Model Explanation for Masked Autoencoders with Finetuning

Zhiyu Xue, Zhiying Li

https://colab.research.google.com/drive/1BZusNV-90M-9utI7WVW3tjsoRDkI_2Vq?usp=sharing

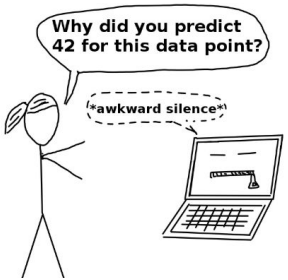
Outline

- Introduction
- Model Explanation Methods
- Experiments
- Result
- Analysis

Outline

- **Introduction**
- Model Explanation Methods
- Experiments
- Result
- Analysis

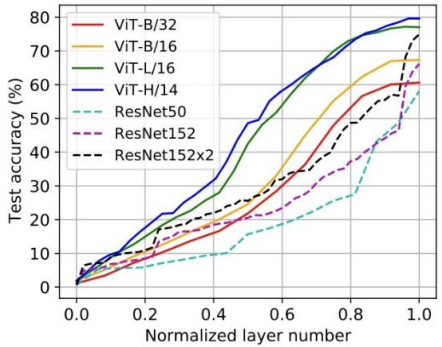
Motivation



$\boldsymbol{x}$
"panda"
57.7% confidence

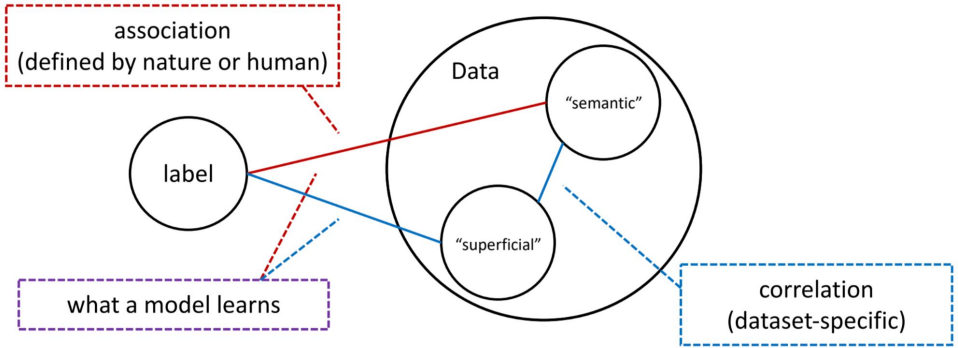
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

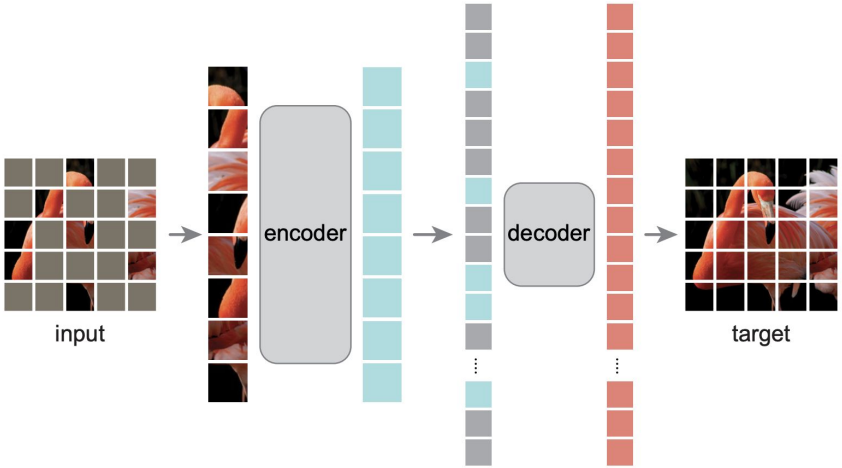


(b) ViTs vs ResNets

BERT
MAE
CLIP
DALLE



MAE



Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, Ross Girshick; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16000-16009

