A Supplementary Material

A.1 Resource Access

Code Repository: Code for the MISO generator and downstream analyses (*e.g.*, code to generate Interpretability Report Cards) can be found at https://github.com/gmachiraju/codex-analysis

Benchmarking Datasets: The described datasets (MISO-1 and MISO-2) are available at the following link: https://drive.google.com/drive/folders/1fOtOn6M41y0VXHtNyxO513Hf5hOphRNa?usp=sharing

A.2 Acknowledgements

Funding: The authors are grateful for institutional support from Stanford Data Science, as well as Biomedical Informatics Training Program at Stanford 2T15LM007033. The authors are also grateful for the support of this research through DARPA Deep Purple Program through DOI program award D17AC00006, DARPA SIMPLEX program award W911NF-15-1-0555, DARPA PAI program award HR00111890036, and NIH awards 1R01GM117097 and 5R01CA249899.

Software Licenses: Figure 1, Figure 2, and Figure 6 were created with BioRender.com.

A.3 MISO-1 Data Generation & Dataset Specifications

To generate MISO-1, we collected 24 source images from Google Images as input bitmaps (*e.g.*, RGB color images). The input bitmaps were chosen for their objects' varied large-scale morphologies and textures, ultimately helping us define borders and foreground regions for downstream image processing. Please refer to Figure 5, associated hyperlinks, copyright, and terms of service in the following subsections.

MISO, written in Python 3.6, takes 2D bitmaps as inputs, rescales them to megapixel size, binarizes them into binary masks (either through manual or automatic thresholds) with 1-valued foregrounds and 0-valued backgrounds, and then generates training and testing images for its six separate scenarios by manipulating the masks' pixel values (*e.g.*, flipping binary values, performing Hadamard products with other masks, applying distortions) and selecting class members. To aid in dataset variation and difficulty, all images have corresponding versions generated with "salt or pepper" fuzziness (*i.e.*, a pixel can take on the opposite extreme value of the large scale morphology with 10% probability). Finally, all images are given uniform random-noise backgrounds between [0,1] that can be filtered or unfiltered before model training and testing to modulate the degree of supervision. A graphical depiction of the pipeline can be found in

Figure 6. MISO-1 images are all at least 2000 × 2000 pixels large and can reach approximately 7000 pixels in either dimension. After performing patching and patch filtering, roughly 80,000 to 100,000 patches are available for training per scenario. MISO-2 is comprised of 3072 × 3072 images, yielding approximately 40,000 training patches for our selected patch size.

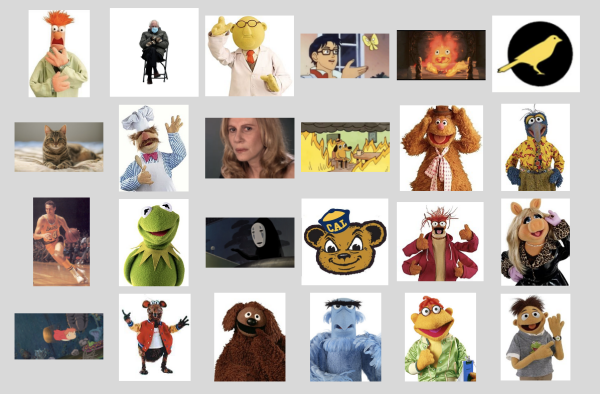


Fig. 5: MISO-1's input 2D bitmaps.

Existing assets — source images for MISO-1: We accessed all 24 source images used in this study (as 2D input bitmaps for MISO) via Google Images. We claim Fair Usage of the source images (and any of its protected works) for the following reasons: (1) solely reading the source images into a computer program (*i.e.*, MISO) and not distributing the images themselves, copies, or adaptations of them; (2) generating synthetic images that use source images loosely but incorporate multiple substantial aesthetic modifications made through MISO (*e.g.*, binarization, noisy background creation, addition of fuzziness, etc. as discussed in the following section); and (3) only distributing the final synthetic image dataset for educational and research purposes. Finally, we underscore the nonprofit, educational, and research-driven nature of MISO's synthetic dataset creation of MISO-1. Links to either the image or link address are below (with the shorter one listed):

1. https://vignette.wikia.nocookie.net/79e4f646-c685-4c69-8ace-d45f7805253a/scale-to-width-down/1200
2. https://www.logodesignlove.com/wp-content/uploads/2011/04/jerry-west-nba-logo.jpg
3. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.meme-arsenal.com%2Fen%2Fcreate%2Ftemplate%2F1519250&psig=AOvVaw3Vyu0Qkdywn3rHVYueQbYo&ust=1633632975111000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCLCHo5y7tvMCFQAAAAAdAAAAABAI
4. https://i.insider.com/5af0a543bd96711f008b4623?width=1136&format=jpeg

