DiG-IN: Diffusion Guidance for Investigating Networks - Uncovering Classifier Differences, Neuron Visualisations, and Visual Counterfactual Explanations

Maximilian Augustin, Yannic Neuhaus, Matthias Hein

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Overview

- **Goal:** Investigate classifiers by using optimization-based image generation on the natural image manifold
- **Finding "Unknown Unknowns"**: Identify failure cases in subpopulations of natural images that are not easily found in existing datasets or described textually
- **Challenges:** Difficult to visualize semantic meanings of concepts (e.g. single neurons) or create counterfactual explanations, as optimizing in pixel space can lead to adversarial samples
- **Approach:** we use images generated by latent diffusion models (Stable Diffusion) as approximation of the natural image manifold and optimize over the generation process
- **DiG-IN**: Our framework for investigating neural networks using diffusion guidance
- **Tasks:** In our paper, we discuss:
  - Generation of maximally disagreeing images
  - Visual counterfactual explanations of classifier decisions
  - Visualization of concepts that maximize/minimize neurons and detection of spurious features

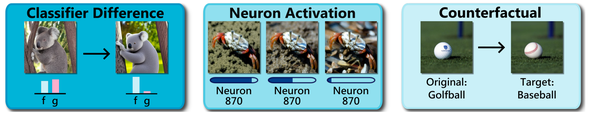
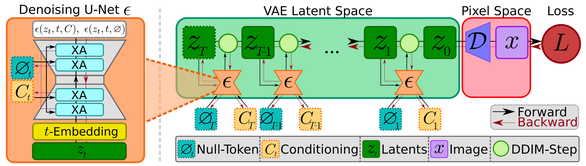


Image Generation Framework

- **Optimization Framework for Image Generation**
- **Guidance:** Directed by one or multiple classifiers (e.g. classifier disagreement and VCEs) or their properties (e.g. neuron activations)
- **Optimization Problem:** Formulate tasks using a loss function L on the generated image
- **Objective:** Search for a starting latent and conditioning to generate an image optimizing loss L without manual prompt tuning or supervision. Optimize latent z_T, conditioning C and null-text $\varnothing$ via standard first-order optimization:

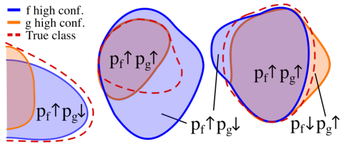
$$\max_{z_T,(C_t)_{t=1}^T,(\varnothing_t)_{t=1}^T} -L\Big(\mathcal{D}\big(\mathbf{z}_0\left(z_T,(C_t)_{t=1}^T,(\varnothing_t)_{t=1}^T\right)\big)\Big)$$

- **Plug-and-Play:** Usable with any vision model without requiring finetuning of the generative model
- **Backpropagation:** through the diffusion graph (20 steps)



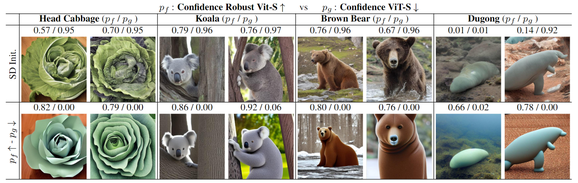
Classifier Disagreement

- Generate images that maximize one classifier's confidence for a class while minimizing it for another classifier
- Useful for showcasing differences due to varying training methods, datasets, or after updating an existing model



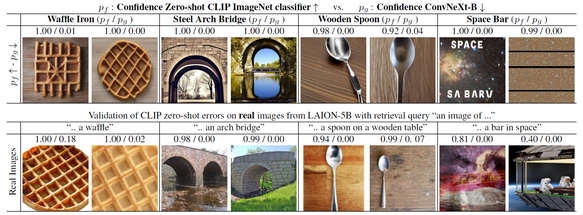
Shape Bias of Adversarial Robust Models

- **Adversarial Robust Models:** Known for having a shape bias
- **Generation:** Maximize conf. of robust ViT and minimize for standard ViT
- **Resulting Images:** Retain shape but have smooth surfaces and little texture, verifying shape bias



