ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΕΛΟΠΟΝΝΗΣΟΥ
UNIVERSITY of the PELOPONNESE

ΔΗΜΟΚΡΙΤΟΣ
DEMOKRITOS

MSc in Data Science, Spring semester 2023

Deep Learning - Report

Eleni Neti   R.N.: 2022202204018

Petros-Fotis Kamberi   R.N.: 2022202204012

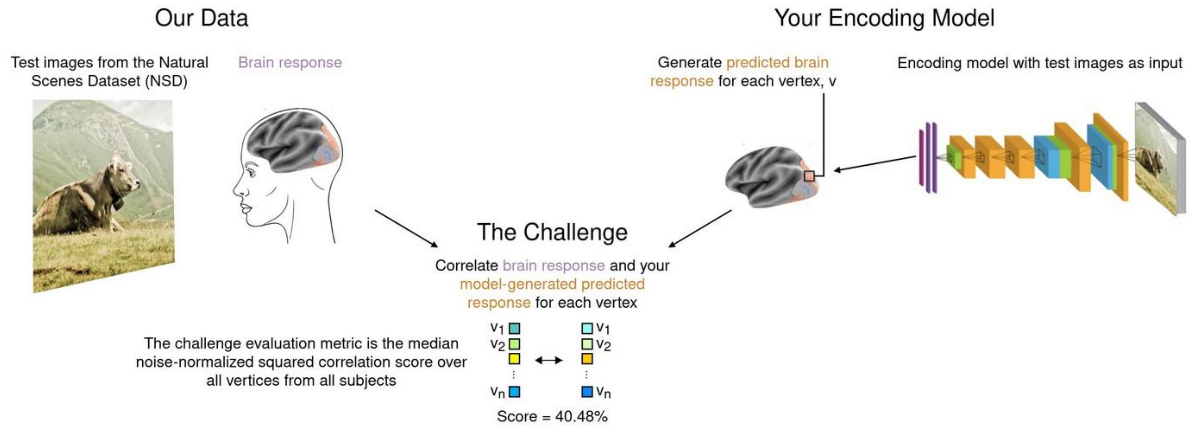
Project description

Our project is based on the running challenge “Algonauts project 2023”, released by the Algonauts project, which is an initiative to establish communication and interdisciplinary interaction between natural and artificial intelligence researchers.

This year’s installment focuses on understanding how the human brain processes natural scenes by utilizing the **Natural Scenes Dataset (NSD)**, which provides a rich collection of fMRI responses to approximately 73,000 naturalistic colored scenes. Regarding the fMRI responses, those were generated during scanning sessions from a total of eight individuals (subjects) who were exposed to visual stimuli.

The primary objective of our project is to construct an encoding model that can transform image pixels into meaningful model features. This model will subsequently map the extracted features to brain data, and particularly fMRI activity. By capturing the intricate relationship between image representations and neural responses, and by training deep learning architectures one can attempt to accurately predict the brain's reaction to various visual stimuli.

Overview of the challenge


Our Data
Test images from the Natural Scenes Dataset (NSD)
Brain response
Your Encoding Model
Generate predicted brain response for each vertex, v
Encoding model with test images as input
The Challenge
Correlate brain response and your model-generated predicted response for each vertex
The challenge evaluation metric is the median noise-normalized squared correlation score over all vertices from all subjects
v1
v2
vn
v1
v2
vn
Score = 40.48%

To mitigate the challenges posed by the problem complexity and limited computational resources, we developed deep neural networks specifically for the first (subject 1) out of eight subjects. Additionally, we constructed separate deep networks for each hemisphere of the brain. This approach allowed us to streamline the computational demands and focus on a more manageable subset of data, enabling efficient training and learning.

Furthermore, our implementation was performed using Google collab, and the notebooks containing our code for the computational algorithms we developed can be found in the GitHub repository provided below.

For the data acquisition process, we have included a guide in the form of a Jupyter notebook [**Guide to access the data.ipynb**] in the project’s github repo, to provide you with step-by-step instructions for acquiring the necessary data.

Before proceeding, it is crucial to carefully follow this guide, which will grant you access to the dataset in an unzipped form. The instructions provided will allow you to conveniently add a shortcut of the dataset to your Google Drive without the need to download or upload anything.

