Interpretation of 3D CNNs for Brain MRI Data Classification

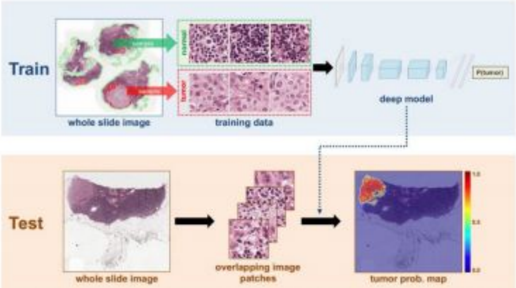
What is Interpretability?

AlphaGo vs. Lee Sedol

ImageNet Challenge

Self-driving Cars

Disease Diagnosis



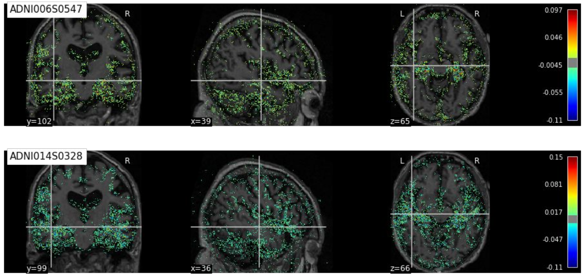
Neural Machine Translation

& More to Come!

Slides by Beomsu Kim, KAIST

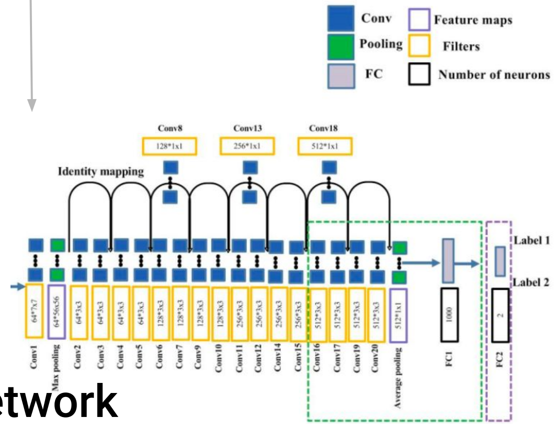
What is Interpretability?

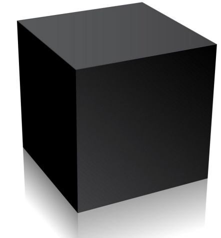
Large datasets



Task solving

Computing power





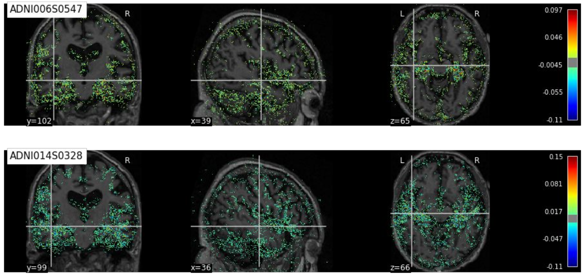
Implicit Information

Deep Neural Network

What is Interpretability?

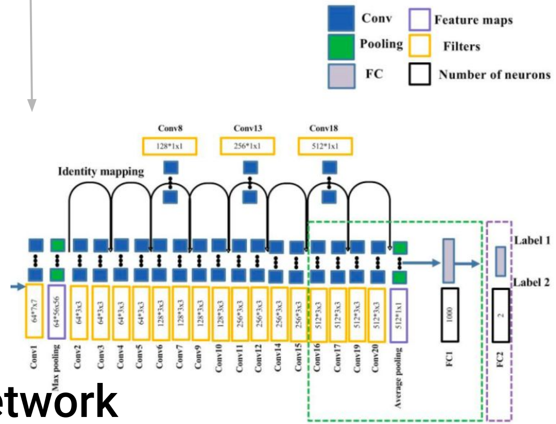
Large datasets

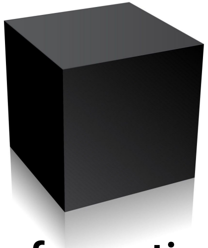
Interpretable information



Task solving

Computing power





Implicit Information

Deep Neural Network

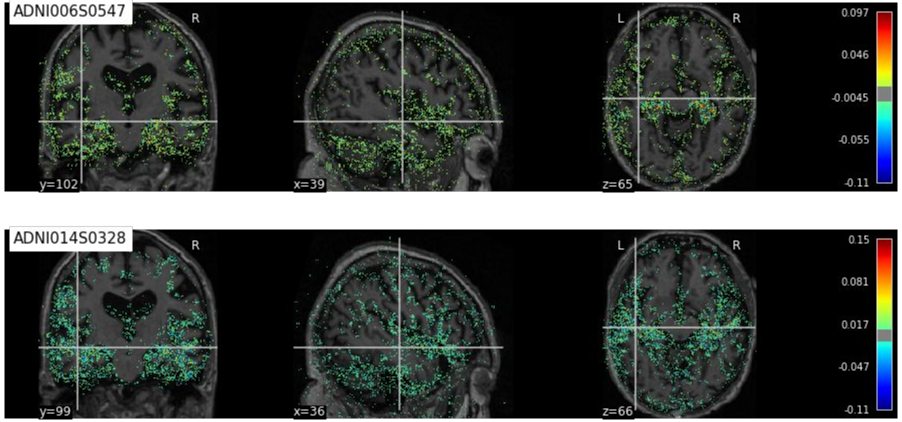
Slides by Beomsu Kim, KAIST

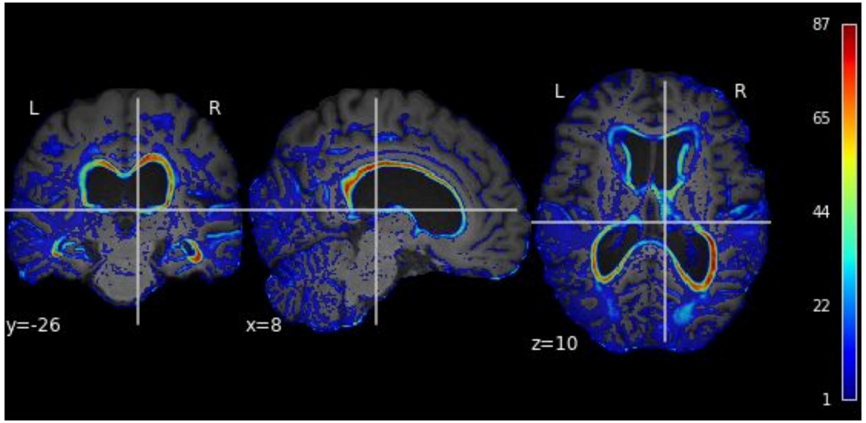
What is Interpretability?

Interpretability in Healthcare

1. Verify that model works as expected

Wrong decisions can be costly and dangerous



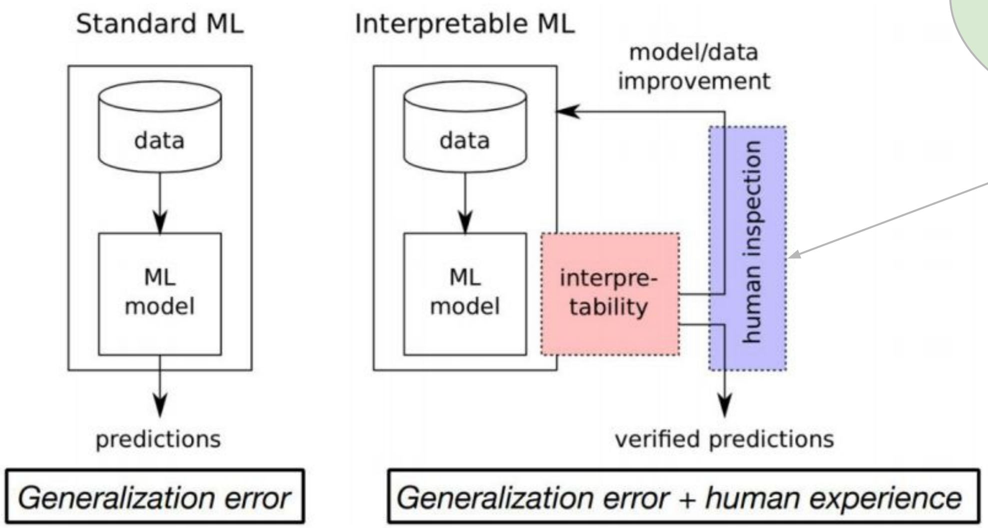


Disease Misclassification

What is Interpretability?

Interpretability in Healthcare

2. Improve / Debug classifier



Interpretation should be in the language of a specialist too

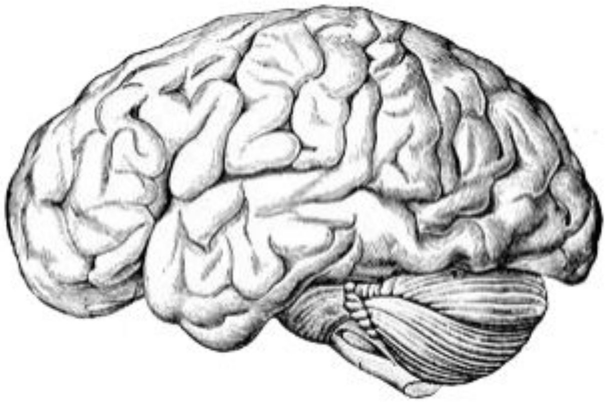
MIT: https://mit6874.github.io/assets/sp2021/slides/l05.pdf

What is Interpretability?

Interpretability in Healthcare

3. Make new discoveries

Learn about the human brain/ biological / chemical mechanisms



MIT: https://mit6874.github.io/assets/sp2021/slides/l05.pdf

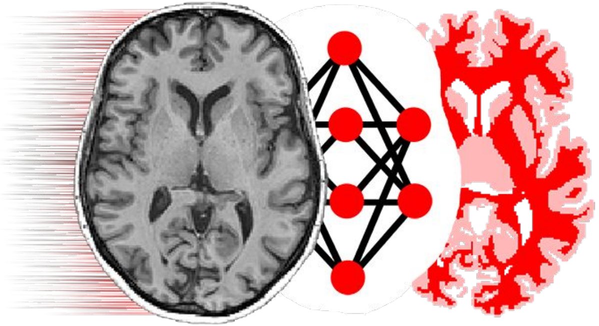
Interpretation

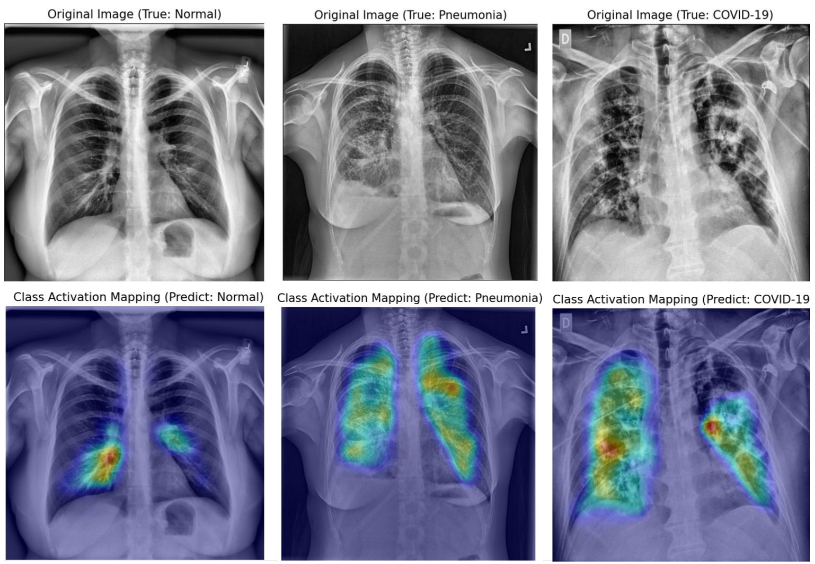
Interpretation was defined as mapping an abstract concept like the output class into a domain example, while **explanation** was defined as a set of domain features such as pixels of an image the contribute to the output decision of the model.