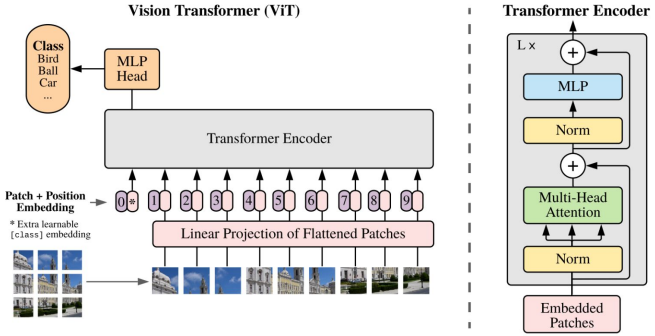


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

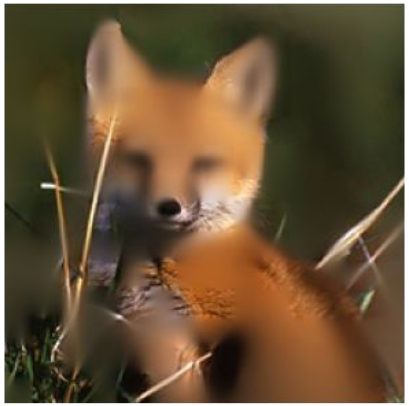
MAE

original

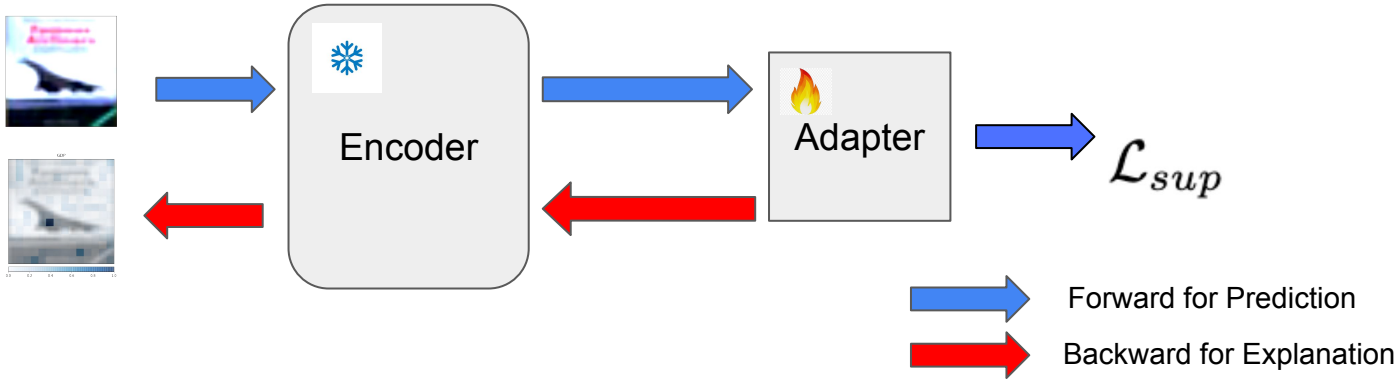
masked

reconstruction

reconstruction + visible



Finetune Structure



$$\mathcal{L}_{sup} = -\frac{1}{|X_{lab}|} \sum_{x \in X_{lab}} y_c log(p_c)$$

Outline

- Introduction
- **Model Explanation Methods**
- Experiments
- Result
- Analysis

Model Explaination

- **Goal:** to understand the mechanism and properties of the learnt representation from finetuned MAE model.
- **Model explanation**: Saliency, GradCAM, DeepLIFT
- **Implementation**: Captum

Saliency Maps

The saliency map of a input image specifies parts of it that contribute the most to the NN activity

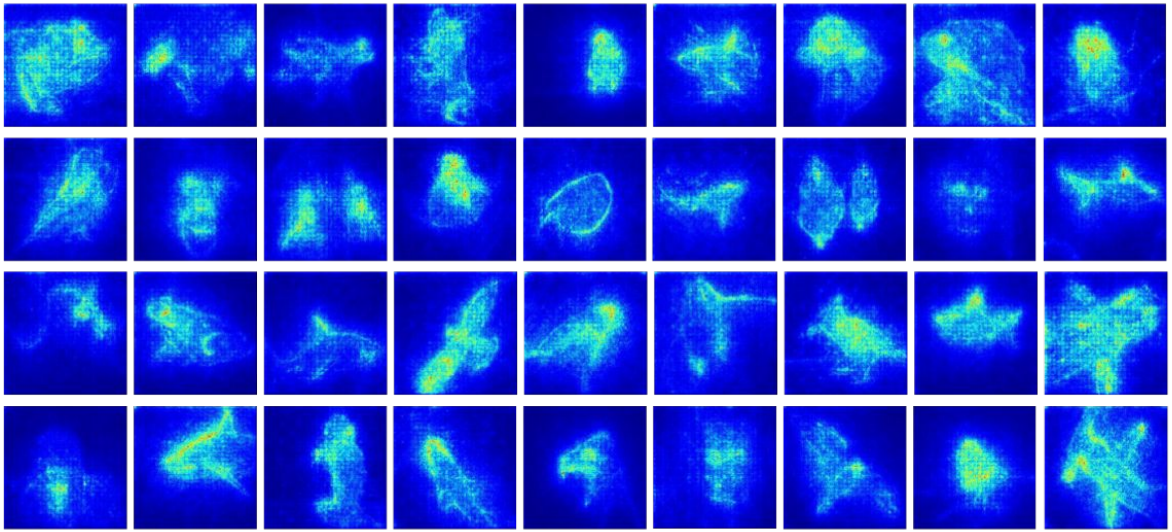
Using Backpropagation to compute the gradient with respect to the input of the network.

$$\mathcal{I}(x; c) = |\nabla_x f_c(x)|$$

The backpropagation for getting saliency maps is done post-training.

Saliency Maps

$$\mathcal{I}(x; c) = |\nabla_x f_c(x)|$$



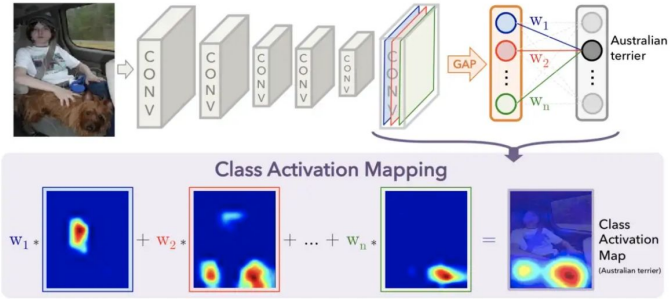
Saliency Map

GradCAM

CAM: average of weighted activation maps of the final convolution layer

Grad-CAM: more versatile

1. Compute the gradient of the logits of the class c w.r.t the activations maps of the final convolutional layer
2. The gradients are averaged across each feature map to give us an importance score
3. Multiply each activation map by its importance score and sum the values.

