epochs and select the model with the best validation accuracy. Note the training process is split into a gradient update at each epoch and a projection stage every 5 epochs. The hyperparameters λ_1, λ_2, and λ_3 are assigned values of $0.8, 0.08$, and 0.04 respectively. These values were determined through an empirical evaluation across five distinct trials. A more rigorous parameter search will be undertaken in preparation for deployment. We employ a sol-based split as proposed by Wagstaff et al. (2021) to evaluate generalization in a realistic setting, wherein training occurs on past data and validation/testing on future data. The term "sol" here refers to a measure of one Mars day. Note that the sol-based split reveals the temporal label shift between train and validation/test set, reflected in the gap between their accuracy across both deep CNNs reported in Table 1. Wagstaff et al. (2021) provide a full description of the dataset generation process with detailed class distributions. A batch size of 80 was used in combination with an Adam optimizer (Kingma and Ba 2015) to train all the classifiers. All classifiers were trained on a single V100 GPU facilitated by the Texas Advanced Computing Center (TACC).

Results and Analysis

We will discuss our quantitative comparison, followed by a discussion on qualitative results and an analysis of the correctness and diversity of prototypes.

Quantitative Analysis

Table 1 reports train, validation, and test set accuracy on the MSL surface data set with and without a prototypical layer. Note, that the number of images in each dataset split is indicated in the heading. In Table 1, we also report the train,

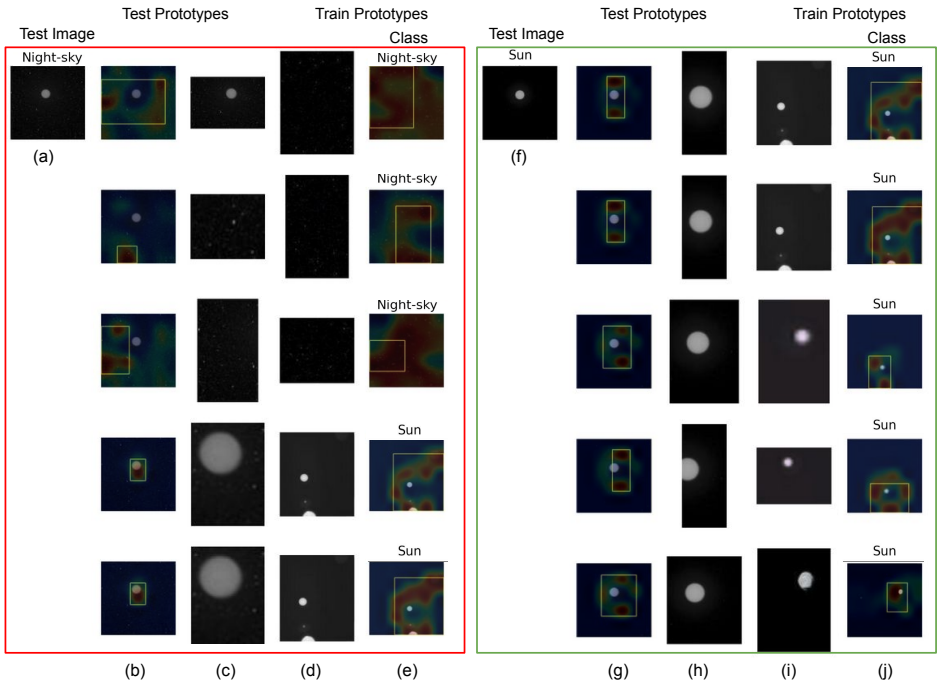

Figure 3: Explanation for two images from class *Sun* showing the difference between evidence when the image is misclassified as *Night Sky* (red, left) vs. when it is classified correctly as *Sun* (green, right) from a VGG19 backbone. The meaning of the columns is identical to Figure 1 where (a - e) represents output for the test image in (a) and (f - j) represents output for the test image in (f). Note the prototypes are ordered from most similar to least.