PROBLEM MOTIVATION

Problem: existing methods of interpretation have not reached a sufficient level of trust to be applied as support and decision-making systems.

- ➢ Lack of evaluation of interpretation methods
- ➢ Interpretation of models towards understanding medical/biological mechanism is still far from reach.





Types of Interpretability in ML

Ante-hoc Interpretability

Choose an interpretable model and train it.

Ex.

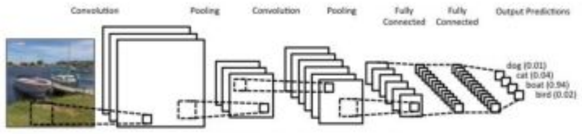
Decision Tree

Problem. Is the model expressive enough to predict the data?

Post-hoc Interpretability

Choose a complex model and develop a special technique to interpret it.

Ex.



Deep Neural Networks

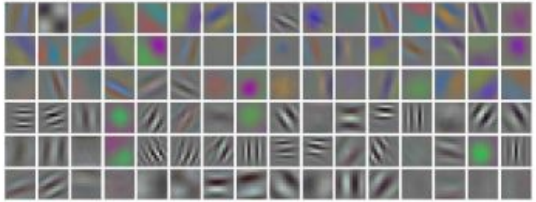
Problem. How to interpret millions of parameters?

Types of Post-hoc Interpretability

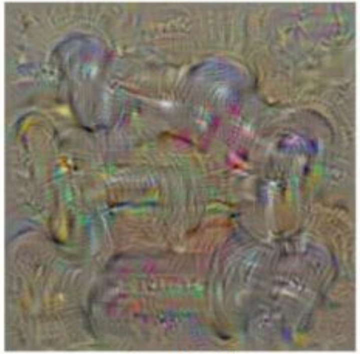
Post-hoc interpretability techniques
can be classified by degree of "locality"

Model ⟷ **Input**

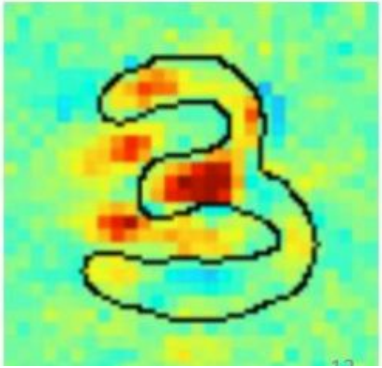
What representations have the DNN learned?



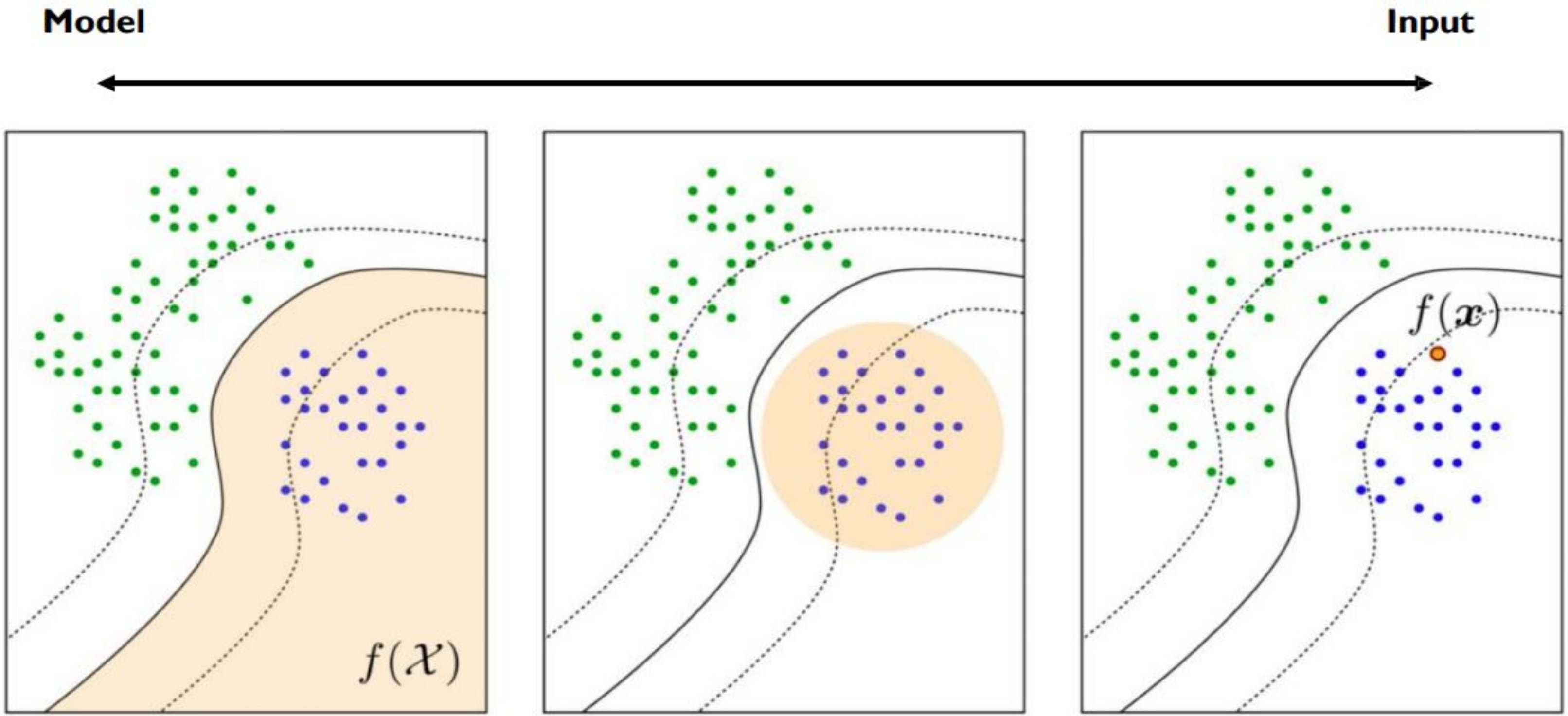
What pattern / image maximally activates a particular neuron?



Explain why input x has been classified as $f(x)$.



Types of Post-hoc Interpretability



Interpreting Deep Neural Networks

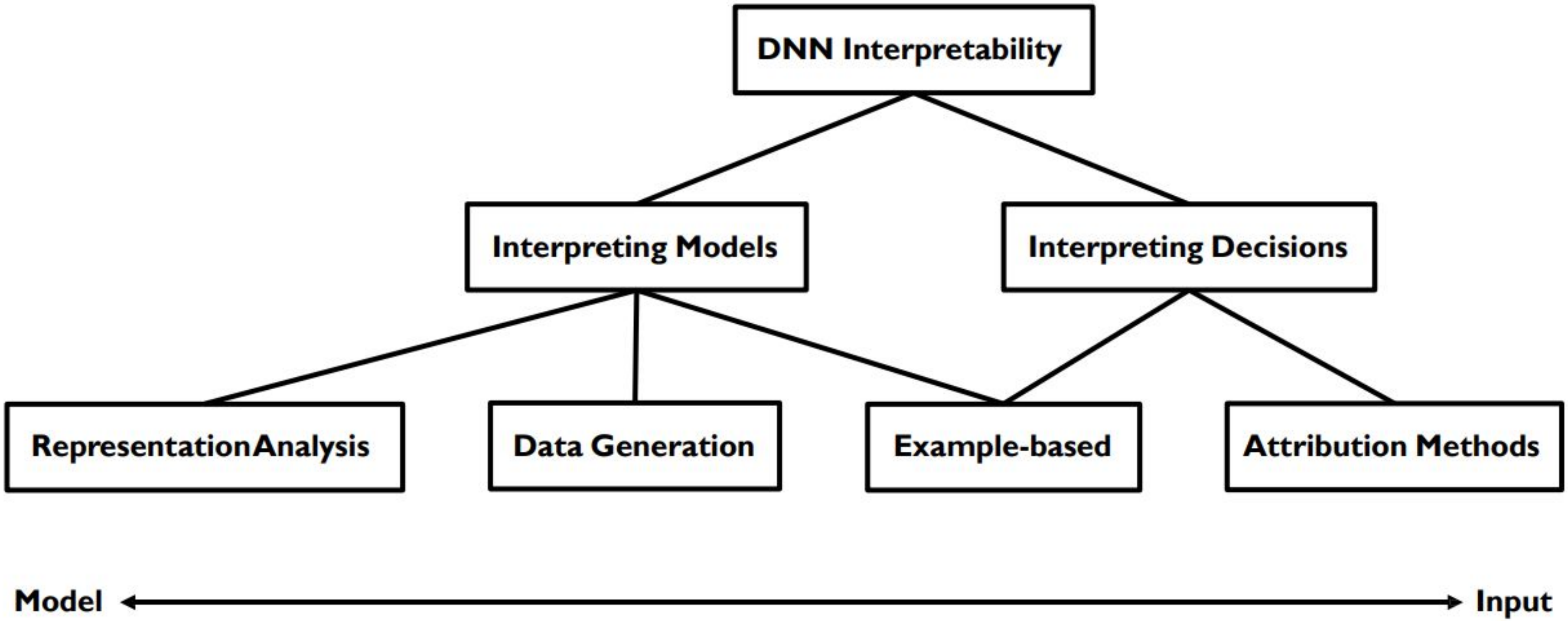
2a. **Interpreting Models** (macroscopic, understand internals) vs. **decisions** (microscopic, practical applications)

2b. **Interpreting Models**: Weight visualization, Surrogate model, Activation maximization, Example-based

2c. **Interpreting Decision**s:

- Example-based
- Attribution Methods: why are gradients noisy?
- Gradient-based Attribution: SmoothGrad, Interior Gradient
- Backprop-based Attribution: Deconvolution, Guided Backpropagation