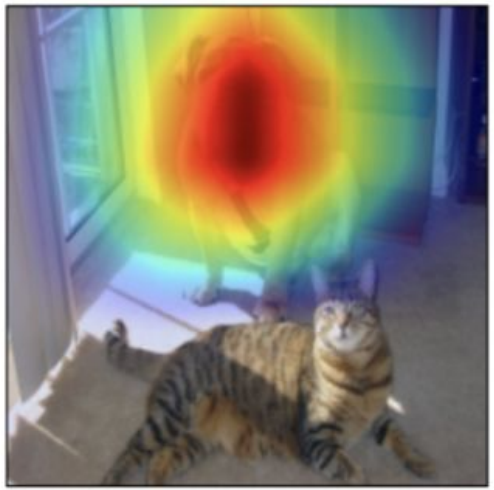


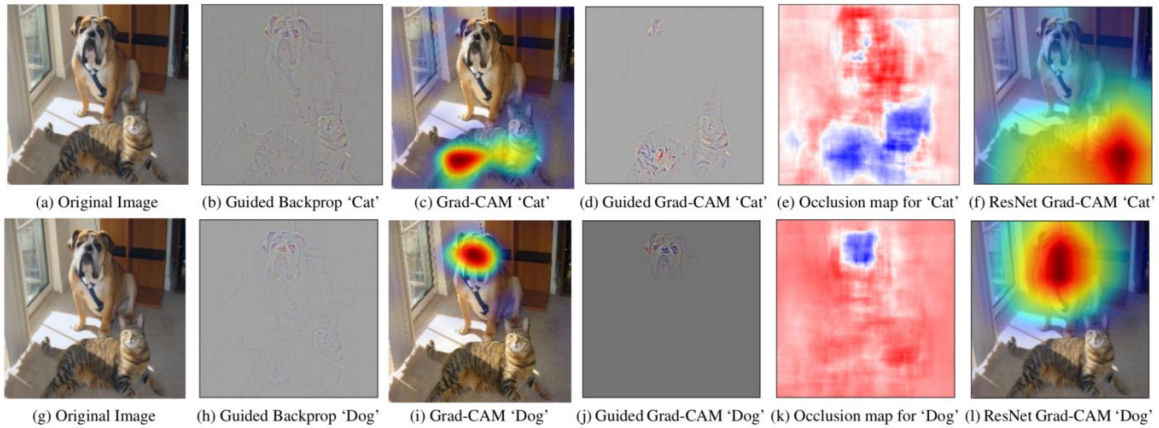
Class Activation Mapping. The figure is borrowed from [Zhou et al. 2016].

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

$$L_{\text{Grad-CAM}}^c = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

GradCAM





Grad-CAM

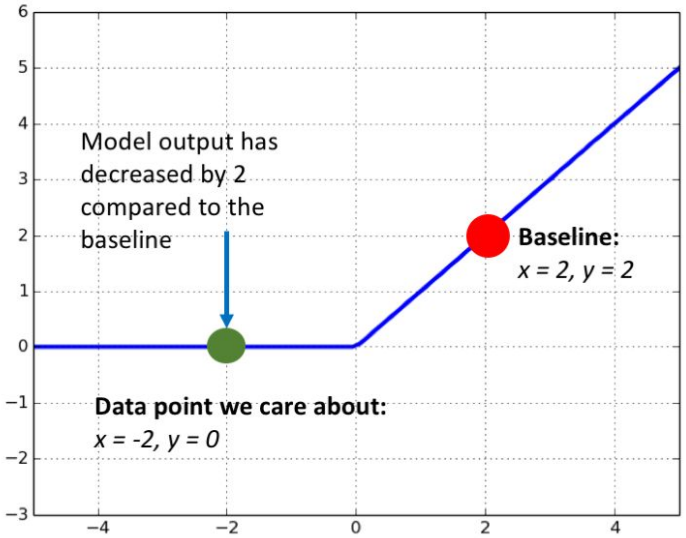
DeepLIFT

The backpropagation based approach is in the form:

$$w_i \times x_i = x_i \times \frac{\partial Y}{\partial x_i}$$

Failing cases: saturation, etc

$$y = ReLU(x) = max(0, x)$$



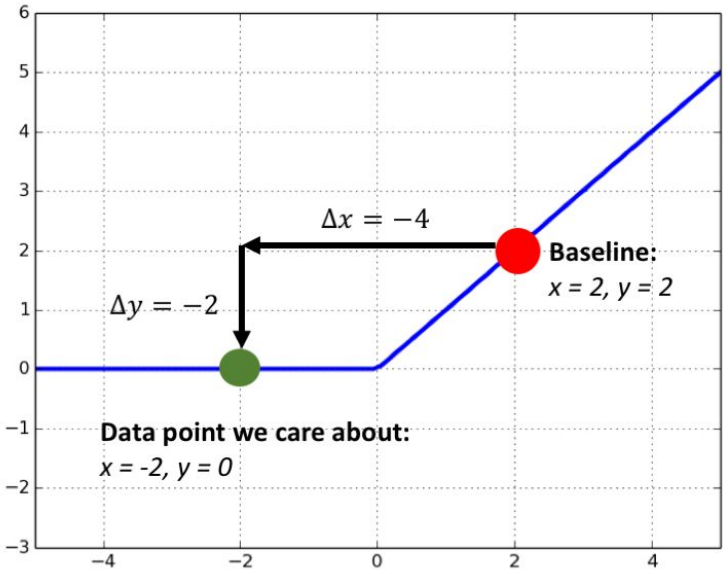
DeepLIFT: propagating activation differences