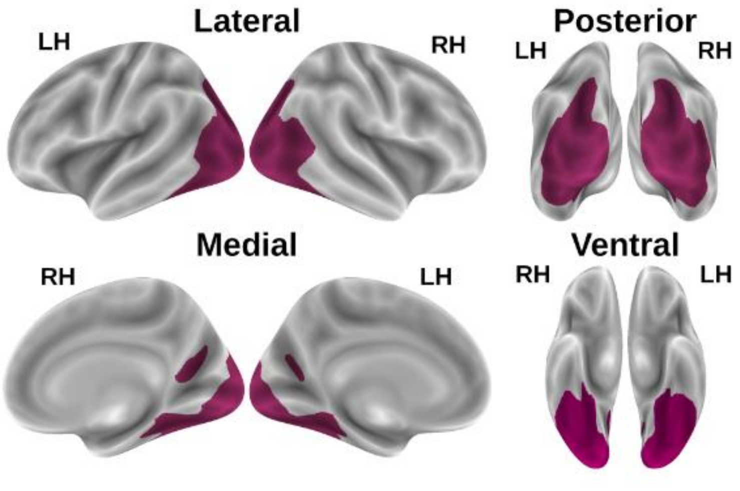
Some of the corresponding fMRI visual response images for testing (ground truth) are of course withheld for the purposes of the challenge, but a validation set (different from the training images) is provided instead.

The following link provides access to all the project material related to our implementation: https://github.com/PFKamberi/Deep_Learning_Project_Algonauts_2023

A small introduction to basic concepts of visual neuroscience

To facilitate a comprehensive understanding of the challenge and the data related to the brain's response to images, it is important to provide an overview of key concepts in visual neuroscience that are central to our project. Gaining familiarity with these concepts is crucial for keeping track of the workflow and the glossary related to the project.

- **fMRI** images: Functional magnetic resonance imaging (fMRI) images capture the functional activity of the brain by measuring changes in blood flow. These images provide insights into the localized neural activity associated with specific mental processes or tasks. They correspond to the regions of the brain that exhibit increased or decreased blood flow, indicating areas that are more active or less active during a particular cognitive or sensory activity.
- **Visual cortex**: Is a region of the brain that plays a central role in processing visual information. It is also organized into distinct areas that specialize in different aspects of visual processing.



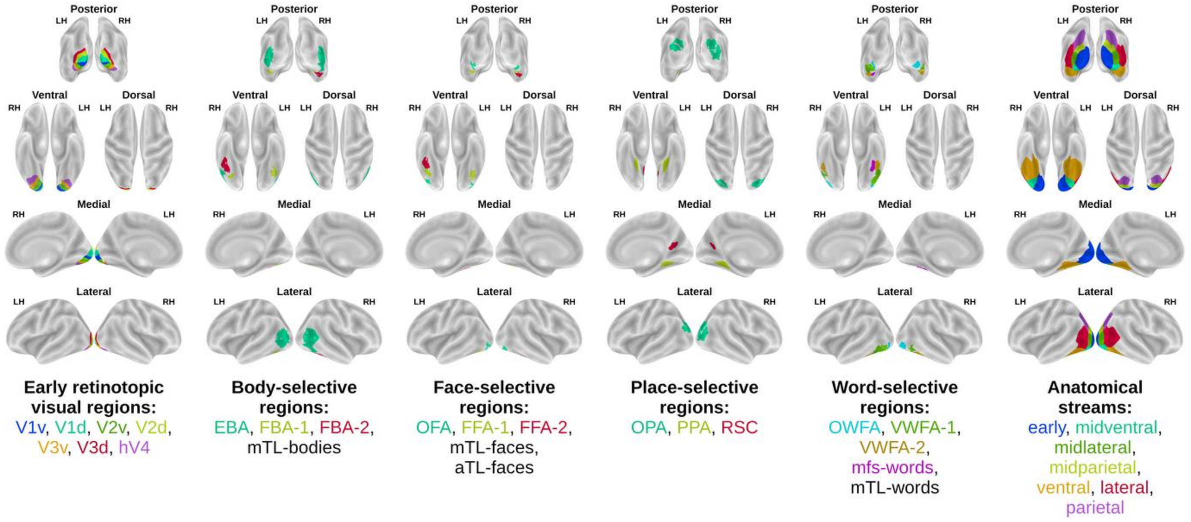
- **Vertices**: Vertices refer to specific locations or points on the surface of the visual cortex.

In neuroimaging studies, researchers often analyze and manipulate data at the level of vertices.

The challenge data comprises a subset of cortical surface vertices that are located within the visual cortex, and are also provided for both brain hemispheres.

Above, a depiction of the cortical surface vertices (lateral, posterior, medial and ventral) used in the challenge (purple) is provided.

- **ROI indices**: In addition to the fMRI data, ROI indices are also provided in order to allow the selection of vertices belonging to specific visual ROIs. The ROIs possess distinct functional properties, and thus they enable us to leverage the functional specialization of ROIs and build models that capture the intricacies of visual processing in a more targeted and accurate manner.