CLIP zero-shot Errors

- **Systematic Misclassification:** CLIP model misclassifies certain subpopulations:
  - Waffles as "waffle iron"
  - Stone bridges as "steel arch bridges"
  - Spoons on a wooden table as "wooden spoon"
  - Images with space and bar as "space bar"
- **Validation:** Find similar real images in LAION-5B (second row)
- **Error Source:** Likely due to text embedding composition of class names

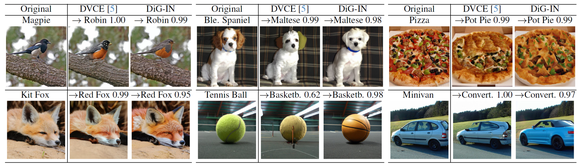


Visual Counterfactual Explanations

- **Counterfactual Reasoning** for Model Understanding
- **Visual Counterfactual Explanation (VCE):**
  - New image for input $\hat{x}$, target class y, and classifier f
  - Classified as y by f
  - Looks realistic (on the natural image manifold)
  - Minimal changes to the input
- **Loss Function:** Includes *background* similarity measure to the original image and predicted probability of f in class y
- **Initialization:** Null-text diffusion inversion
- **Prior Methods:** Require dataset-specific adv. robust and diffusion model (e.g. DVCEs)
- **DiG-IN:** Training-free, produces VCEs for any classifier on any natural image dataset (Universal VCE - UVCE)

ImageNet VCEs

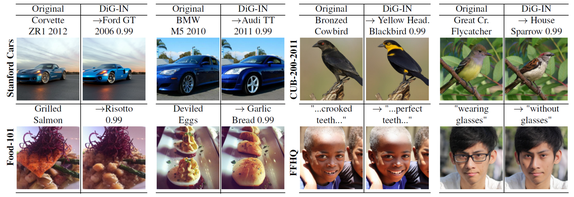
- **DVCEs vs. DiG-IN UVCEs:**
  - **Realism:** UVCEs are more realistic looking
  - **Texture Changes:** UVCEs produce fine-grained texture changes (e.g. "Red Fox", "Basketball")
  - **Geometric Transformations:** UVCEs handle complex geometric changes (e.g. "Pot Pie", "Convertible") better than DVCEs
  - **Coherence:** DVCEs can fail to create coherent objects



UVCEs on other Datasets

- **DiG-IN UVCEs:** First method that can create realistic VCEs for *any* classifier trained *on any* natural image dataset.

For Example:
- Stanford Cars (ResNet101)
- CUB-200 (ViT-B)
- Food-101 (ViT-B)
- FFHQ (CLIP zero-shot attribute classifier)

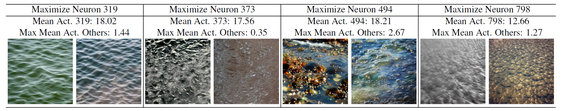


Neuron Visualisations

- Visualize semantic meaning of neurons
- Focus neurons in the last layer of a classification model that capture more semantically meaningful concepts

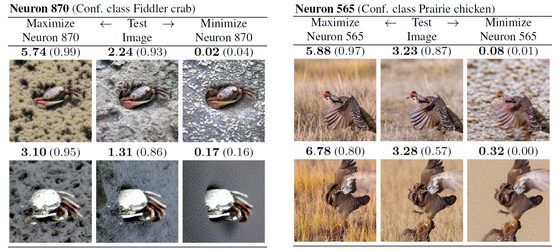
Neuron Visualizations:

- Generate Prototypical Examples for Target Neuron n
- **Traditional Method:** Inspect highly active training images, can be ambiguous due to varied subpopulations
- **DiG-IN Neuron Visualisations:** Fully-automated workflow that allows to identify semantic concepts captured by a neuron
- **Distinguish** subtle differences between similar neurons. Different water neurons capture different kinds of water



Neuron Counterfactuals:

- **Initialization:** Start from inverted test image
- **Goal:** Max- and minimze neuron activation in the background
- **Loss Function:** Includes *foreground* similarity measure to the original image and target neuron activation
- **Maximizing** the neuron enhances feature, **minimizing** removes it:
  - Sand for "Fiddler crab", Dry grass for "Prairie chicken"
- **Implication:** Neurons correspond to spurious features – the class object is always fully visible, but confidence varies highly depending on the activation of the spurious neuron



Links

Code

CVPR

arXiv