Interpreting Deep Neural Networks

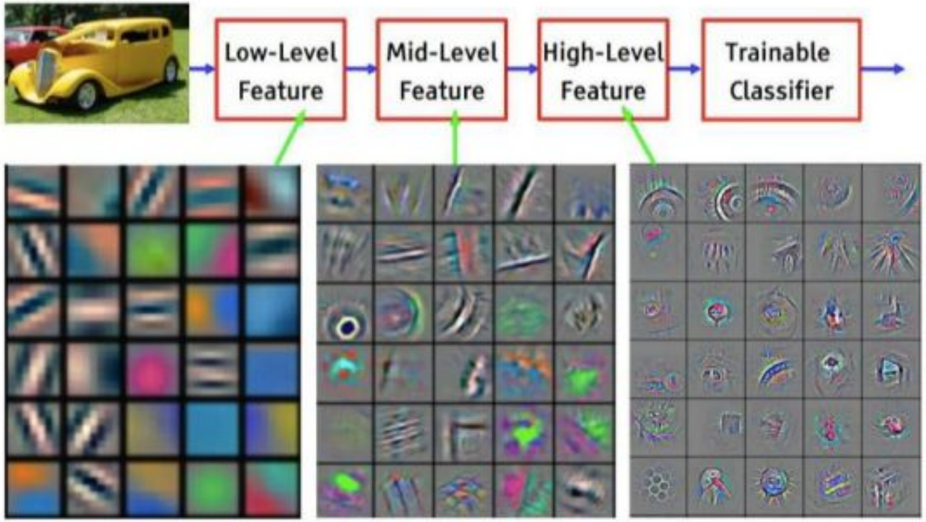


Slides by Beomsu Kim, KAIST

Interpreting models

- Representation analysis:

 - **Weight Visualization** (Filter visualization in CNN)

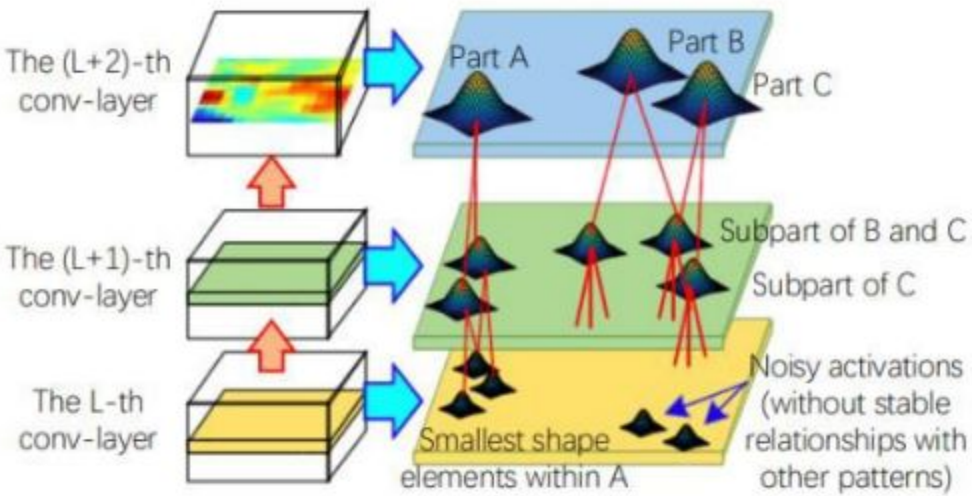


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Interpreting models

- Representation analysis:

- Weight Visualization
- **Surrogate Model** (train an Interpretable ML model on the Outputs of our "Black Box" model with the specific goal of interpreting it.)



Interpreting models

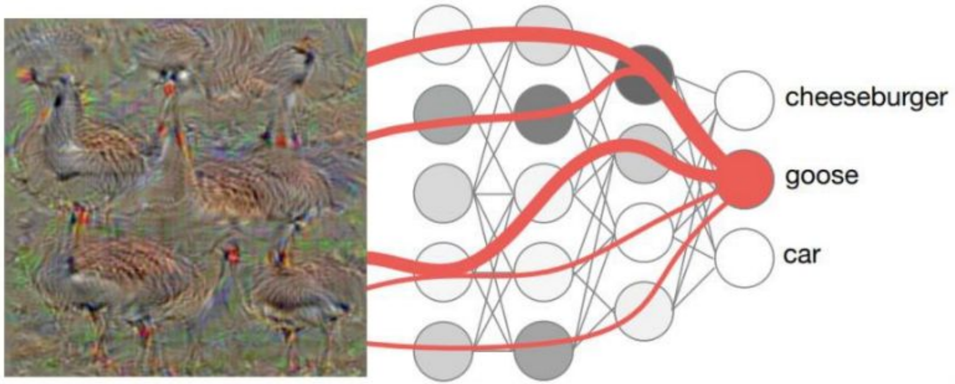
- Representation analysis:

- Weight Visualization
- Surrogate Model

- Data generation - activation maximization approach (finding patterns that maximize the activation of a neuron)

\+ builds typical patterns for given classes (e.g. beaks, legs), unrelated background objects are not present in the image

\- Does not resemble class-related patterns, lowers the quality of the interpretation for given classes



Find the most likely input pattern for a given class

Interpreting models

- Representation analysis:

- Weight Visualization
- Surrogate Model

- Data generation - activation maximization approach
- **Example-based** (Find image instances that represent / do not represent the image class)

Interpreting models

Limitation

Question:
What would be the best image to interpret the class “motorcycle”?

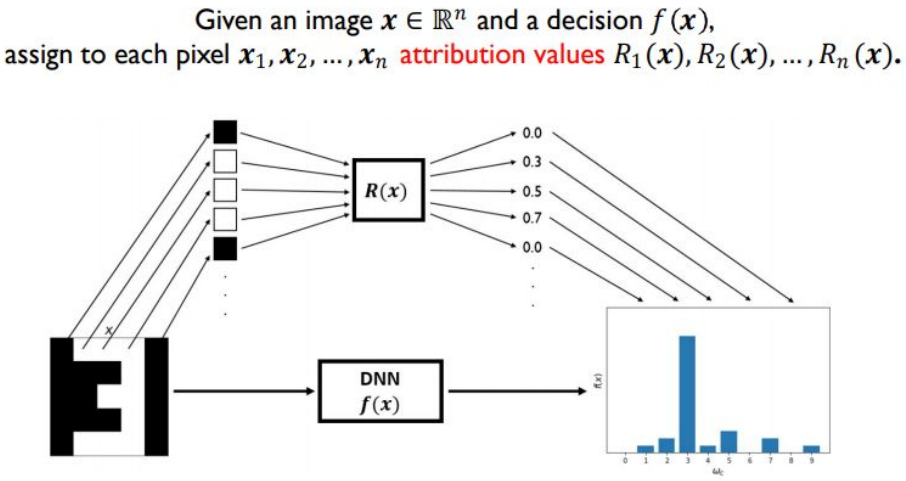
Interpreting decisions. Example-based

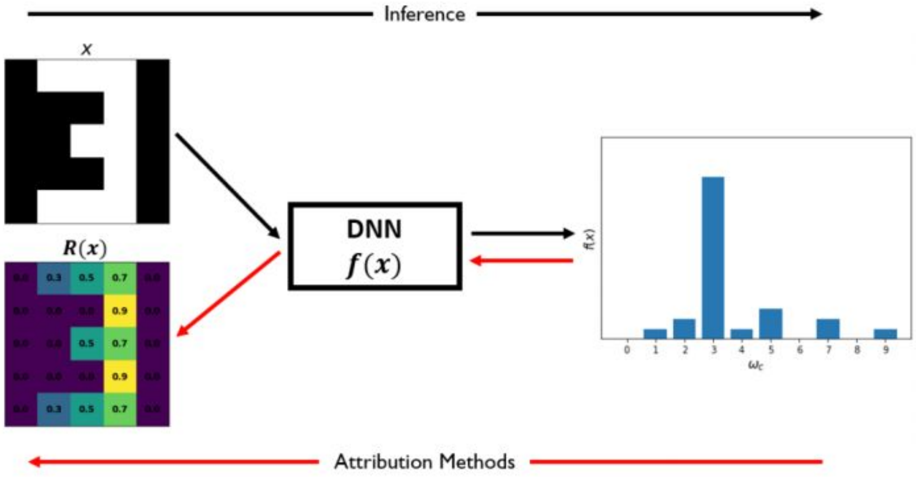
- Example-based

(Which training instance
influenced the decision most?)

Interpreting decisions. Attribution Methods

Heatmap visualization

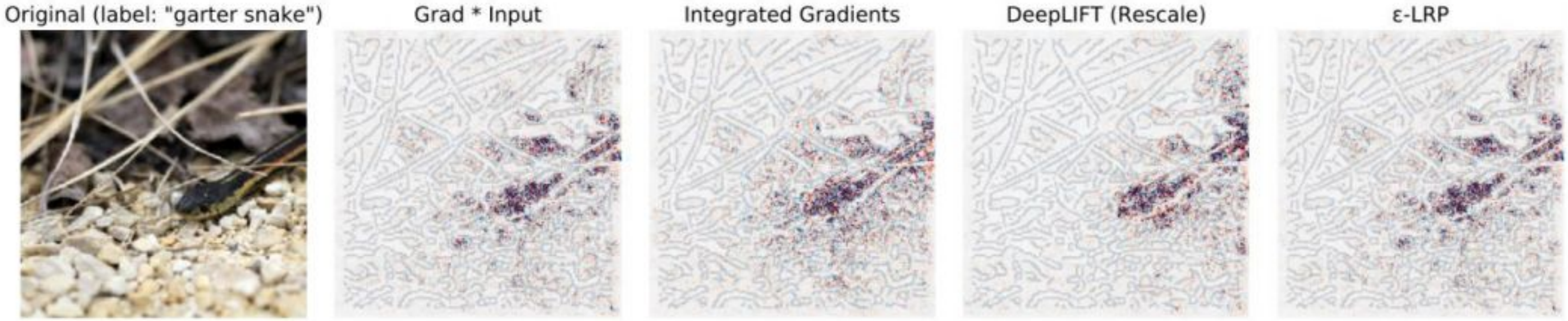




Interpreting decisions. Attribution Methods

$$Saliency(\boldsymbol{x}) \coloneqq \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$$

Attributions visualized as heatmaps



Interpreting decisions. Attribution Methods

Gradient-based Methods
- Add noise: Perturb the input x to $x*$ and use $\nabla_{x^*} f \; x^*$.
- Some methods take the average over the perturbation set $\{x_1^*, x_2^*, \ldots, x_n^*\}$.

Backprop-based Methods
- Modify the backpropagation algorithm.

Interpretation methods

Gradient-based backpropagation methods: Methods that backpropagate an importance signal from the output towards the input. *The common idea is to compute the gradient of the network's prediction with respect to the input, holding the weights fixed. This determines which input elements (e.g., which pixels in case of an input image) need to be changed the least to affect the prediction the most.*

Perturbation-based forward propagation methods: Methods that perturb the input and probe its possible effects on the prediction of the network.