Posterior
LH
RH
Ventral
Dorsal
Medial
Lateral
Early retinotopic visual regions: V1v, V1d, V2v, V2d, V3v, V3d, hV4
Body-selective regions: EBA, FBA-1, FBA-2, mTL-bodies
Face-selective regions: OFA, FFA-1, FFA-2, mTL-faces, aTL-faces
Place-selective regions: OPA, PPA, RSC
Word-selective regions: OWFA, VWFA-1, VWFA-2, mfs-words, mTL-words
Anatomical streams: early, midventral, midlateral, midparietal, ventral, lateral, parietal

ROIs surface plots. Visualizations of subject 1 ROIs on surface plots. Different ROIs are represented using different colors. The names of missing ROIs are left in black.

- **Brain voxel**: It stands for volumetric pixel, which is a three-dimensional unit of measurement, analogous to a pixel in two-dimensional images. In brain imaging, each voxel corresponds to a specific point within the brain volume and contains information about the brain's structure or function.

Model building

Transfer learning

The idea behind transfer learning is that if someone has gone through the effort of training a big model on a bunch of data, one can probably use that already trained model as a starting point for their problem. In essence, we want to transfer to our problem all the information that a model has extracted from some different but related task, hence facilitating improved performance, and better utilization of the available resources.
Especially, since images have a lot of structural similarities, we could take a model trained on almost any large image classification task and use it to help us with our task.

For our purpose, we leveraged from Pytorch.hub the ResNet50 and VGG16 architectures.
A few words about them…

- **ResNet50**: Is a deep neural network with 50 layers, as its name suggests. As a member of the ResNet architecture family, it introduces residual connections that bypass a few layers and directly feed the input to deeper layers. This helps alleviate the vanishing gradient problem and facilitates the flow of gradients during training, enabling better optimization and the training of deeper networks.

- **VGG16**: Is a convolutional neural network architecture, consisting of 16 layers, including convolutional and fully connected layers. The architecture is known for its uniformity, where each convolutional block contains multiple 3x3 convolutional layers stacked together. It primarily uses 3x3 convolutional filters, enabling the network to capture local patterns and resulting in more discriminative feature representations.

Model Architectures

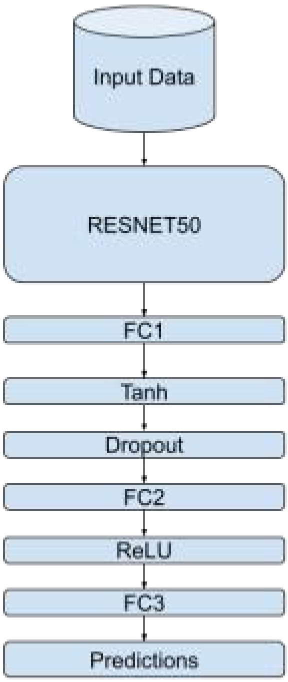
In the context of this project, two models were created using the pretrained ResNet50 and VGG16 architectures. The last layer of each pretrained model was removed, and additional layers were added to tailor the network for the specific regression task. The proposed model, named **LinearizingEncodingModel**, consists of three fully connected layers. The input dimension of the model corresponds to the number of features in the image representation, while the output dimension represents the number of brain **voxels**.

To enhance the expressiveness and flexibility of the model, several optional layers were included. The first fully connected layer, fc1, maps the input features to a hidden dimension, hidden_dim1. Activation functions and batch normalization can be optionally applied after fc1 to introduce non-linearity and improve convergence. Similarly, dropout regularization can be incorporated to mitigate overfitting. The output of fc1 is then passed through the second fully connected layer, fc2, which further transforms the hidden representation to hidden_dim2. Similar to fc1, activation functions, batch normalization, and dropout can be optionally employed after fc2. Finally, fc3 maps the hidden representation to the output dimension, representing the predicted voxel values.

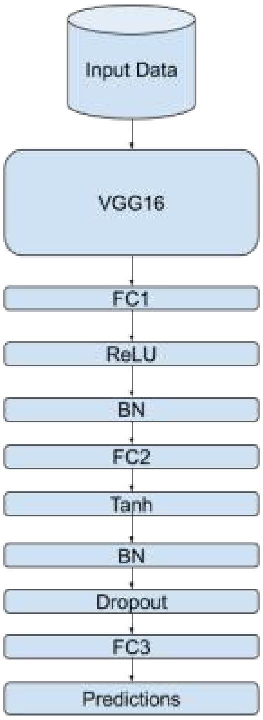
During the forward pass, the input is propagated through the defined layers in sequence, with optional activation functions, batch normalization, and dropout applied at appropriate stages. This process allows the model to learn complex mappings from image features to brain voxel values. The model architecture provides flexibility in terms of layer configurations, allowing for customization based on experimental requirements and optimization results.

The optuna autoML tool was utilized to determine a configuration of the transfer learning model architectures, including the subsequent inclusion of the LinearizingEncodingModel layers. The following diagrams offer a visual representation of these architectures, presenting a clear overview of the network structure and the arrangement of the added layers:

The structure of the transfer learning model that uses the pretrained RESNET50 model as determined by optuna.