Choice of baseline is important

$$x_i \times \frac{\partial Y}{\partial x_i} \rightarrow (x_i - x_i^{baseline}) \times \frac{Y - Y^{baseline}}{x_i - x^{baseline}}$$

DeepLIFT

$$y = ReLU(x) = max(0, x)$$

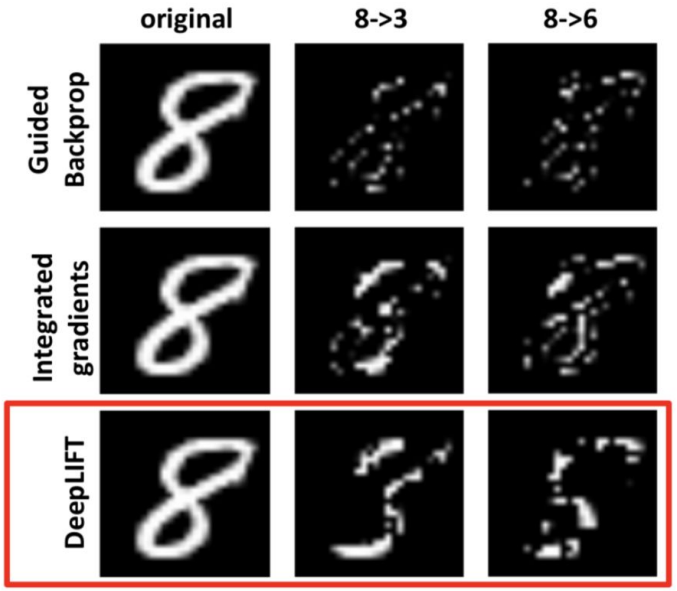


1. Calculating the slope

$$\frac{\Delta y}{\Delta x} = \frac{-2}{-4} = 0.5$$

2. Finding the feature import

$$\Delta x \times \frac{\Delta y}{\Delta x} = -4 \times 0.5 = -2$$



Outline

- Introduction
- Model Explanation Methods
- **Experiments**
- Result
- Analysis

Metric

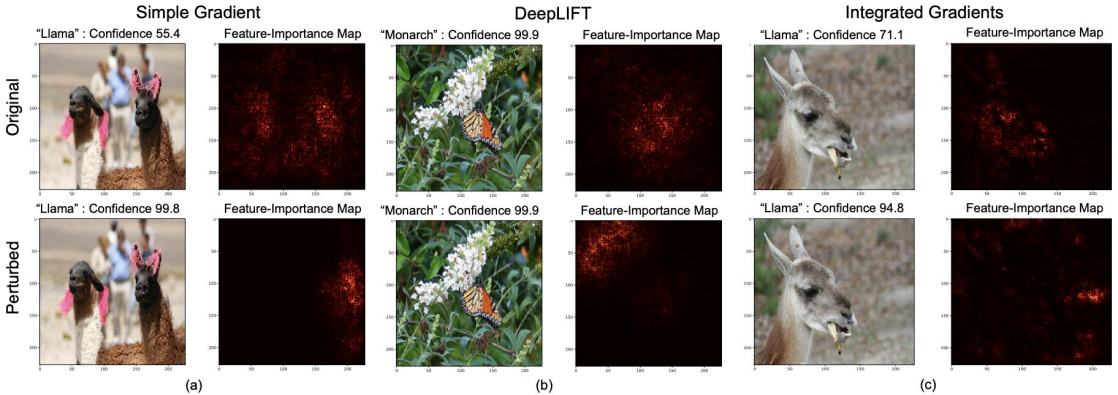


Figure 1: **Adversarial attack against feature-importance maps.** We generate feature-importance scores, also called saliency maps, using three popular interpretation methods: (a) simple gradients, (b) DeepLIFT, and (c) integrated gradients. The **top row** shows the the original images and their saliency maps and the **bottom row** shows the perturbed images (using the center attack with $\epsilon = 8$, as described in Section 3) and corresponding saliency maps. In all three images, the predicted label does not change from the perturbation; however, the saliency maps of the perturbed images shifts dramatically to features that would not be considered salient by human perception.

Yeh C K, Hsieh C Y, Suggala A, et al. On the (in) fidelity and sensitivity of explanations[J]. Advances in Neural Information Processing Systems, 2019, 32.

Ghorbani A, Abid A, Zou J. Interpretation of neural networks is fragile[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 3681-3688.

Algorithm 1 Iterative feature importance Attacks

Input: test image $\boldsymbol{x}_t$, maximum norm of perturbation ϵ, normalized feature importance function $\boldsymbol{I}(\cdot)$, number of iterations P, step size α

Define a dissimilarity function D to measure the change between interpretations of two images:

$$D(\boldsymbol{x}_t, \boldsymbol{x}) = \begin{cases} -\sum_{i \in B} \boldsymbol{I}(\boldsymbol{x})_i & \text{for } \textbf{top-k} \text{ attack} \\ \sum_{i \in \mathcal{A}} \boldsymbol{I}(\boldsymbol{x})_i & \text{for } \textbf{targeted} \text{ attack} \\ ||\boldsymbol{C}(\boldsymbol{x}) - \boldsymbol{C}(\boldsymbol{x}_t)||_2 & \text{for } \textbf{mass-center} \text{ attack,} \end{cases}$$

where B is the set of the k largest dimensions of $\boldsymbol{I}(\boldsymbol{x}_t)$, $\mathcal{A}$ is the target region of the input image in targeted attack, and $\boldsymbol{C}(\cdot)$ is the center of feature importance mass[a].