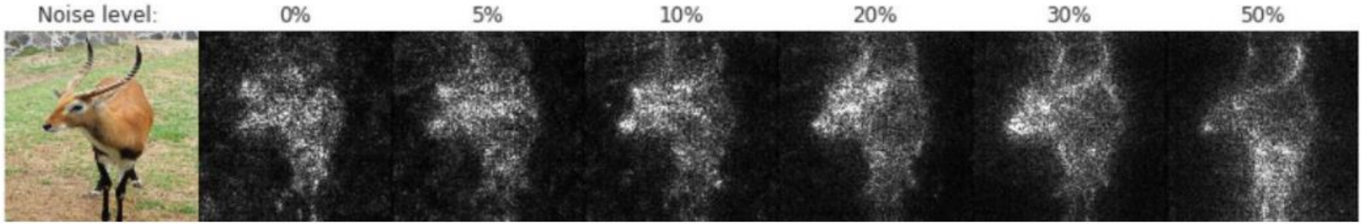
Interpreting decisions. Gradient-based Attribution

Integrated Gradients - Generate a linear interpolation between the baseline and the original image

Guided Integrated Gradients - minimizes noise by moving in the direction of lowest associated partial derivatives.

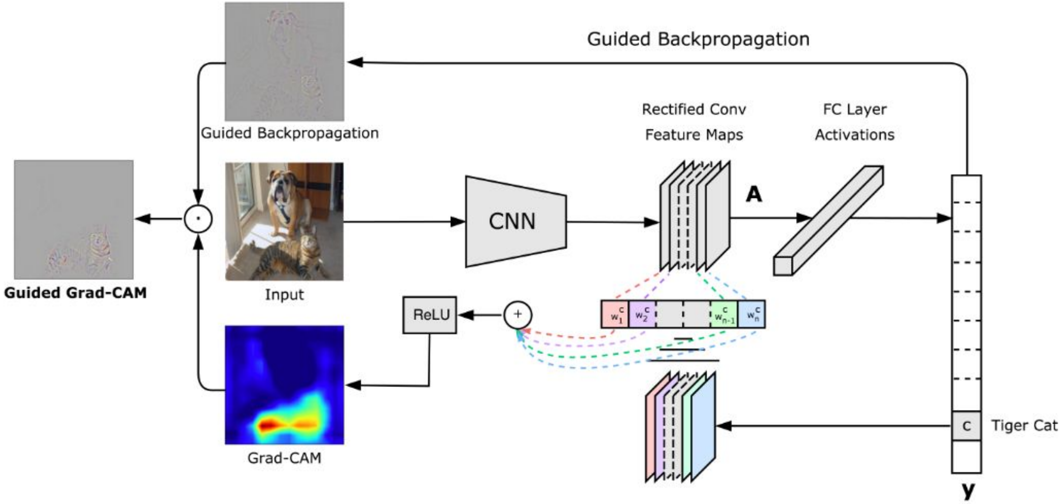
SmoothGrad -often significantly denoises this sensitivity mask. This technique adds pixel-wise Gaussian noise to many copies of the image, and simply averages the resulting gradients

$$SmoothGrad(\boldsymbol{x}) := \frac{1}{n}\int_{1}^{n} \frac{\partial f(\boldsymbol{x}^*)}{\partial \boldsymbol{x}^*}, \quad \boldsymbol{x}^* = \boldsymbol{x} + \mathcal{N}(0, \sigma^2)$$



Grad-CAM – Gradient-weighted Class Activation Mapping

The Idea: *to take the gradients of the target class flowing into the final convolutional layer to produce a heatmap highlighting the important regions in the image to predict the concept.*



Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization Ramprasaath R. Selvaraju ·https://arxiv.org/pdf/1610.02391.pdf

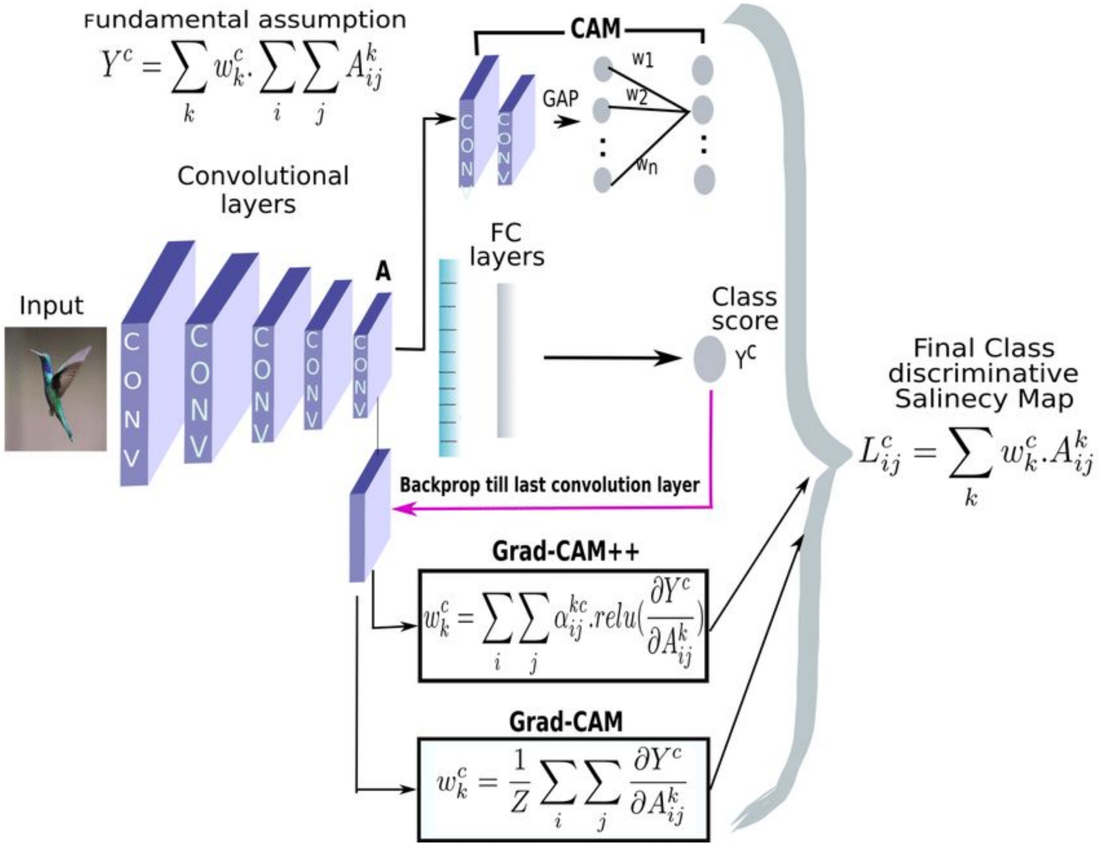
Algorithm

1. Select the class of interest (target class)
2. Calculate the gradient of the class logit nd the activation maps
3. Average over activations using the global average pooling
4. Obtain he neuron importance weights coefficients α k c for each map (this weight α c k represents a partial linearization of the deep network downstream from A, and captures the 'importance' of feature map k for a target class c)
5. Consider a linear combination
6. Apply ReLU
7. Interpolate the heat-map (increase the dimension)

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

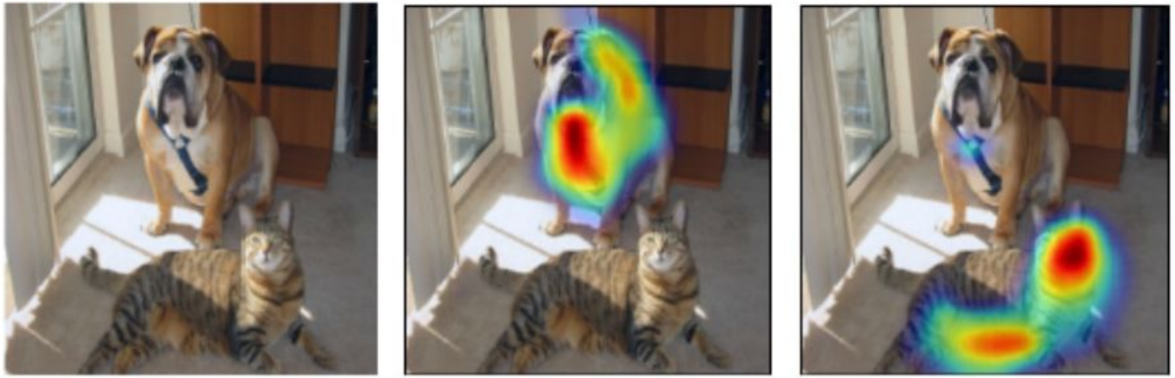
$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Grad-CAM



Fundamental assumption
$Y^c = \sum_k w_k^c . \sum_i \sum_j A_{ij}^k$
CAM
GAP
w1
w2
wn
CONV
Convolutional layers
FC layers
A
Input
Class score
Y^c
Final Class discriminative Salinecy Map
$L_{ij}^c = \sum_k w_k^c . A_{ij}^k$
Backprop till last convolution layer
Grad-CAM++
$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ij}^k})$
Grad-CAM
$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$

Grad-CAM – Gradient-weighted Class Activation Mapping



(a) Original Image (b) Cat Counterfactual exp (c) Dog Counterfactual exp

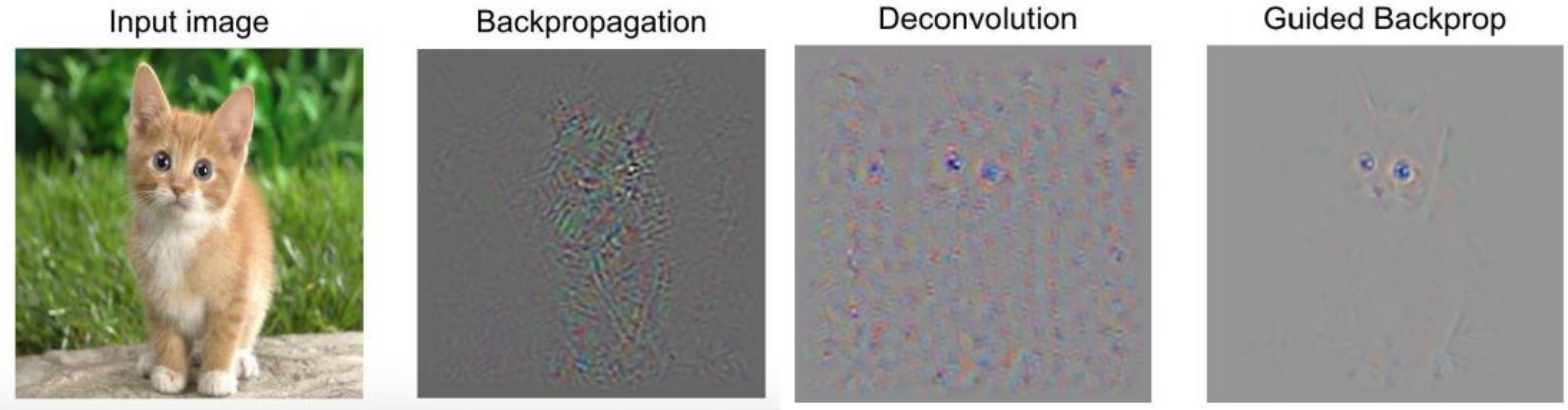
Guided Backprop

Idea: using gradient backpropagation as it is except at the ReLU stages. Guided Backpropagation basically combines vanilla backpropagation and DeconvNets when handling the ReLU nonlinearity:

- Like DeconvNets, in Guided Backpropagation we only backpropagate positive error signals – i.e. we set the negative gradients to zero (ref). This is the application of the ReLU to the error signal itself during the backward pass.
- Like vanilla backpropagation, we also restrict ourselves to only positive inputs.