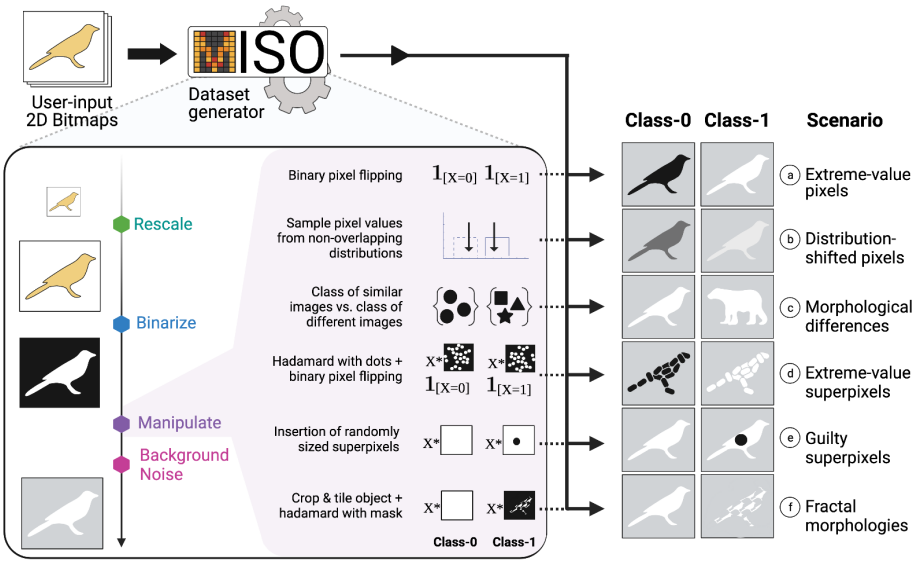


Fig. 6: A schematic of the pipeline used by the MISO generator pipeline to construct MISO-1.

5. https://c.files.bbci.co.uk/151AB/production/_111434468_gettyimages-1143489763.jpg
6. https://www.google.com/url?sa=i&url=https%3A%2F%2Fknowyourmeme.com%2Fmemes%2Fthis-is-fine&psig=AOvVaw1TWJ55w3hpwXFp5qACijJT&ust=1633633735757000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCNCAnou-tvMCFQAAAAAdAAAAABAD
7. https://cdn.vox-cdn.com/thumbor/S9Laj6tGT1pkHegZB-8GDVqZSl4=/0x29:1869x1008/fit-in/1200x630/cdn.vox-cdn.com/uploads/chorus_asset/file/20010436/ponyo.jpg
8. https://www.syfy.com/sites/syfy/files/styles/1200x680_hero/public/calcifer-howls-moving-castle.jpg
9. https://canarycenter.stanford.edu/core-facilities/preclinical-imaging/training-request1/_jcr_content/main/panel_builder/panel_1/image.img.476.high.png/canary-center-logo.png
10. https://4.bp.blogspot.com/-nkz9NPgC5wI/Utbtdib4v_I/AAAAAAAAVqY/N9HVQncqZ7w/s1600/Rizzo.png
11. https://upload.wikimedia.org/wikipedia/en/2/22/MissPiggy.jpg
12. https://upload.wikimedia.org/wikipedia/en/thumb/6/6d/Walter_%28Muppet%29.jpg/220px-Walter_%28Muppet%29.jpg
13. https://upload.wikimedia.org/wikipedia/en/d/df/ScooterMuppet.jpg
14. https://upload.wikimedia.org/wikipedia/en/b/b4/SamTheEagle.jpg
15. https://upload.wikimedia.org/wikipedia/en/b/b5/Rowlf_the_Dog.jpg

16. https://upload.wikimedia.org/wikipedia/en/0/0a/Gonzo_the_Great.jpg
17. https://upload.wikimedia.org/wikipedia/en/6/62/Kermit_the_Frog.jpg
18. https://upload.wikimedia.org/wikipedia/en/d/dd/Dr._Bunsen_Honeydew.jpg
19. https://i.etsystatic.com/16306234/r/il/94533b/2833342520/il_1140xN.2833342520_i7j8.jpg
20. https://upload.wikimedia.org/wikipedia/en/5/51/Fozzie_Bear.jpg
21. https://upload.wikimedia.org/wikipedia/en/3/32/Pepe_the_King_Prawn_%28Muppet%29.jpg
22. https://cdn2.apstatic.com/photos/climb/113334801_medium_1500210180.jpg
23. https://upload.wikimedia.org/wikipedia/en/e/e7/The_Swedish_Chef.jpg
24. https://upload.wikimedia.org/wikipedia/en/5/59/Beaker_%28Muppet%29.jpg

A.4 Design Configurations for Patch-based Convolutional Neural Networks

This section outlines and summarizes major design configurations observed in the literature. These choices conceivably can each impact prediction and explanation correctness. Thus, future studies should explicitly state configuration choices to facilitate the benchmarking process.