Initialize $\boldsymbol{x}^0 = \boldsymbol{x}_t$

for $p \in \{1, \ldots, P\}$ **do**

Perturb the test image in the direction of signed gradient[b] of the dissimilarity function:

$$\boldsymbol{x}^p = \boldsymbol{x}^{p-1} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} D(\boldsymbol{x}_t, \boldsymbol{x}^{p-1}))$$

If needed, clip the perturbed input to satisfy the norm constraint: $||\boldsymbol{x}^p - \boldsymbol{x}_t||_\infty \leq \epsilon$

end for

Among $\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^P\}$, return the element with the largest value for the dissimilarity function and the same prediction as the original test image.

[a] The center of mass is defined for a $W \times H$ image as: $\boldsymbol{C}(\boldsymbol{x}) = \sum_{i \in \{1,\ldots,W\}} \sum_{j \in \{1,\ldots,H\}} \boldsymbol{I}(\boldsymbol{x})_{i,j} [i, j]^T$

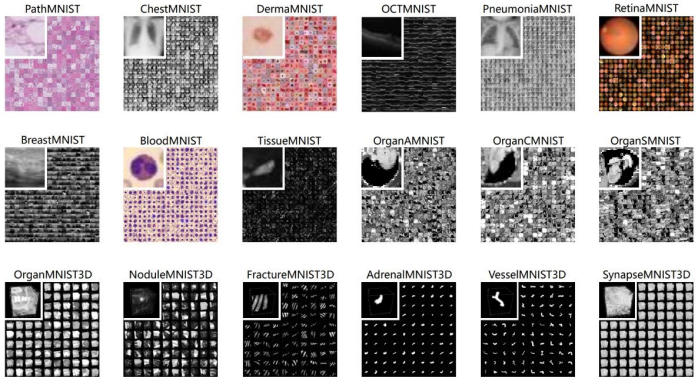
[b] In ReLU networks, this gradient is 0. To attack interpretability in such networks, we replace the ReLU activation with its smooth approximation (softplus) when calculating the gradient and generate the perturbed image using this approximation. The perturbed images that result are effective adversarial attacks against the original ReLU network, as discussed in Section 4.

CIFAR10

The CIFAR-10 dataset consists of 60000 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. In our experiment, considering the limitation of computing resources, we only use 1000 images for training data.

MedMNIST (ISBI 2021)

We introduce *MedMNIST v2*, a large-scale MNIST-like collection of standardized biomedical images, including 12 datasets for 2D and 6 datasets for 3D. All images are pre-processed into 28x28 (2D) or 28x28x28 (3D) with the corresponding classification labels, so that no background knowledge is required for users. Covering primary data modalities in biomedical images, MedMNIST v2 is designed to perform classification on lightweight 2D and 3D images with various data scales (from 100 to 100,000) and diverse tasks (binary/multi-class, ordinal regression and multi-label). The resulting dataset, consisting of 708,069 2D images and 9,998 3D images in total, could support numerous research / educational purposes in biomedical image analysis, computer vision and machine learning. We benchmark several baseline methods on MedMNIST v2, including 2D / 3D neural networks and open-source / commercial AutoML tools.



For more details, please refer to our paper:

MedMNIST v2: A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification (arXiv)

MedMNIST2D

MedMNIST2D	Data Modality	Tasks (# Classes/Labels)	# Samples	# Training / Validation / Test
PathMNIST	Colon Pathology	Multi-Class (9)	107,180	89,996 / 10,004 / 7,180
ChestMNIST	Chest X-ray	Multi-Label (14) Binary-Class (2)	112,120	78,468 / 11,219 / 22,433
DermaMNIST	Dermatoscope	Multi-Class (7)	10,015	7,007 / 1,003 / 2,005
OCTMNIST	Retinal OCT	Multi-Class (4)	109,309	97,477 / 10,832 / 1,000
PneumoniaMNIST	Chest X-Ray	Binary-Class (2)	5,856	4,708 / 524 / 624
RetinaMNIST	Fundus Camera	Ordinal Regression (5)	1,600	1,080 / 120 / 400
BreastMNIST	Breast Ultrasound	Binary-Class (2)	780	546 / 78 / 156
BloodMNIST	Blood Cell Microscope	Multi-Class (8)	17,092	11,959 / 1,712 / 3,421
TissueMNIST	Kidney Cortex Microscope	Multi-Class (8)	236,386	165,466 / 23,640 / 47,280
OrganAMNIST	Abdominal CT	Multi-Class (11)	58,850	34,581 / 6,491 / 17,778
OrganCMNIST	Abdominal CT	Multi-Class (11)	23,660	13,000 / 2,392 / 8,268
OrganSMNIST	Abdominal CT	Multi-Class (11)	25,221	13,940 / 2,452 / 8,829