validation, and test set accuracy of all classifiers for only images that were classified with a confidence higher than 90% under column "Acc(0.9)" and the percentage of examples under the 90% confidence threshold (omitted from the Atlas display) listed as "Abst Rate". This is crucial as only classification results with a confidence level of 0.9 or greater are delivered to the Atlas. We also report the baseline accuracy when predicting the majority class under "Most Common", due to the imbalanced nature of our dataset. At an abstention rate less than 20%, the ResNet18 architecture fine-tuned on the MSL surface dataset yields the highest "Acc(0.9)" of 86.39% with an abstention rate of 15.5%. We observe a drop of $\sim 2.5\%$ in "Acc(0.9)" and a 9 percentage points rise (from 15.5% to 24.5%) in abstention rate compared to its uninterpretable counterpart. A slight drop in accuracy is expected as the addition of the prototypical layer constraints the information passed on to the final layers to be the prototypes. The work by Chen et al. (2019) suggests an ensemble of different backbone networks to further close the performance gap between a network with and without a prototypical layer. Furthermore, in future research endeavors, we intend to systematically examine the impact of employing a prototypical layer on the abstention rate, particularly following the application of model calibration techniques(Guo et al. 2017).

Qualitative Analysis

Figure 1 displays results for a sample image from the *Mastcam cal target* class. The calibration target, as seen in column (d), rows 1 and 3, emerges as the predominant prototype for classification. In contrast, the prototypes representing the background, located in column (d), rows 2 and 4, are secondary in influence. Notably, a significant portion of training images from the *Mastcam cal target* class, captured by the rover's integrated cameras, showcased a rocky backdrop surrounding the calibration target. This observation indicates that both the calibration target and its background contribute as evidence. Specifically, the background operates as negative evidence, evidenced by its negative weights of -0.45 and -0.44 in column (d), rows 2 and 4, respectively, guiding the accurate classification of the *Mastcam cal target*.

To demonstrate the agreement between the correctness of prototypes and classification results, we present prototypes from two different test images from class *Sun* in Figure 3. The evidence set marked in red(left) explains what led to misclassifying the image as class Night Sky while green(right) indicates a true positive. The top-3 prototypes in the incorrect classification focus on the emptiness of the night sky. This behavior might be a result of some training images from class *Night Sky* having similar features as class *Sun*, with both classes sharing visual attributes like the empty sky. On the other hand, when the image was classified correctly, the top evidence highly correlates to what a human might consider salient of the Sun, i.e., a bright, round shape. Therefore, using an interpretable content classification system can provide a deeper insight into the classifier's decision when compared to its un-interpretable counterpart.

	Train (n=5920)			Validation (n=300)			Test (n=600)		
	Acc	Acc (0.9)	Abst Rate	Acc	Acc (0.9)	Abst Rate	Acc	Acc (0.9)	Abst Rate
Most common (baseline)	26.3%	-	-	24.7%	-	-	31.2%	-	-
VGG19 (MSLNet)	99.4%	99.8%	1.5%	81.6%	85.6%	14.3%	81.3%	85.6%	12%
VGG19 + P-MSLNet	99.3%	99.8%	4.8%	77.3%	83.1%	21%	75.1%	82.5%	19.83%
VGG19 + P-MSLNet - TempCal	99.3%	99.7%	2.5%	77.3%	80.07%	14.6%	75.1%	94.5%	63.3%
VGG19 + P-MSLNet - VectorCal	99.3%	99.8%	2.4%	81.3%	94.8%	55%	79.6%	94.9%	53.8%
ResNet18 (MSLNet)	100%	100%	0%	83%	87.6%	21.6%	79.5%	86.3%	15.5%
ResNet18 + P-MSLNet	96.4%	98.6%	7.3%	76%	84.5%	24.3%	74.8%	83%	24.5%
ResNet18 + P-MSLNet - TempCal	96.4%	98.9%	10.6%	76%	94.9%	60.6%	74.8%	91.63%	66.1%
ResNet18 + P-MSLNet - VectorCal	96.5%	98.8%	10.4%	77.6%	95.2%	57.66%	76.8%	92.2%	61.1%

Table 1: Table showing train, validation and test accuracy along with threshold accuracy and abstention rate of different deep networks trained on the MSL surface dataset. MSLNet is a regular deep CNN, P-MSLNet is its Prototypical version. Note the performance metrics reported in this table do not incorporate the diversity loss; they solely aim to elucidate the influence of employing a prototypical layer.

