

Data and demo codes for Shen, Horikawa, Majima, and Kamitani (2019) Deep image reconstruction from human brain activity. PLoS Comput. Biol.

<http://dx.doi.org/10.1371/journal.pcbi.1006633>.

☆ 180 stars  39 forks

★ Unstar

 Watch ▼

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

 master ▼

...



ShuntaroAoki ...

on 3 Jun



[View code](#)

README.md

Deep Image Reconstruction

Data and demo code for [Shen, Horikawa, Majima, and Kamitani \(2019\) Deep image reconstruction from human brain activity](#). PLOS Computational Biology. The preprint is available at bioRxiv ([Shen et al., 2017, Deep image reconstruction from human brain activity](#)).

Requirements

- Python 2.7
- [icnn](#)
- Numpy
- Scipy
- Pillow (PIL)
- Caffe with up-convolutional layer
 - <https://github.com/dosovits/caffe-fr-chairs> (Branch: deepsim)
 - Both CPU and GPU installation are OK

Usage

Preparation

1. Download data files from figshare (see [data/README.md](#)).
2. Download Caffe networks (see [net/README.md](#)).

DNN feature decoding from brain activity

You can skip the feature decoding from brain activity since we provide the decoded DNN features used in the original paper (see [data/README.md](#)).

We used the same methodology in our previous study for the DNN feature decoding (Horikawa & Kamitani, 2017, [Generic decoding of seen and imagined objects using hierarchical visual features, Nat Commun.](#)). Demo programs for Matlab and Python are available at <https://github.com/KamitaniLab/GenericObjectDecoding>.

Image reconstruction from decoded CNN features

We provide seven scripts that reproduce main figures in the original paper.

- 1_reconstruct_natural_image.py
 - Reconstructing **natural images** from the CNN features decoded from the brain with deep generator network (DGN); reproducing results in Figure 2.
- 2_reconstruct_natural_image_without_DGN.py
 - Reconstructing natural images from CNN features decoded from the brain **without** deep generator network (DGN); reproducing results in Figure 3A.
- 3_reconstruct_natural_image_different_combinations_of_CNN_layers.py
 - Reconstructing natural images from CNN features decoded from the brain with **different combinations of CNN** layers; reproducing results in Figure 4.
- 4_reconstruct_shape_image.py
 - Reconstructing colored artificial shapes from CNN features decoded from the brain; reproducing results in Figure 6A.
- 5_reconstruct_shape_image_different_ROI.py
 - Reconstructing colored artificial shapes from CNN features decoded from multiple visual areas in the brain; reproducing results in Figure 7A.
- 6_reconstruct_alphabet_image.py
 - Reconstructing alphabetical letters shapes from CNN features decoded from the brain; reproducing results in Figure 6B.
- 7_reconstruct_imagined_image.py
 - Reconstructing imagined image from CNN features decoded from the brain; reproducing results in Figure 8.

Data

- fMRI data: [Deep Image Reconstruction@OpenNeuro](#)
- Decoded CNN features: [Deep Image Reconstruction@figshare](#)

Notes

Enable back-propagation in the DNNs

In the demo code, we use pre-trained [VGG19](#) and pre-trained [deep generator network \(DGN\)](#) ([Dosovitskiy & Brox, 2016, Generating Images with Perceptual Similarity Metrics based on Deep Networks. arXiv.](#)). To enable make back-propagation, the following line should be added to the prototxt files (the file describes the configuration of the DNN):

```
force_backward: true
```

Get DNN features before ReLU

In our study, we defined DNN features of conv layers or fc layers as the output immediately after the convolutional or fully-connected computation (i.e., before applying the Rectified-Linear-Unit (ReLU)). However, as default setting of the pre-trained DNNs, ReLU operation is an in-place computation, which will override the DNN features we need. To In order to use the DNN features before the ReLU operation, you need to modify the prototxt file as below (taking the VGG19 prototxt file as an example).

Original:


```
layers {  
  bottom: "conv1_1"  
  top: "conv1_1"  
  name: "relu1_1"  
  type: RELU  
}
```

Modified:

```
layers {  
  bottom: "conv1_1"  
  top: "relu1_1"  
  name: "relu1_1"  
  type: RELU  
}
```

No releases published

Contributors 3

-  **ShuntaroAoki** ShuntaroAoki
-  **horikawa-t** horikawa-t
-  **shen-gh** shen-gh
-

Languages