Preprocessing Configurations

Patch size: This is a major hyperparameter for PatchCNNs. This must be chosen based on application domain (*i.e.*, image resolution, pixel count, expected object sizes, etc.) to balance the trade-off between a sufficiently large training set size and sufficient patch heterogeneity. Namely, the larger the patch size, the dataset gains patch heterogeneity due to containing more objects or a larger fraction of objects. However, this leads to smaller training and testing set sizes. The smaller the patch size, a dataset can lose patch heterogeneity, but can enjoy larger training and testing set sizes. In histopathology for example, a patch size is typically selected to capture a few dozen cells per patch. For MISO-1, we arbitrarily chose a patch size of 96×96 pixels. For MISO-2, we chose a patch size of 224×224 pixels, which is somewhat common for $10\times$ microscopy imagery.

Patch sampling strategy: This design choice surrounds the sampling of constituent patches from the originating megapixel images. A popular choice to maximize dataset size is to enforce a Cartesian grid and patch the image via a sliding window [17, 49]. Additionally, a 50% x- and y- shift can also be applied to the starting position to capture objects on the patch borders of the original grid. This study followed this two-part Cartesian grid sampling approach. Random sampling is a popular option when many source images are available.

Patch normalization: While channel-wise mean or median normalization has been practiced for some time, new methods for domain-specific applications (*e.g.*, multiplexed histopathology [41]) have helped to offer solutions that remove technical error in imaging. For the MISO-1 and MISO-2 datasets, we skip normalization since all pixel values are defined between [0,1].

Patch filtering functions & rules: Depending on the prediction task and the contents of the images where patches originate, a distinction can potentially be made between the background and foreground of the image. This step is typically employed in histopathology settings since specimens do not take up the whole slide on which they are imaged. In this setting, background patches are typically filtered out by some function or rule (*e.g.*, intensity threshold [17]). If a background-foreground distinction cannot be made, the whole image can simply be used for patching. For the experiments conducted in this study, we performed background-filtration (*i.e.*, a filtering rule centered around patch pixel mean of 0.5 with tolerance of ± 0.05) to maximize the level of supervision for our models. This resulted in approximately 100K patches per scenario.

Patch dataset augmentation: Data augmentation via rotations and reflections has become a popular strategy for increasing sample sizes in image domains where rotational equivariance is assumed (*e.g.*, histopathology, remote sensing, cosmology). For simplicity and due to sufficient dataset size, we did not use augmentation techniques.

Patch labeling functions & rules: Many of the current strategies are fuzzy and can be conceptualized as the weak supervision attributed with fuzzy labeling or data programming [83]. At the image-level, labels are often described as *coarse* [85] — *i.e.*, labels actually refer to specific aspects of images. Studies today often take an ILI approach (see paper body) to patch labeling [17]. Pseudo-labeling is also an active area of research [7]. *Proxy-based labeling*, as we call it, is another approach that uses an informative channel and has garnered recent attention [49]. We performed ILI patch labeling in this study due to its simplicity and popularity.

Stage-1 Model Configurations

Patch data loading for mini-batch creation: Standard practice is to perform randomized data loading into PatchCNN architectures, furthering the IID assumption made by current popular models. We perform randomized data loading with a mini-batch sizes of 36 patches for MISO-1 and 7 patches for MISO-2.

Patch-level classifier architecture & loss function: While there are many CNN architectures that could operate as PatchCNNs, we chose two representative architectures for experimentation: *VGG-19* [96] and *VGG with Attention (VGG-Att)* with pre-pooling Attention modules [51]. In tandem with ILI patch labeling and randomized data loading, a popular patch-level loss function is the standard binary cross-entropy loss [17]. We follow suit for our PatchCNNs.

Training time: Number of epochs for training a model. We trained models for 10 epochs on MISO-1 and 20 epochs for MISO-2.