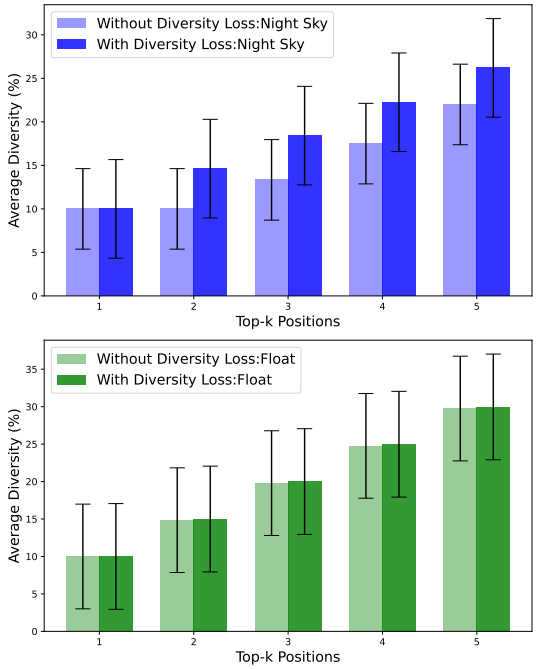

Figure 4: Comparison of the average prototype diversity over 100 prototypes for the most and least diverse classes, plotted against the position of the prototype based on the order of evidence (sorted based on importance), denoted as k, used for classifying MSL surface data. Class *Night Sky* sees significant improvement while class *Float Rock* has no improvement in diversity from the inclusion of the diversity loss term.

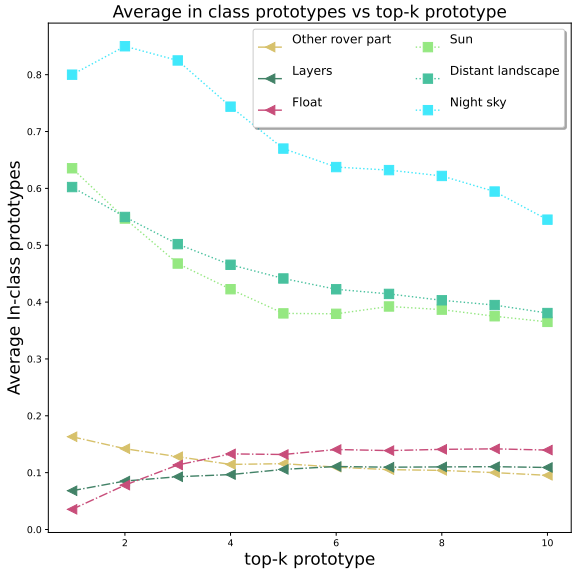

Figure 5: Average number of In-class prototypes for top-3 (square) and bottom-3 (triangle) most correct classes vs position of prototype in the order of evidence used k for classifying MSL surface test data by the VGG19 version of ProtoPNet.