Thus, the gradient is “guided” by both the input and the error signal.

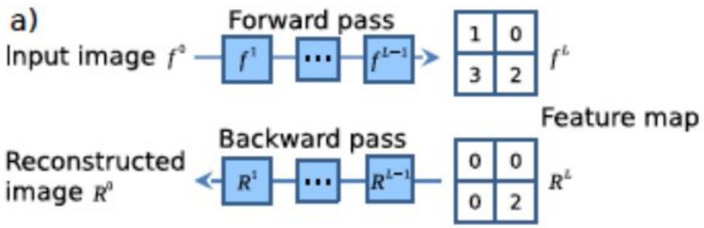
Guided Backprop



Observation: Removing more gradient leads to sharper visualizations

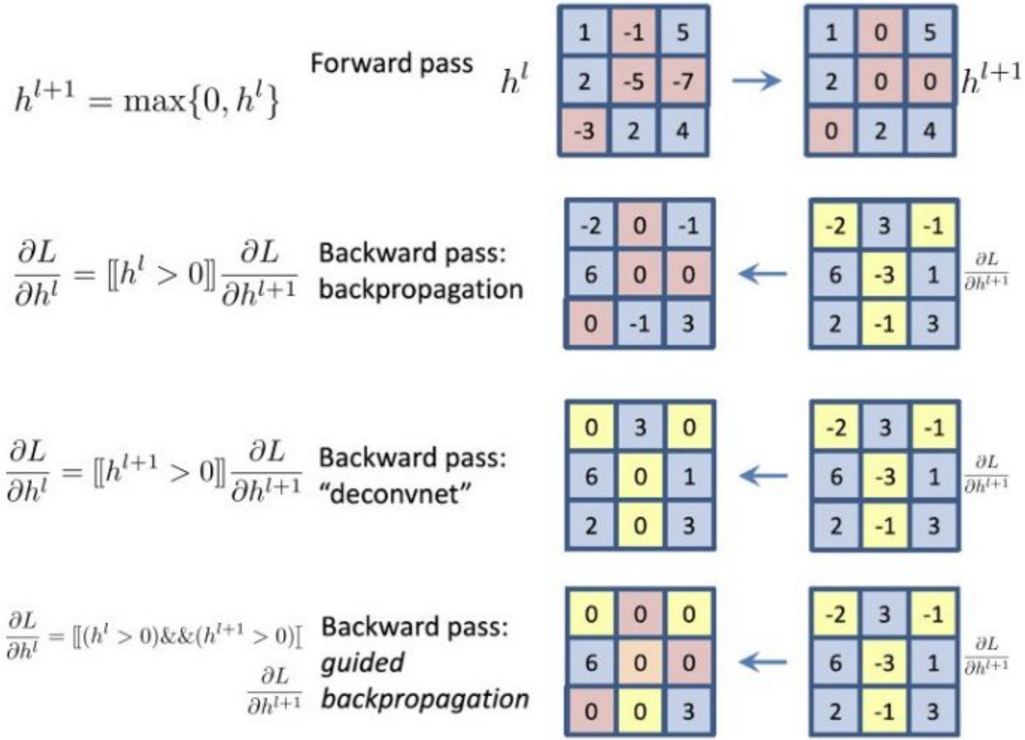
Guided Backprop

1) Select the layer we want to render; Select the activation on the feature map of this layer in order to find out which area on the image it responds to, then nullify the rest of the activations.
2) run the resulting feature map in DeconvNet: first we do unpooling,then rectification and then apply the transposed convolutional filter corresponding to this convolutional layer in the original network to reconstruct the activations in the previous layer that led to this activation.
3) Thus, at the output, we get those pixels that influenced the activation of the selected neuron.



Guided Backprop

1. Before applying ReLU, there are negative values on the feature map in some places, after ReLU they will be zero.
2. Then, on the backward pass in the reverse relu, we will zero the values in the same places as in the forward (while some values may remain negative and spread further, and some positive ones will vanish).
3. The second reverse relu operation in the guided backprob is essentially a regular relu. That is, having a feature map, when it propagates, we will zero out negative values.and leave positive.

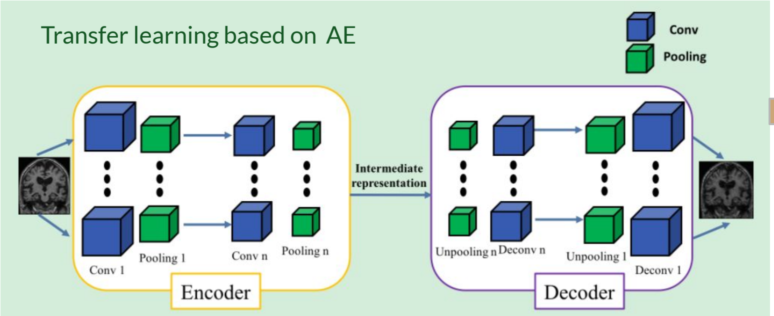


Meaningful Perturbation

The aim of saliency is to identify which regions of an image x_0 are used by the black box to produce the output value $f(x_0)$.
The idea: observing how the value of f(x) changes as x is obtained "deleting" different regions R of x_0.

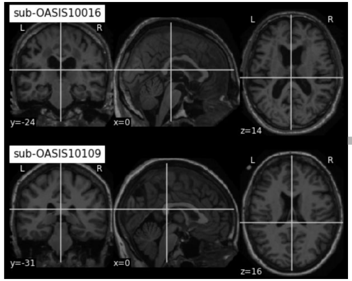
$$[\Phi(x_0; m)](u) = \begin{cases} m(u)x_0(u) + (1 - m(u))\mu_0, & \text{constant}, \\ m(u)x_0(u) + (1 - m(u))\eta(u), & \text{noise}, \\ \int g_{\sigma_0 m(u)}(v - u)x_0(v)\, dv, & \text{blur}, \end{cases}$$