Stage-2 Model Configurations

Patch aggregation functions & rules for image-level classification: There exist many possible approaches to perform image-level classification, ranging from simple decision rules to trainable functions (with inputs as individual patch predictions, prediction probabilities, hidden vectors, etc.) [17]. One popular example of a decision rule is the *Multiple Instance Learning* (*MIL*) rule (*i.e.*, max pooling) operating on independent patch votes. The MIL rule has seen widespread use in megapixel image classification [17, 108], often where the salient objects indicative of class label are relatively small compared to the image size and thus necessitate a highly sensitive classifier. For their simplicity, we also construct six decision rules based on patch votes and probabilities: (1) *top-20 majority voting*, or taking patches with the top-20 prediction probabilities and performing a majority vote with a 0.5 threshold; (2) the MIL rule, or max-pool of the patch predictions; (3) *majority voting*, or mean-pool with a 0.5 threshold; (4) *weighted majority voting*, or the average expectation of patch votes and patch probabilities with a 0.5 threshold; (5) *caucus max-pooling*, or iterative max-pooling using 10-by-10 blocks of patch predictions; and (6) *caucus majority voting*, or iterative mean-pooling using 10-by-10 blocks of patch predictions with a 0.5 threshold.

Explanation Configurations

Explanation Mapping technique: Since patches are the unit of analysis for a PatchCNN, we compute Explanation Maps at the patch-level. For their simplicity, we compute Saliency Maps (for VGG-19 and VGG-Att) and Attention Maps (for VGG-Att) per input patch at test-time.

Explanation Map post-processing: We take the average of the absolute saliency and attention scores [3] for each patch in order to both speed up runtime and eventually smooth out ROIs constructed from salient objects. Our approach concatenates these scores to construct an array of scores per image that we refer to as *Stitched Saliency Maps* (*SSMs*) and *Stitched Attention Maps* (*SAMs*). Binarization of the SSMs or SAMs was also performed with the popular adaptive threshold of double the mean saliency or attention score [13]. Finally, to deal with spurious ROIs, studies have shown the utility of taking the Hadamard product between predictions (*i.e.*, PPMs) and binarized explanation maps (*i.e.*, SSMs or SAMs) as a post-processing step [3]. We skipped this step to observe unaltered explanations for MISO-1.

A.5 Interpretability Report Cards: Additional Tables

We present additional statistics for our analyses in Table 3, Table 4, and Table 5.

Table 3: Patch- and image-level classification statistics generated on held-out test sets per scenario. Patch-level statistics assume ILI labeling data as ground truth. Image-level statistics reflect the maximum value over all six patch aggregation functions or decision rules used. Only patches kept post-filtration were used to generate these statistics. A **boldface** result indicates a superior score between architectures for a given scenario (directionality denoted by ↑, ↓).

	Patch-level (ILI labeling)			Image-level		
Scenario	AUROC$^{\uparrow}$	AUPRC$^{\uparrow}$	AP$^{\uparrow}$	AUROC$^{\uparrow}$	AUPRC$^{\uparrow}$	AP$^{\uparrow}$
VGG-19						
EVP	1.000	1.000	1.000	1.000	1.000	1.000
DSP	0.500	0.750	0.500	0.500	0.750	0.500
MD	0.500	0.796	0.592	0.596	0.750	0.543
EVSP	0.500	0.750	0.500	0.501	0.750	0.501
GSP	0.500	**0.750**	0.500	0.500	0.750	0.500
FM	0.500	0.711	0.423	0.501	0.750	0.503
MISO-2	0.500	**0.752**	0.504	0.515	0.750	0.506
VGG-Att						
EVP	1.000	1.000	1.000	1.000	1.000	1.000
DSP	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
MD	**0.905**	**0.933**	**0.926**	**1.000**	**1.000**	**1.000**
EVSP	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
GSP	**0.520**	0.539	**0.538**	**1.000**	**1.000**	**1.000**
FM	**0.996**	**0.997**	**0.997**	**1.000**	**1.000**	**1.000**
MISO-2	**0.556**	0.593	**0.593**	**0.790**	**0.772**	**0.781**

Table 4: Additional PPM statistics and PCM statistics on held-out test sets per scenario. Notably, the GSP scenario is omitted as was the case in Table 1 due to the difficulty in assessing correctness given the ILI patch labeling regime. A **boldface** result indicates a superior score between architectures for a given scenario (directionality denoted by $\uparrow, \downarrow$). †Generated mean value required removal of a handful of test images to avoid runtime or memory issues. $^{\clubsuit}$Denotes a binary map evaluation. $^{\spadesuit}$Denotes a non-binary map evaluation. $^{\diamondsuit}$Denotes a structural evaluation. $^{\heartsuit}$Denotes an IID patch-level evaluation.