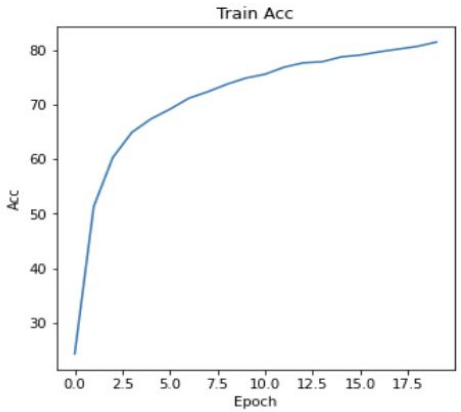
https://github.com/MedMNIST/MedMNIST

(Maybe) A Benchmark for Transfer/Multi-task Learning

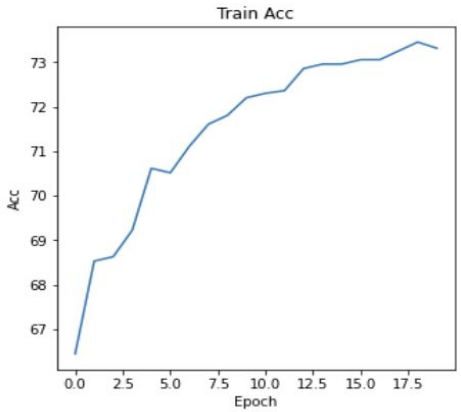
Outline

- Introduction
- Model Explanation Methods
- Experiments
- **Result**
- Analysis

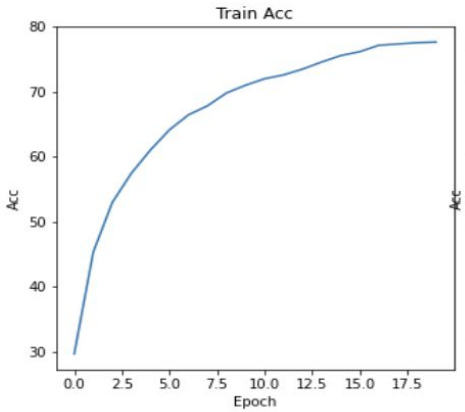
Training Curves



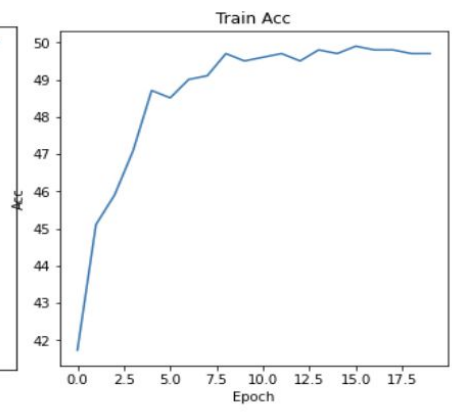
CIFAR10



DermaMNIST