Diversity and In-class Prototypes

To understand the nature of evidence used globally across classes, we present a quantitative evaluation of prototypes for their correctness and diversity. Evaluating the correctness of a prototype will help us to identify classes that use evidence from other classes, i.e., classes sharing visual attributes. In evaluating the diversity of prototypes learned during training, we aim to understand the generalizability of each class from the deep network's perspective. This is based on the assumption that a more diverse set of prototypes for a class enhances its ability to generalize. With our experimental setup, we noticed a 2 percentage point rise in test accuracy and no significant drop in train accuracy by introducing the diversity loss. Figure 4 shows a plot of the average number of unique training images from which the most similar prototypes for correctly classified test images come vs. the position of prototype $k \in [1, 5]$. Ideally, we expect a linear growth rate when every prototype comes from a different image. Figure 4 reports the results obtained using the VGG19 version of the ProtoPNet for the top-1 and bottom-1 most diverse classes. Class *Night Sky* has the least diverse set of prototypes learned during training, with at most $\sim$ 4 unique prototypes among ten prototypes. Class

Night Sky sees significant improvement in diversity from the inclusion of the diversity loss term proposed in this paper. Note there is no change in diversity for class *Float Rock* with the most diverse set of prototypes.

Finally, in Figure 5, we report the average number of in-class prototypes present in top-k most similar prototypes, i.e., the evidence used for classification that comes from an image that is the same class as the test image. Examining in-class prototypes provides insights into how much an image class depends on other classes. From Figure 5, it is evident that class *Night Sky* seems to have the most in-class prototypes on average across all classes, with almost 80% of the images having top-1 evidence coming from class *Night Sky*. Similarly, classes *Float Rock* and *Layered Rock* seems to have the least number of in-class prototypes with only 10% - 15% images containing in-class evidence indicating the lack of enough evidence during training between the two classes or the two classes have a lower inter-class variance.

Deployment Plan

The interpretable content based classification system studied in this paper will be evaluated for deployment in place of its un-interpretable counterpart currently in operation on the PDS Image Atlas[2]. Our deployment plan is as follows:

1. We plan to create a comprehensive user study with different levels of information being displayed to the user to assess their relative merits. Some questions we hope to answer are: How many prototypes should be displayed to the user? How much feedback is the user willing to provide?
2. Deploy the Proto-MSLNet classifier on MSL archives at the PDS Imaging Node after calibrating the posterior probabilities to inform decisions about which (high-confidence) images will be shown.
3. Driven by prior research in explanation visualization (Gunning et al. 2021; Vasu et al. 2021), we plan to modularize the cost of computing explanations to maintain user engagement while only providing explanations on-demand as explanations for simple decisions do not help the user. We plan to work with our software development and User Interface (UI)/User Experience (UX) design teams at the PDS Imaging Node to integrate the UX design shown in Figure 6 on the Atlas.
4. In addition to providing explanations, we would like to close the loop by allowing user to report misclassified images or erroneous evidence as shown in Figure 6. We would like to quantify and investigate methods to incorporate user feedback into model development. Our model refinement protocol incorporates a tripartite feedback integration strategy: (1) Database consolidation of misclassification feedback; (2) Analytical review to ascertain error causes; (3) Model enhancement based on prevalent error causes, including label rectification or training set expansion, prior to model retraining.

[2]https://pds-imaging.jpl.nasa.gov/search/

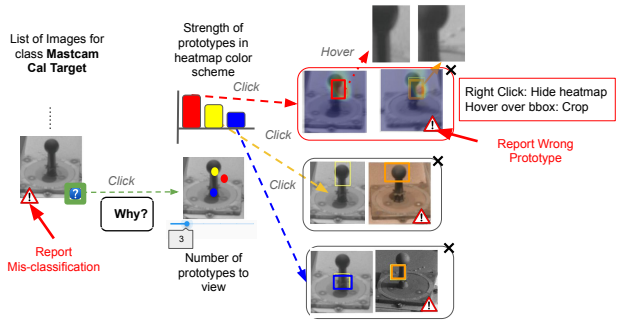

Figure 6: An illustration of user experience being considered for explanation visualization.