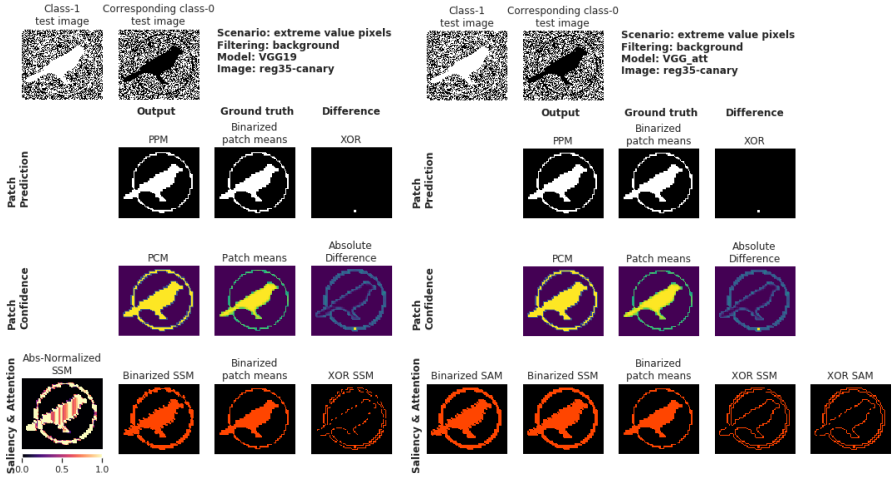
	PPMs			PCMs	
Scenario	Dice (F_1)$^{\clubsuit\heartsuit\uparrow}$	Jaccard$^{\clubsuit\heartsuit\uparrow}$	Overlap$^{\clubsuit\heartsuit\uparrow}$	MAE$^{\spadesuit\heartsuit\downarrow}$	SSIM$^{\spadesuit\diamondsuit\uparrow}$
VGG-19					
EVP	0.952 ± 0.012	0.914 ± 0.020	0.999 ± 0.000	0.235 ± 0.042	0.593 ± 0.062
DSP	0.499 ± 0.100	0.499 ± 0.100	0.500 ± 0.100	0.108 ± 0.010	0.585 ± 0.066
MD	0.499 ± 0.092	0.499 ± 0.092	0.500 ± 0.093	0.136 ± 0.010	0.625 ± 0.040
EVSP	0.363 ± 0.075	0.293 ± 0.063	0.498 ± 0.100	**0.081** ± 0.016	0.639 ± 0.059
FM	0.451 ± 0.090	0.413 ± 0.083	0.500 ± 0.100	0.198 ± 0.027	0.572 ± 0.063
MISO-2	**0.567**† ± 0.170	**0.554**† ± 0.166	0.577† ± 0.173	**0.223** ± 0.036	**0.227** ± 0.074
VGG-Att					
EVP	0.952 ± 0.012	0.914 ± 0.020	0.999 ± 0.000	0.235 ± 0.042	0.593 ± 0.062
DSP	**0.949** ± 0.013	**0.909** ± 0.021	**0.999** ± 0.000	**0.227** ± 0.029	**0.677** ± 0.042
MD	**0.719** ± 0.050	**0.633** ± 0.063	**0.732** ± 0.050	**0.115** ± 0.006	**0.674** ± 0.014
EVSP	**0.861** ± 0.033	**0.790** ± 0.047	**0.998** ± 0.001	0.132 ± 0.017	**0.676** ± 0.027
FM	**0.942** ± 0.012	**0.896** ± 0.020	**0.998** ± 0.001	**0.089** ± 0.010	**0.811** ± 0.020
MISO-2	0.505† ± 0.093	0.387† ± 0.088	**0.748**† ± 0.088	0.248 ± 0.024	0.133 ± 0.045

Table 5: Additional SSM and SAM statistics on held-out test sets per scenario. A **boldface** result indicates a superior score between architectures for a given scenario (directionality denoted by $\uparrow, \downarrow$). *Indicates that SAMs yielded a top score for VGG-Att. $^{\clubsuit}$Denotes a binary map evaluation. $^{\heartsuit}$Denotes an IID patch-level evaluation.