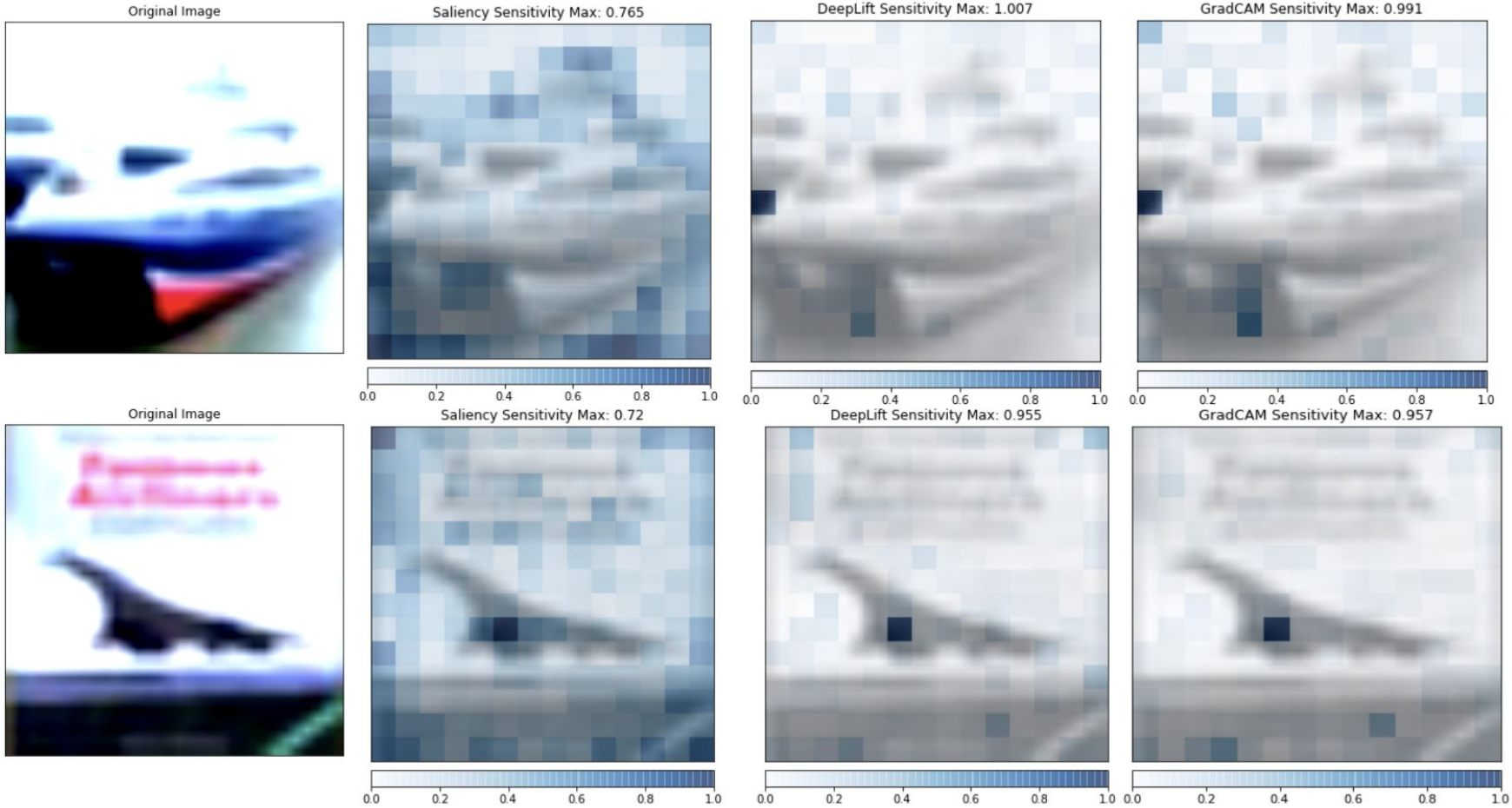


BloodMNIST

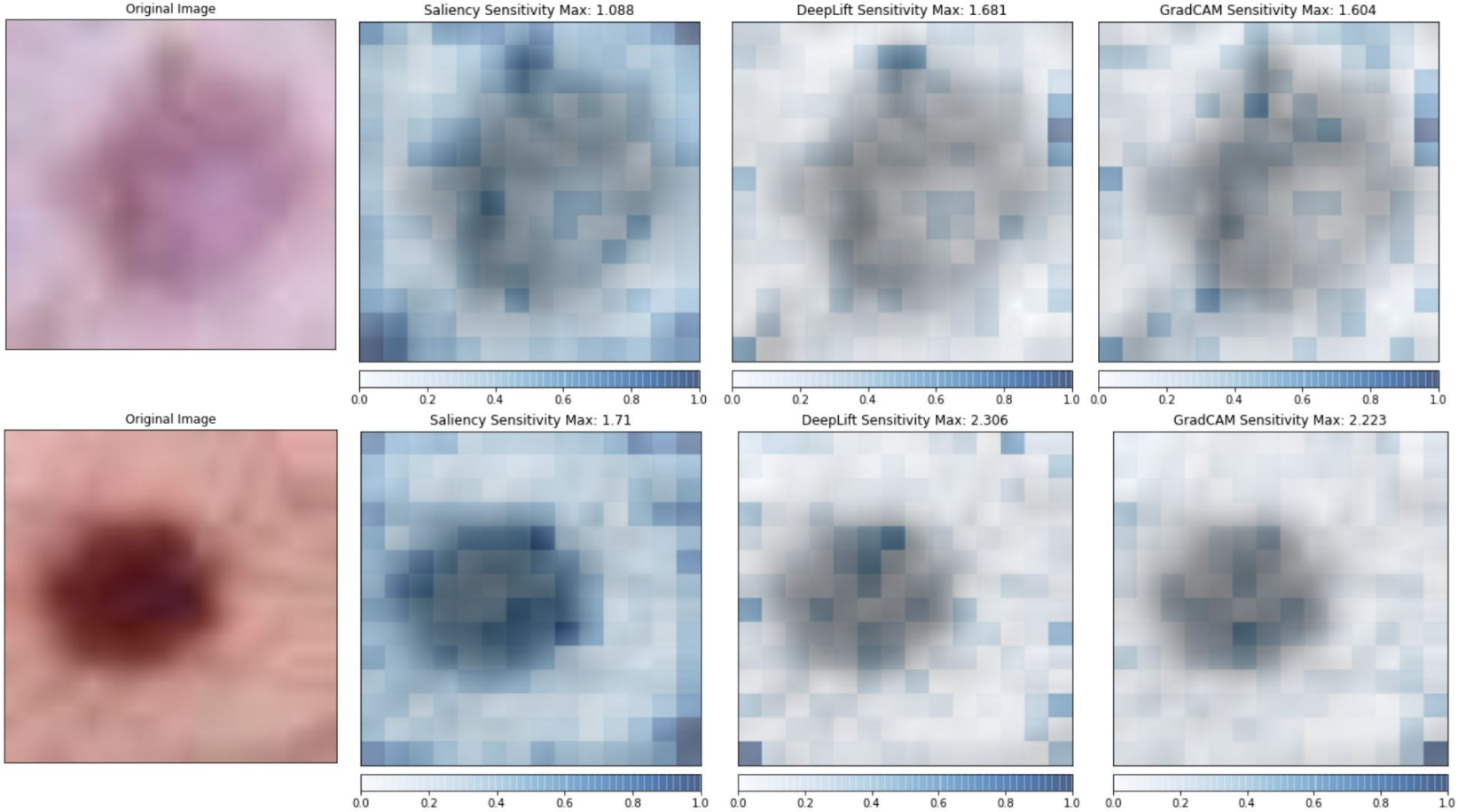


RetinaMNIST

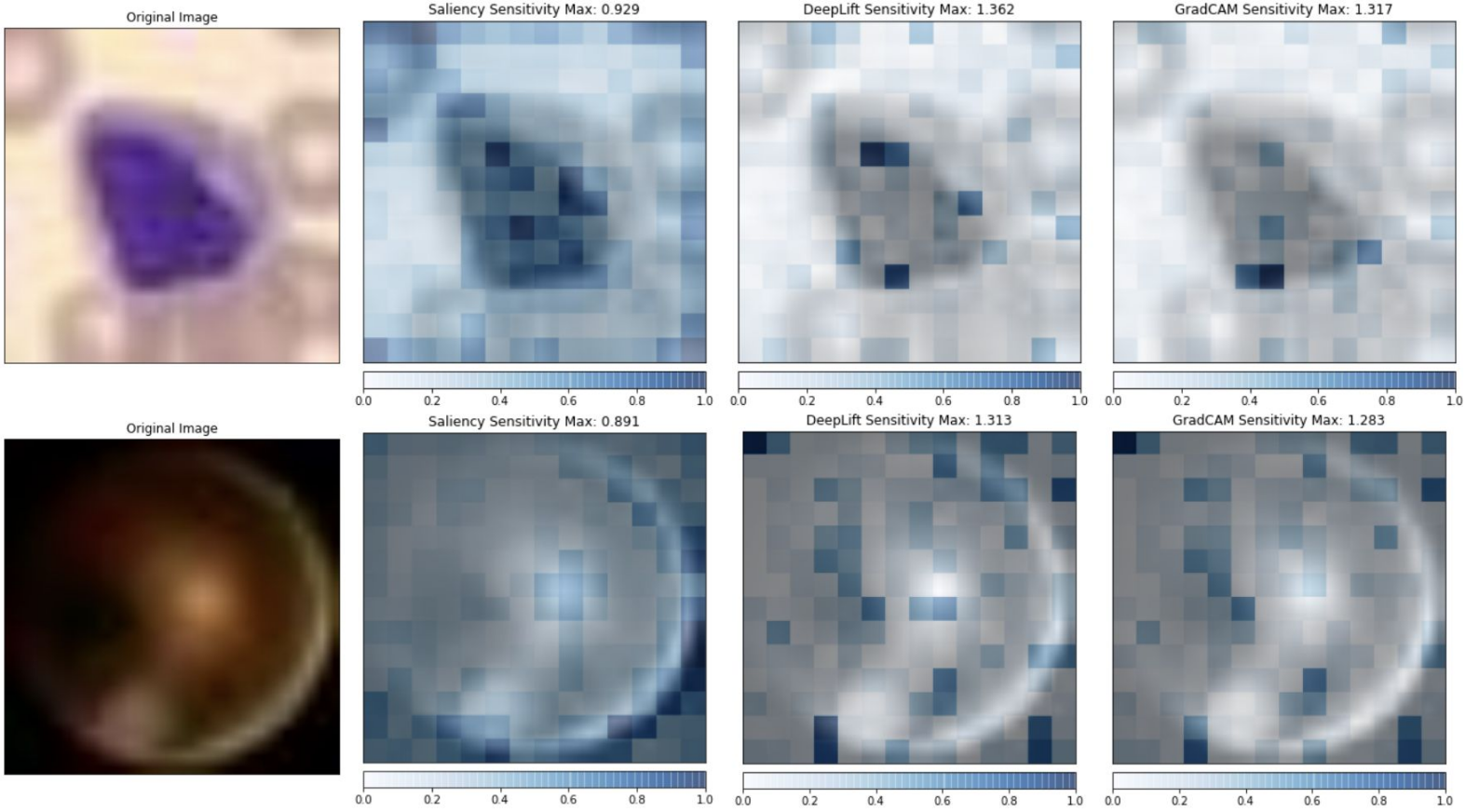
Results on CIFAR10



Results on DermaMNIST



Results on BloodMNIST and DermaMNIST



Future Work

1. Explanation Network for Large-scale Pretrained Model
2. Patch-wise Robustness/Explanation of ViT

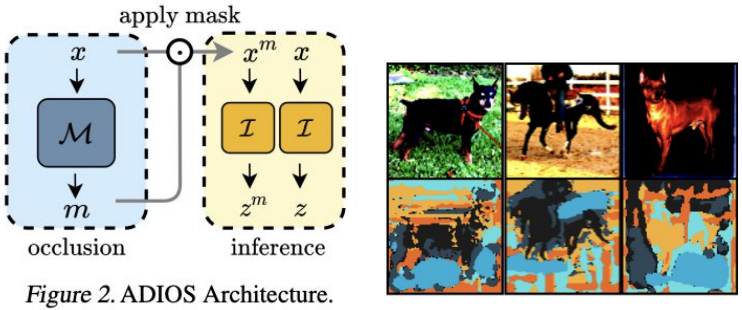
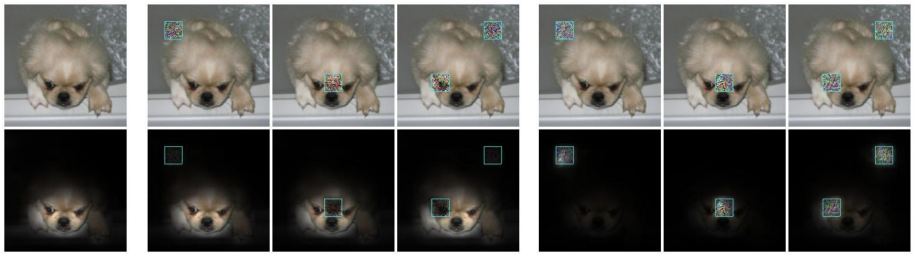


Figure 2. ADIOS Architecture.

Shi, Yuge, et al. "Adversarial masking for self-supervised learning." *International Conference on Machine Learning*. PMLR, 2022.



(a) Clean Image (b) with Naturally Corrupted Patch (c) with Adversarial Patch

Gu, Jindong, Volker Tresp, and Yao Qin. "Are vision transformers robust to patch perturbations?." *European Conference on Computer Vision*. Springer, Cham, 2022