METHODS

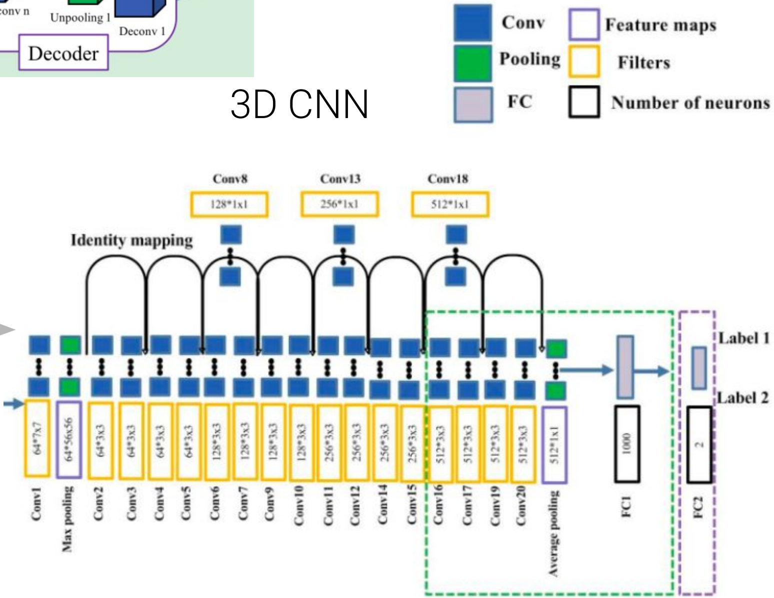


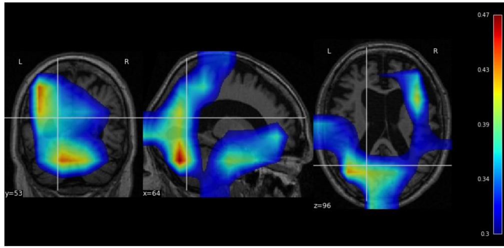
Baseline 3D CNN

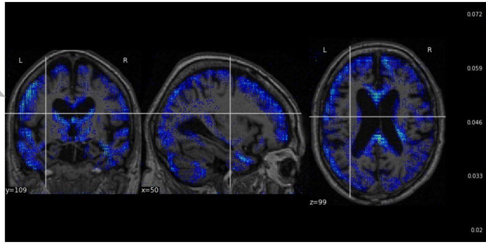
3D CNN + tranf.learning



3D CNN



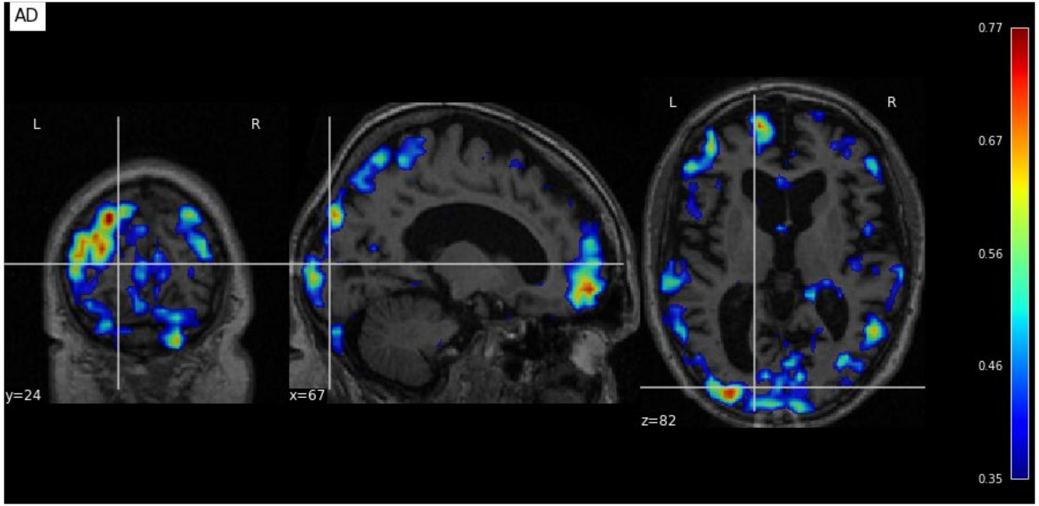




Meaningful Perturbation

ADNI - AD/CN

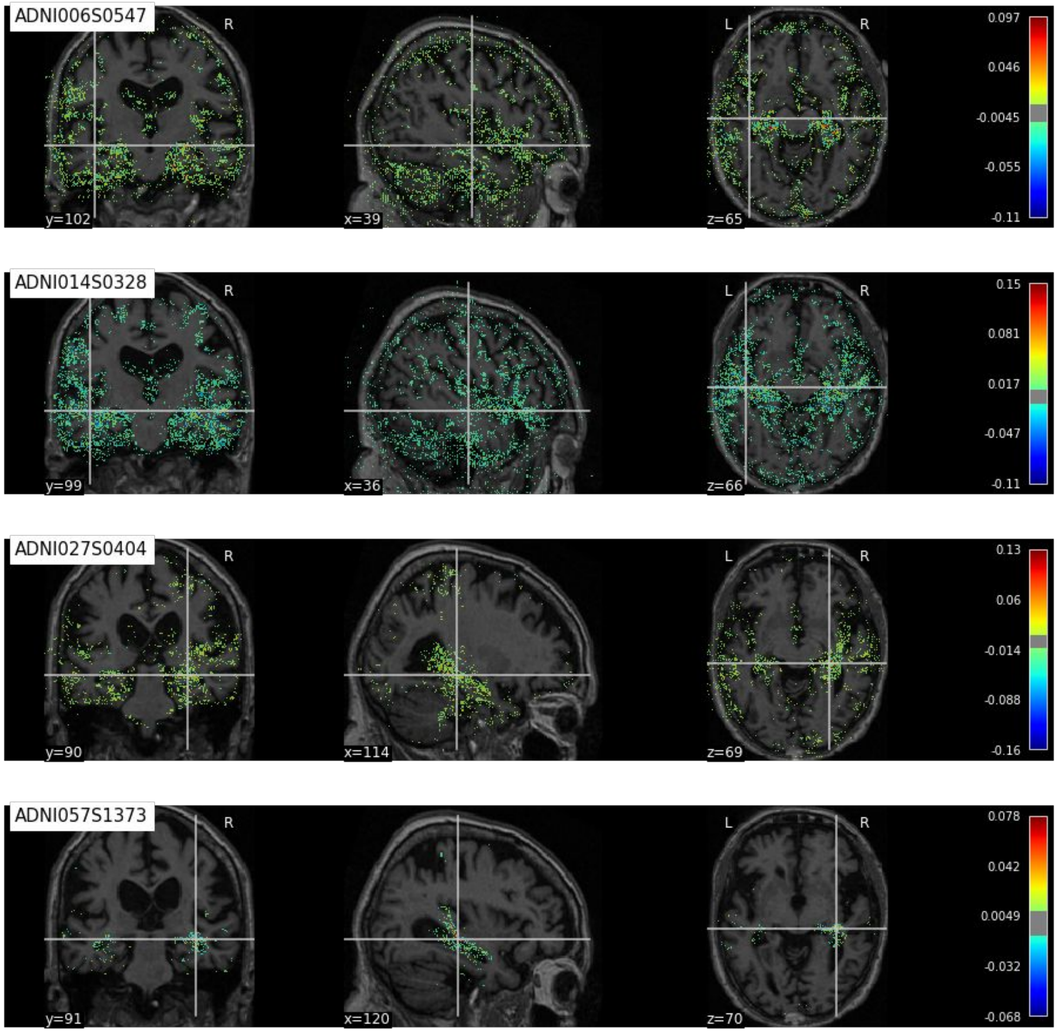
Model without skull



Guided Backprop

ADNI - AD/CN

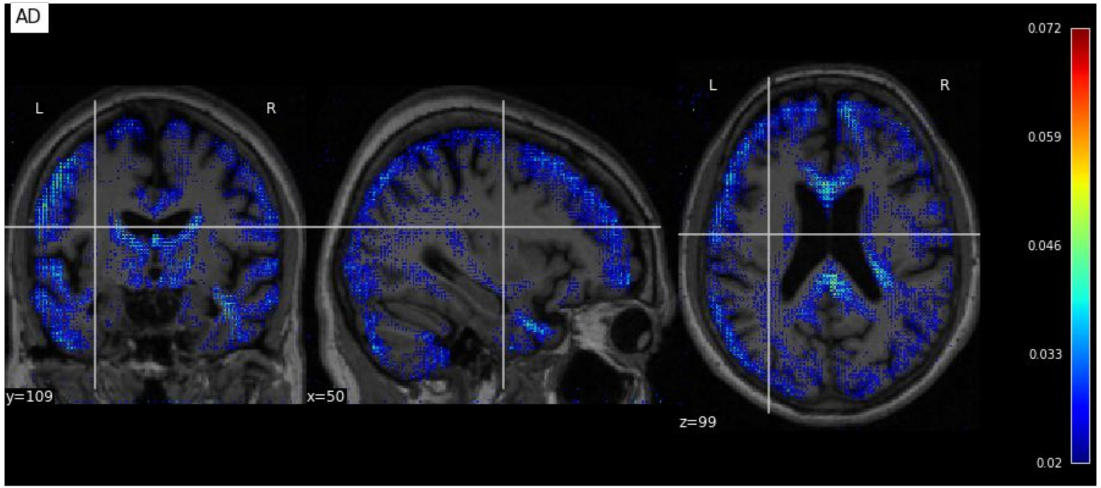
Model without skull

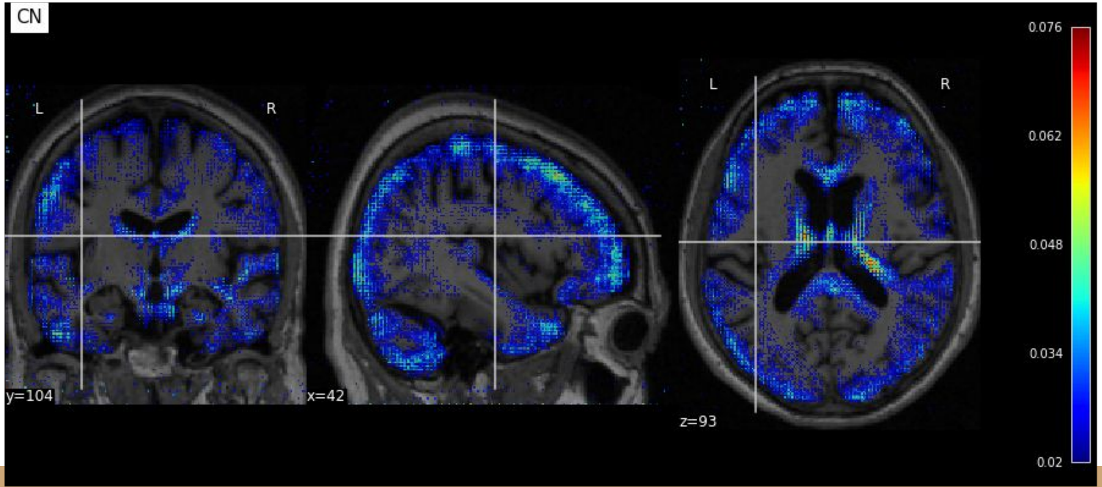


Guided Backprop

ADNI - AD/CN

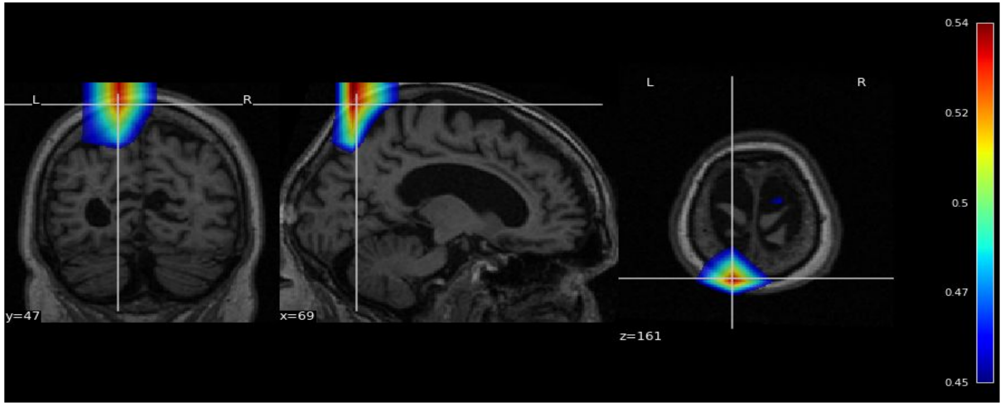
Model without skull

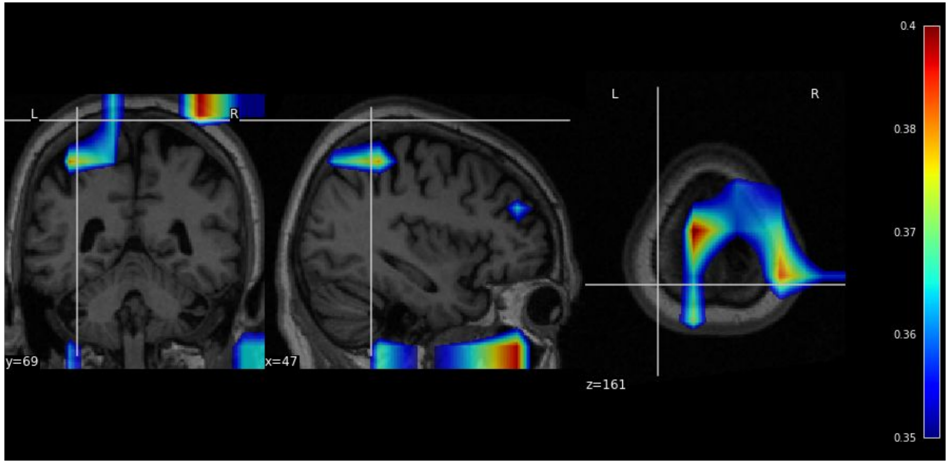




Grad-CAM

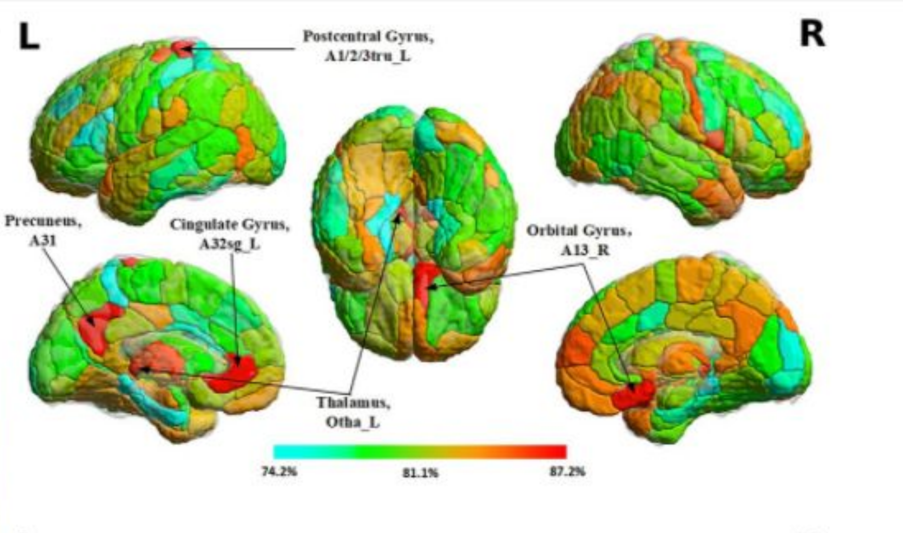
ADNI - AD/CN
Model without skull

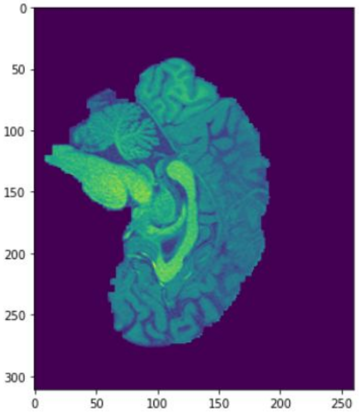


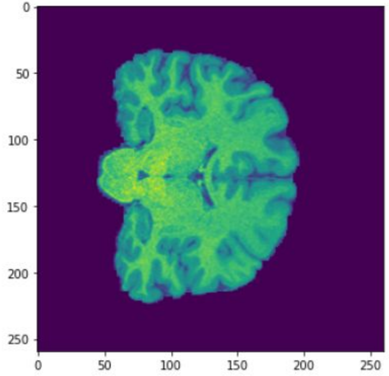


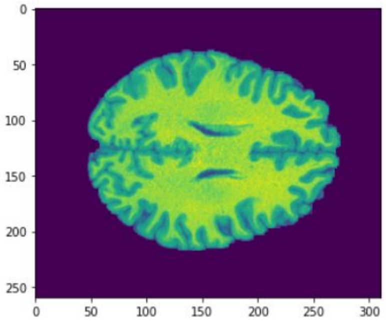
Task of gender patterns recognition

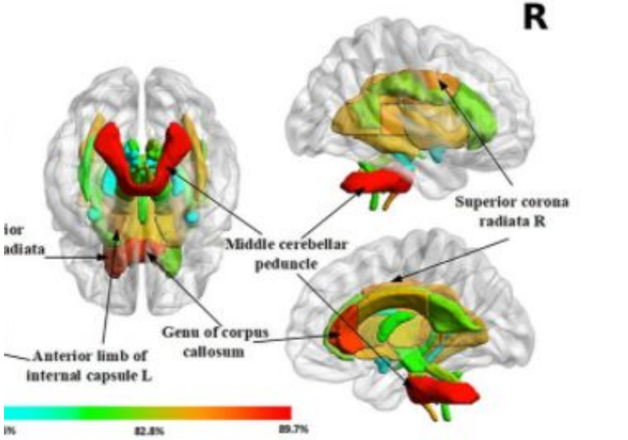
- Human Connection Project (HCP)
    - 1112 subjects
        - 575 female
        - 535 male



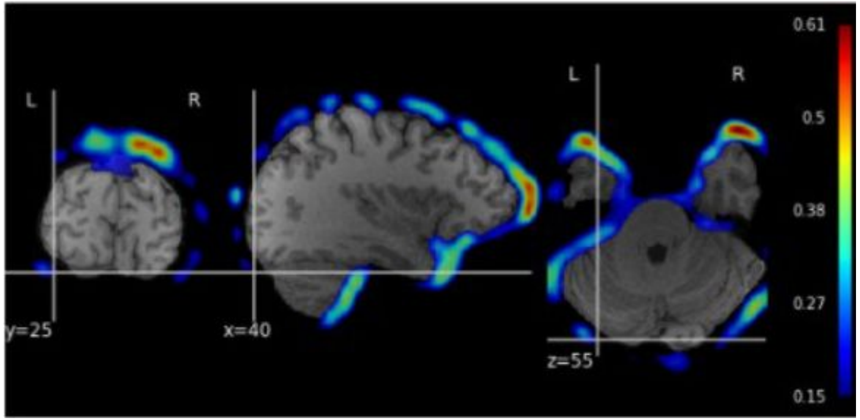


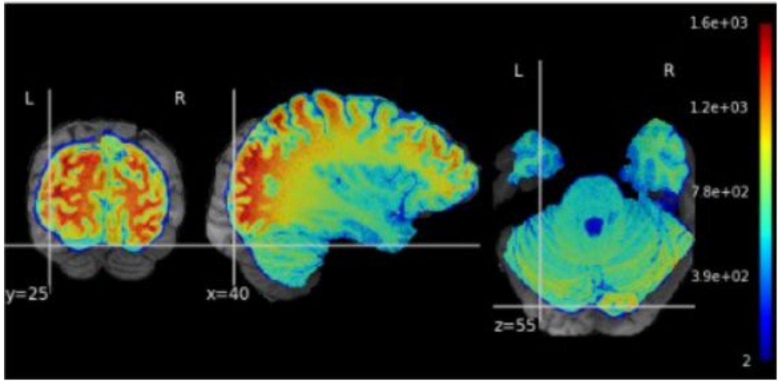