Scenario	Dice (F_1)$^{\clubsuit\heartsuit\uparrow}$	Jaccard$^{\clubsuit\heartsuit\uparrow}$	Overlap$^{\clubsuit\heartsuit\uparrow}$	MAE (binary)$^{\clubsuit\heartsuit\downarrow}$	Sensitivity$^{\clubsuit\heartsuit\uparrow}$	Specificity$^{\clubsuit\heartsuit\uparrow}$
VGG-19						
EVP	0.406 ± 0.098	0.327 ± 0.094	0.961 ± 0.012	0.348 ± 0.064	0.385 ± 0.117	0.981 ± 0.006
DSP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.413 ± 0.051	0.000 ± 0.000	**1.000** ± 0.000
MD	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.394 ± 0.049	0.000 ± 0.000	**1.000** ± 0.000
EVSP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.206 ± 0.043	0.000 ± 0.000	**1.000** ± 0.000
GSP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	**0.020** ± 0.004	0.000 ± 0.000	**1.000** ± 0.000
FM	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.331 ± 0.058	0.000 ± 0.000	**1.000** ± 0.000
MISO-2	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.660 ± 0.023	0.000 ± 0.000	**1.000** ± 0.000
VGG-Att						
EVP	**0.525** ± 0.093	**0.428** ± 0.092	**0.974** ± 0.040	**0.300** ± 0.063	**0.471** ± 0.107	**0.990** ± 0.006
DSP	**0.528*** ± 0.105	**0.446*** ± 0.099	**0.930*** ± 0.068	**0.285*** ± 0.066	**0.493*** ± 0.114	0.990* ± 0.005
MD	**0.567*** ± 0.092	**0.476*** ± 0.086	**0.956*** ± 0.015	**0.259*** ± 0.060	**0.519*** ± 0.096	0.988* ± 0.004
EVSP	**0.710*** ± 0.040	**0.573** ± 0.071	**0.999** ± 0.001	**0.111*** ± 0.025	**0.877** ± 0.076	0.945* ± 0.013
GSP	**0.357** ± 0.069	**0.247** ± 0.056	**0.694** ± 0.055	0.064 ± 0.018	**0.620** ± 0.071	0.945 ± 0.020
FM	**0.601*** ± 0.051	**0.453*** ± 0.051	**0.982*** ± 0.012	**0.215*** ± 0.047	**0.621*** ± 0.089	0.971* ± 0.010
MISO-2	**0.243*** ± 0.056	**0.144*** ± 0.037	**0.997** ± 0.004	**0.569*** ± 0.041	**0.144*** ± 0.037	0.999 ± 0.001

A.6 Interpretability Report Cards: Illustrative Inputs

We also provide illustrative examples (*i.e.*, local explanations) that corroborate average test set performance results found in the presented tables. These examples can be found below.



(a) VGG-19 **(b)** VGG-Att

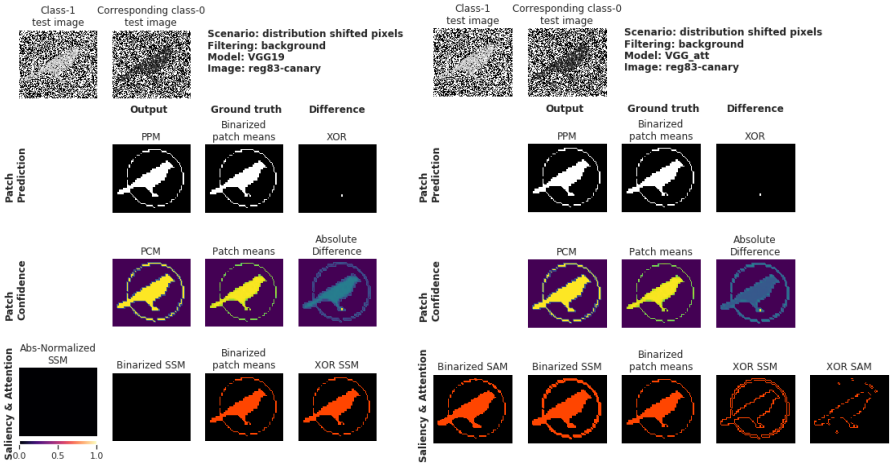
Fig. 7: Example Report Cards for extreme-value pixels (EVP).

A.7 Limitations, Cautions, & Opportunity Areas

Limitations of this study pertain mostly to the simplicity in MISO-1's dataset properties, experimental protocol, and evaluation statistics. One such dataset limitation is MISO-1's and MISO-2's usage of 1-channel images. These provided datasets do not focus on testing differentially expressed textures (at either local or global scales) given its greater difficulty of generation and domain-specific nature. Future work will explore this line of experimentation.