Conclusion

In this paper, we present our plans to deploy an interpretable content-based search on the Planetary Data System (PDS) Image Atlas. Building upon prior research on prototypical networks, we introduced a novel enhancement through the integration of a diversity cost. We report the results for classifiers trained on Mars images acquired from instruments on the Curiosity rover. Firstly, we demonstrated that an inherently interpretable network could be trained for imagery from Mars with a minimal performance drop of 2.5 percentage points with a 9 percentage points rise in abstention rate compared to its non-interpretable counterpart. In addition to highlighting the benefits of having an interpretable system through qualitative examples, we also report quantitative metrics that help us judge the quality of evidence learned for an image class. The ability to provide evidence used for content classification lets us debug spurious evidence used by the classifier and paves the way for a feedback mechanism from its users to improve model performance. In future work, we plan to investigate the effect of such reporting mechanisms on overall system improvement. We also plan to investigate different visualizations of prototypes to understand user preferences. While this paper focused on results from classifiers trained on the MSL surface dataset, we intend to broaden our research to include other classifiers such as the Mars Reconnaissance Orbiter and Mars Exploration Rover present on the Atlas. Overall, the work presented in this paper aims to render content classification across all missions on the NASA PDS system transparent and interpretable accelerating scientific discovery and aiding individual curiosity.

Acknowledgments

We thank the PDS Imaging Node for the continuing support of this work. We also thank the numerous volunteers involved in labeling Mars images enabling the work in this paper. The computing resources were provided by the Texas Advanced Computing Center (TACC). This research was partially funded through the NSF grant CNS-1941892, the Industry-University Cooperative Research Center on Pervasive Personalized Intelligence, and was carried out at the Jet

Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Aamodt, A.; and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1): 39–59.

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.

Bau, D.; Zhou, B.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6541–6549.

Bau, D.; Zhu, J.-Y.; Strobelt, H.; Zhou, B.; Tenenbaum, J. B.; Freeman, W. T.; and Torralba, A. 2019. GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chen, C.; Li, O.; Tao, D.; Barnett, A.; Rudin, C.; and Su, J. K. 2019. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32.

Gunning, D.; Vorm, E.; Wang, J. Y.; and Turek, M. 2021. DARPA's explainable AI (XAI) program: A retrospective. *Applied AI Letters*, 2(4): e61.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330. PMLR.

Khorram, S.; Lawson, T.; and Fuxin, L. 2021. iGOS++ integrated gradient optimized saliency by bilateral perturbations. In *Proceedings of the Conference on Health, Inference, and Learning*, 174–182.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kolodner, J. L. 1992. An introduction to case-based reasoning. *Artificial intelligence review*, 6(1): 3–34.

Lakkaraju, H.; and Bastani, O. 2020. "How do I fool you?" Manipulating User Trust via Misleading Black Box Explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 79–85.

Lucas, M.; Lerma, M.; Furst, J.; and Raicu, D. 2022. RSI-Grad-CAM: Visual explanations from deep networks via Riemann-Stieltjes integrated gradient-based localization. In *International Symposium on Visual Computing*, 262–274. Springer.

Nauck, D.; and Kruse, R. 1999. Obtaining interpretable fuzzy classification rules from medical data. *Artificial intelligence in medicine*, 16(2): 149–169.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215.

Rui, Y.; Huang, T.; Ortega, M.; and Mehrotra, S. 1998. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5): 644–655.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on computer vision*, 618–626.

Vasu, B.; Hu, B.; Dong, B.; Collins, R.; and Hoogs, A. 2021. Explainable, interactive content-based image retrieval. *Applied AI Letters*, 2(4): e41.

Wagstaff, K.; Lu, S.; Dunkel, E.; Grimes, K.; Zhao, B.; Cai, J.; Cole, S. B.; Doran, G.; Francis, R.; Lee, J.; et al. 2021. Mars image content classification: Three years of NASA deployment and recent advances. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15204–15213.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, N.; Donahue, J.; Girshick, R.; and Darrell, T. 2014. Part-based R-CNNs for fine-grained category detection. In *European conference on computer vision*, 834–849. Springer.

Zheng, H.; Fu, J.; Mei, T.; and Luo, J. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, 5209–5217.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.