HCP: RESULTS





Smart augmentation: Optimal Scale based on Optimal Transport (OT)

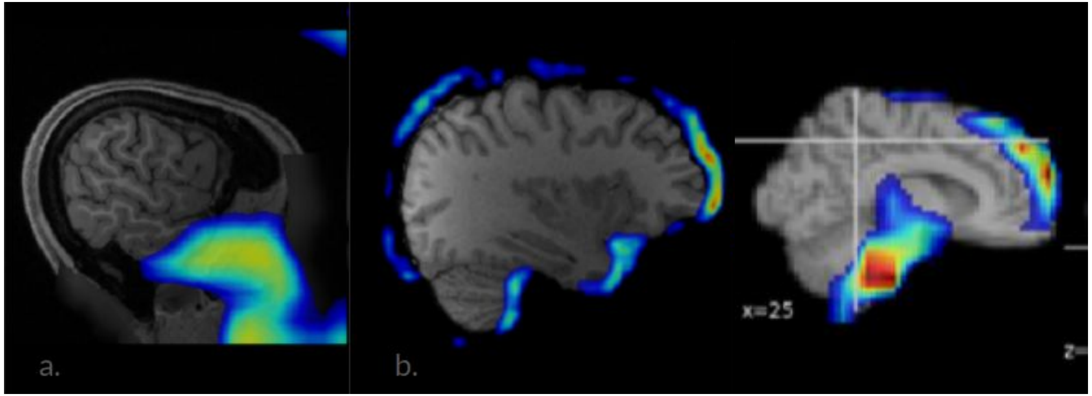
one continuous probability distribution (men) is transformed into another (women)

the correct scaling coefficient for each subject

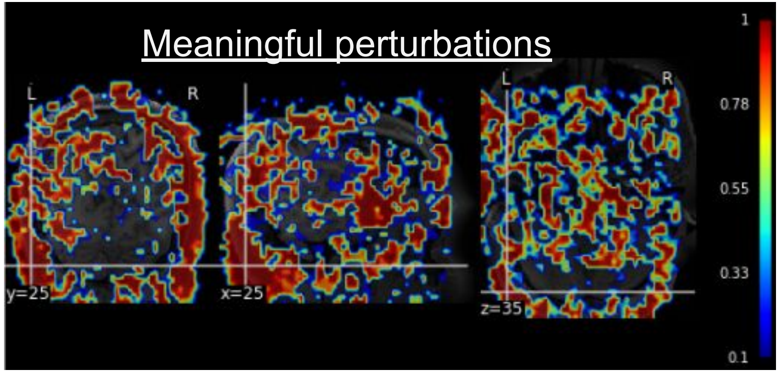
HCP: RESULTS

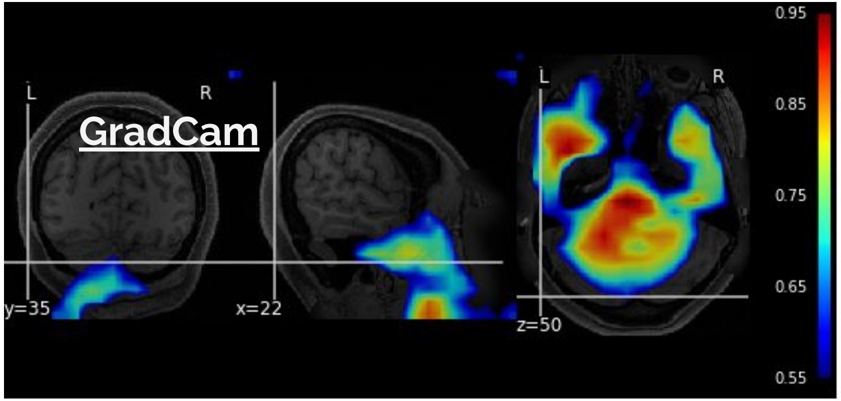
	Accuracy on CV3, Mean (STD)	
	Training	Validation
No MRI data preprocessing	**0.991 (0.001)**	0.976 (0.037)
Skull stripping (SS)	0.943 (0.012)	0.916 (0.094)
SS augmented with rotation	0.989 (0.013)	0.933 (0.018)
SS augmented with rotation and scaling	0.984 (0.016)	0.964 (0.020)
SS augmented with optimal scaling	0.996 (0.009)	**0.984 (0.075)**

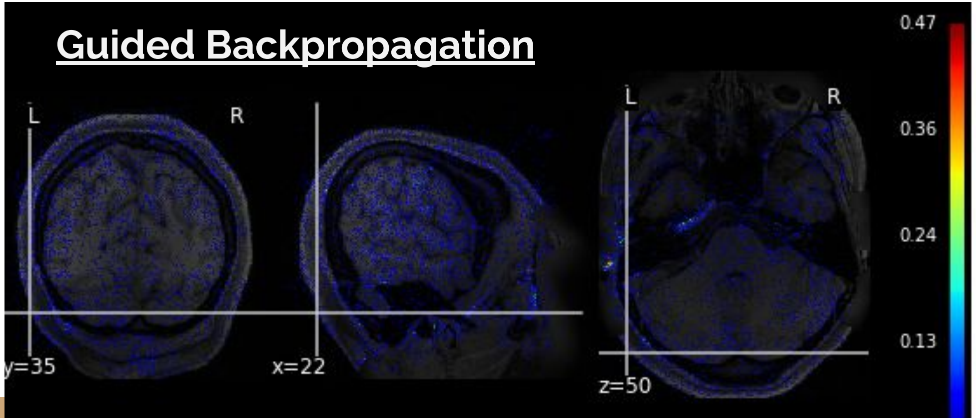
Example of GradCAM attention map (0 class):
(a) on data without preprocessing:the model pays attention to nasopharynx and Adam's apple area;
(b) on data with skull stripping:model pays attention to the difference in brain size;
(c) data trained with advanced augmentation:optimal scaling force the model to base its decision only on the brain structures