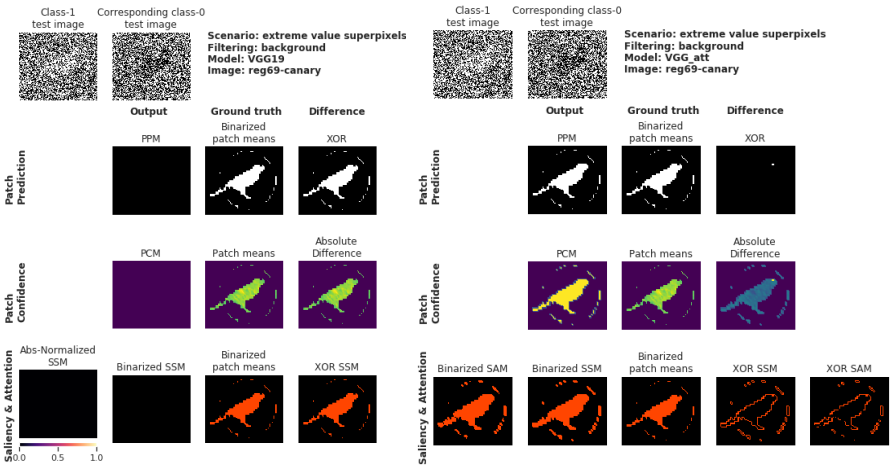
Regarding experimentation, all models trained on MISO-1 were trained for 10 epochs and all models trained on MISO-2 were trained for 20 epochs for uniformity. Future work should implement early stopping and other practices to ensure the most appropriate architecture selection. Future work should also toggle various design configurations (Appendix A.4) to understand their effects on predictions and explanations — including the patch size hyperparameter, patch filtering and labeling functions, and more sophisticated Explanation Mapping techniques [84, 90]. To quantify spurious correlations, patch filtering can be omitted for training and testing, and SSM and SAM Sensitivity (among other statistics) can be computed.

Regarding Stage-1 architectures, simple baselines were chosen. Both equivariance [25, 48, 73, 104] and more sophisticated Attention-based architectures (*e.g.*,



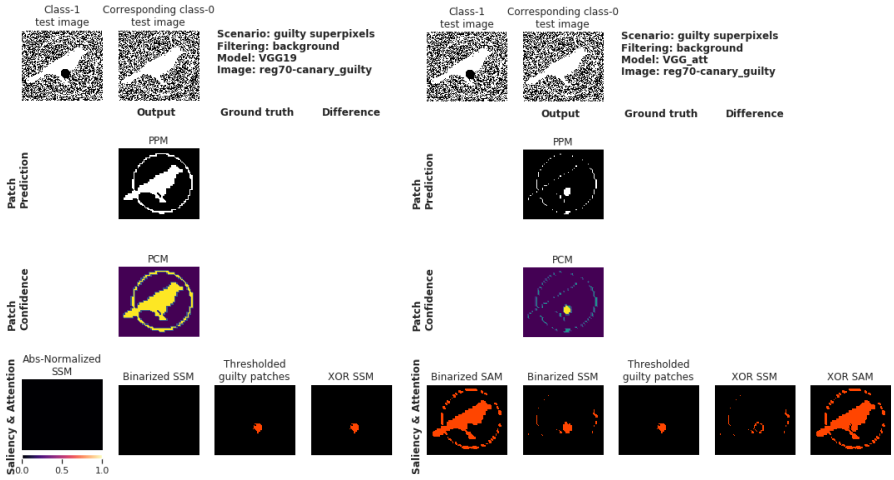
(a) VGG-19 **(b)** VGG-Att

Fig. 8: Example Report Cards for distribution-shifted pixels (DSP).