Training without pre-processing and without skull-stripping





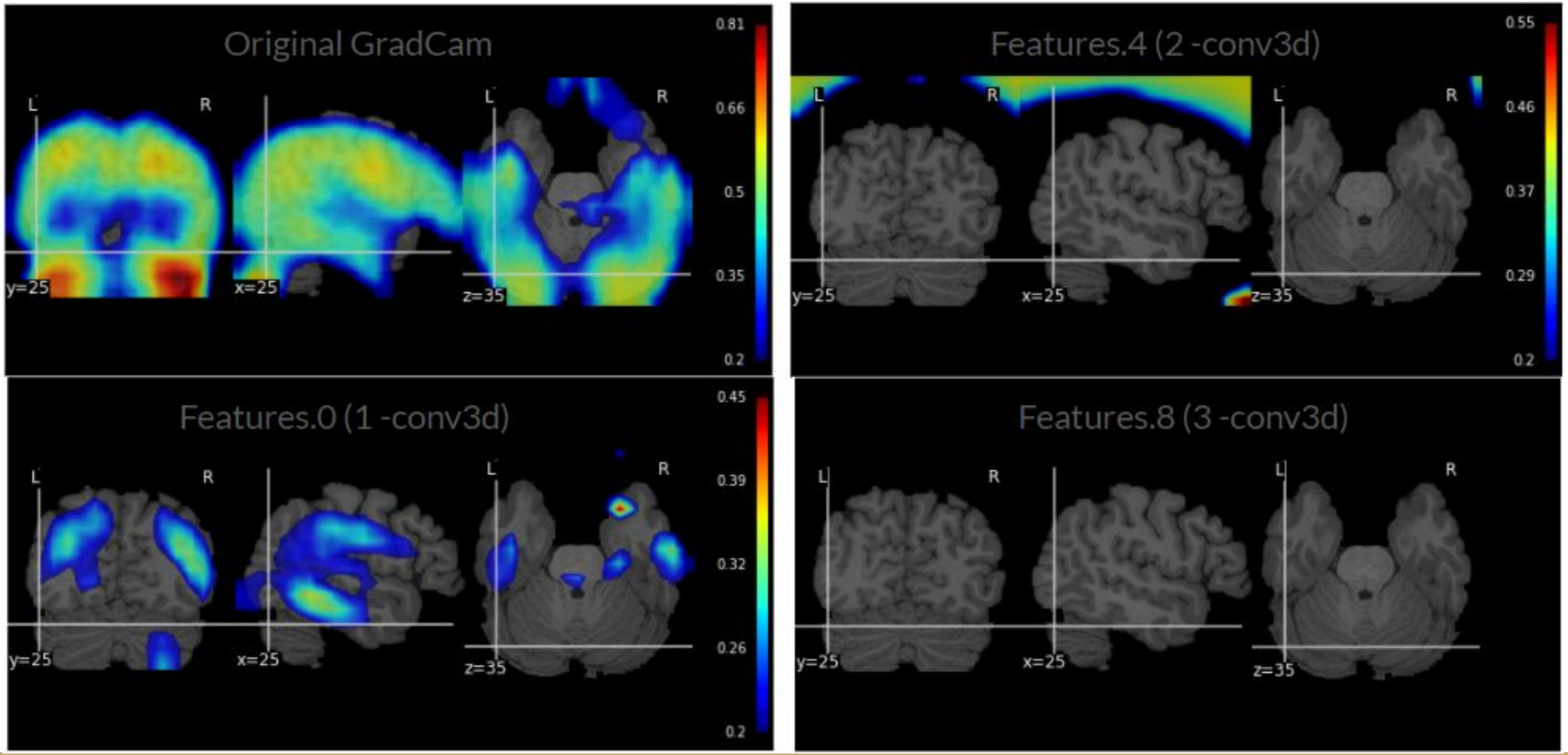


Sanity Check

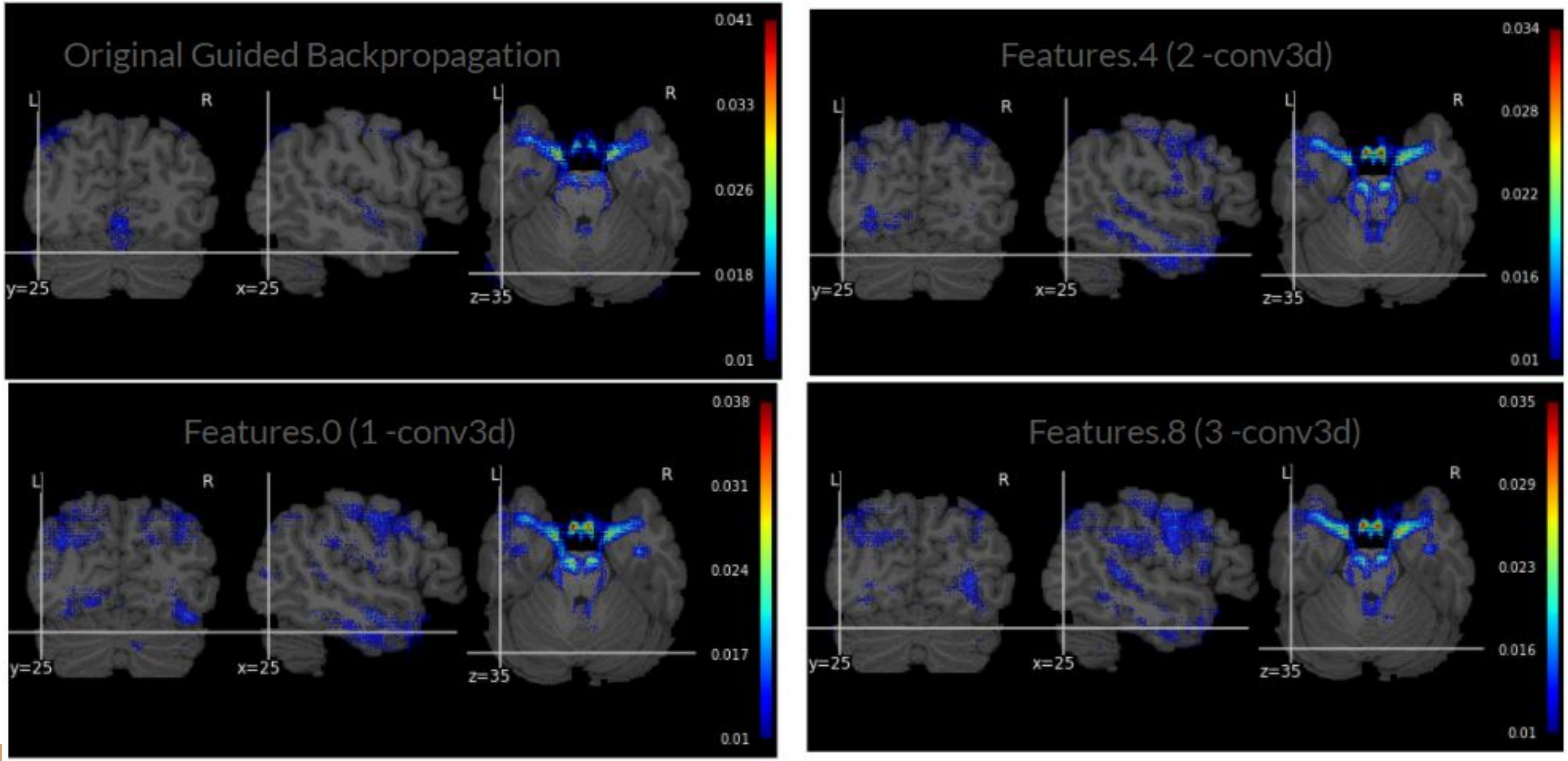
Model Randomization Test

randomize the weights of a model starting from the top layer, successively, all the way to the bottom layer. This procedure destroys the learned weights from the top layers to the bottom ones. We compare the resulting explanation from a network with random weights to the one obtained with the model's original weights

Model Randomization Test GradCam



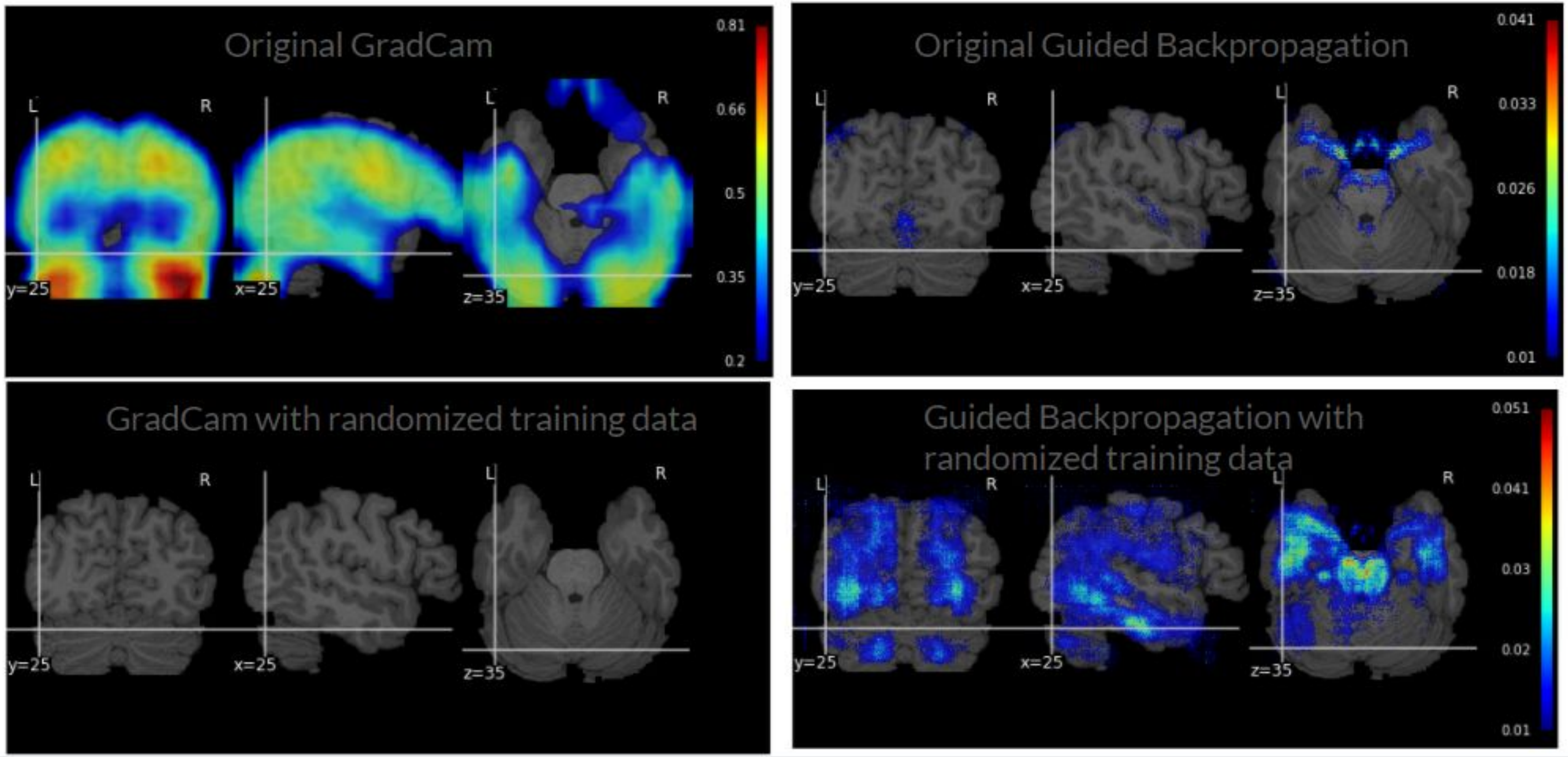
Model Randomization Test
Guided Backpropagatio



Data Randomization Test

permute the training labels and train a model on the randomized training data.
A model achieving high training accuracy on the randomized training data is forced to memorize the randomized labels without being able to exploit the original structure in the data. We now compare saliency masks for a model trained on random labels and one trained true labels.

Data Randomization Test



Guided Backprop

on the forward pass, we “remember” the positions of the maximum activation values for each layer, that is, the values of the local maximums in each pooling area (for example, a 2x2 square) are stored in so-called switches. Then, on the backward pass, we return the maximum activations to the same places using these switches.