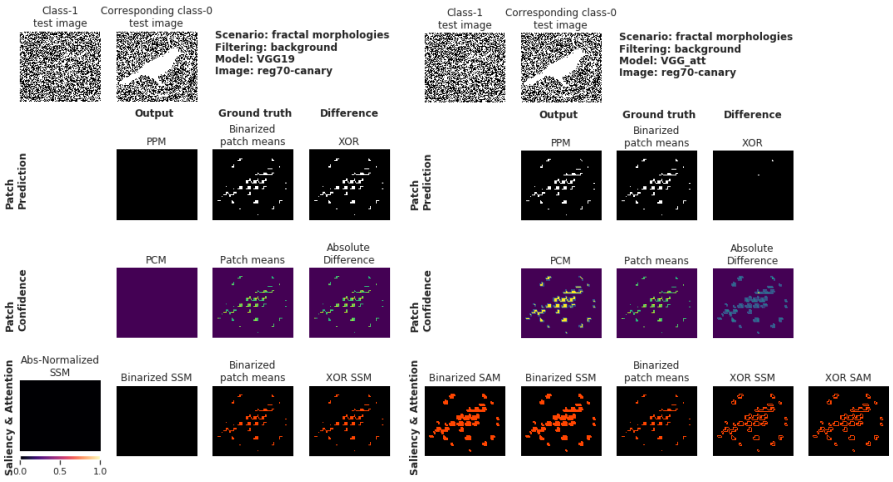
(a) VGG-19 **(b)** VGG-Att

Fig. 9: Example Report Cards for extreme-value superpixels (EVSP).



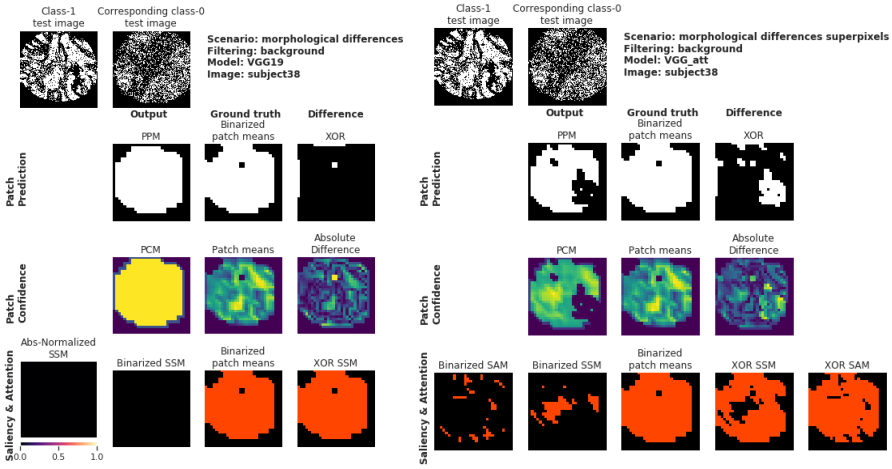
(a) VGG-19 **(b)** VGG-Att

Fig. 10: Example Report Cards for guilty superpixels (GSP).



(a) VGG-19 **(b)** VGG-Att

Fig. 11: Example Report Cards for fractal morphologies (FM).



(a) VGG-19 **(b)** VGG-Att

Fig. 12: Example Report Cards for MISO-2.

ViTs, Hierarchical ViTs [71]) should be pursued. Furthermore, the community should explore architectures and training regimes that soften the assumptions made by two-stage modeling approaches. Recent works in Multi-task Auxiliary Learning frameworks [66, 70] combine Stages 1 and 2 of the standard PatchCNN pipeline and have been proposed for segmentation [35] and classification [102]. Additionally, recent works to encode patch locations contextualized embeddings during supervision [6, 92, 114] show promise. Finally, Contrastive Learning approaches should be used to mitigate spurious correlations and increase specificity of salient ROIs [120]. While all these approaches show promise, their explanations have not been quantitatively evaluated via wsSOD and should be explored using MISO-1 and MISO-2.

Regarding explanations, this study made large use of Saliency Mapping, a standard gradient-based explanation method. State-of-the-art Explanation Mapping methods [84, 91] should be explored as well to find optimal explanation configurations. In particular, Class Activation Maps [91] specialize in identifying differentially expressed salient objects and should be explored further.

Regarding evaluation, despite having patch prediction probabilities structured as PCMs, we were unable to generate average per-image classification statistics (*e.g.*, AUROC, AUPRC, and AP) due to ILI labeling and background filtration resulting in patches that belong to a single class per image in most scenarios (other than GSP). This study also highlights the need for new evaluation statistics tailored to the modeling paradigms used for megapixel images. Due to the common practice of background filtering via patch filtering functions (*e.g.*, in histopathology), standard statistics result in inflated scores across PPMs, PCMs, SSMs, and SAMs. Specifically, statistics that score non-binary patches independently (*e.g.*, MAE) are not very discriminative in settings where majority of the image is comprised of background pixels. Because of this, an over-reliance on

these statistics need to be avoided and should instead drive the creation of new statistics.

We also note the absence of some recently proposed evaluation methods used in the field. Firstly, we plan to expand PCM evaluation by computing the S-measure and continuous Dice coefficients (cDCs) [93] against PPM ground truths. Future work will also expand PPM, SSM, and SAM evaluation through the use other structural similarity approaches used for masks [101, 119] and in more general image processing settings [27, 45].