The structure of the transfer learning model that uses the pretrained VGG16 model as determined by optuna.



Hyperparameter Tuning using optuna AutoML tool

As previously mentioned, the optuna autoML tool was employed to determine the optimal model architecture for both the ResNet50 and VGG16-based transfer learning models. However, in addition to selecting the layers, optuna was also utilized to determine various hyperparameters for each of the two models. The following hyperparameters were fine-tuned using optuna: hidden_dim1 and hidden_dim2, which define the dimensions of the hidden layers fc1 and fc2 respectively. The values were selected within a range that is logarithmically scaled based on the input dimension of the model. Moreover, optuna also determined the activation functions for these layers. The trial suggested either using ReLU or Tanh activation functions, providing flexibility in choosing the most suitable non-linearity.

Batch normalization, a technique used to improve network performance and stability during training, was another hyperparameter optimized by optuna. The trial suggested whether to include batch normalization for the first and second hidden layers (bnorm1 and bnorm2). This allowed the model to adaptively normalize the layer inputs, aiding in the optimization process.

Dropout regularization, which prevents overfitting by randomly dropping units during training, was also considered. Optuna determined whether to include dropout regularization for the first and second hidden layers (dropout1 and dropout2), along with the corresponding dropout ratios (dropout_ratio1 and dropout_ratio2). The dropout ratios were uniformly selected between 0.0 and 0.5, providing a range of regularization strengths.

Mitigating Overfitting

To mitigate overfitting in the training of the models, several measures were implemented, including the use of regularization techniques and early stopping. The following hyperparameters were specifically chosen to prevent overfitting:

- **Batch Normalization** (bnorm1, bnorm2): Batch normalization was incorporated as an option for the first and second hidden layers. By normalizing the layer inputs, batch normalization helps to stabilize and regularize the training process, reducing the risk of overfitting.

- **Dropout Regularization** (dropout1, dropout_ratio1, dropout2, dropout_ratio2): Dropout regularization was also included as an option for the first and second hidden layers. Dropout randomly drops a certain percentage of units during training, forcing the model to learn more robust and generalized representations. The dropout ratios (dropout_ratio1 and dropout_ratio2) were chosen uniformly between 0.0 and 0.5 to provide a range of regularization strengths.

Furthermore, **early stopping** was implemented during training to prevent overfitting.The early stopping mechanism follows a criterion based on the validation loss. Specifically, if there is no improvement in the validation loss for three consecutive epochs, training is halted to prevent further overfitting. In this case, the comparison between the current validation loss and the best validation loss is performed by considering the two significant decimal digits. By rounding the validation loss to two decimal places, it ensures that a meaningful improvement is required to reset the early stopping counter. This approach provides a more precise and stringent criterion for determining whether the model's performance has plateaued, allowing for timely termination of training to avoid overfitting.

By incorporating batch normalization, dropout regularization, and early stopping, the models were equipped with effective mechanisms to combat overfitting. These techniques help to generalize the learned representations and prevent the models from memorizing the training data, resulting in more robust and accurate predictions on unseen data.

Furthermore, optuna optimized other important hyperparameters such as learning_rate, optimizer (including choices between Adam and SGD), and weight_decay. These hyperparameters directly influence the training dynamics and regularization capabilities of the models, which are extremely useful in avoiding overfitting. The specific values of each hyperparameter, as determined by optuna, can be found in the code accompanying this report.

Batch Size and Number of Epochs

Due to resource limitations imposed by Colab Pro, the batch size and number of epochs were constrained for the ResNet50 and VGG16-based architectures. For the ResNet50-based architecture, the batch size was limited to 150, and the number of epochs was set to 50. Similarly, for the VGG16-based architecture, the number of epochs was limited to 50.

These limitations were determined based on the maximum amount of epochs that could be run initially without early stopping. By running the models for the maximum number of epochs, we observed signs of overfitting, where the model's performance on the training data continued to improve while the performance on the validation data started to plateau or deteriorate. To address this overfitting issue, we eventually incorporated early stopping, as described in the previous section, to prevent further training when no improvement in the validation loss was observed.

It is worth noting that larger batch sizes were not feasible due to memory limitations. Increasing the batch size requires more memory resources to process a larger number of samples in parallel, which exceeded the capabilities of the available hardware. Therefore, the batch size was set to a manageable value to ensure efficient training without exhausting the memory resources.

Results

Model Evaluation choices

The performance evaluation of the developed models involved several metrics and visualizations to assess its effectiveness in mapping images to brain voxels. To understand the model's learning dynamics, learning curves of the training and validation loss were plotted. These curves provide insights into how the model's performance evolves over epochs, indicating whether the model is underfitting or overfitting.

Residual plots were also examined to assess the quality of the model's predictions. By plotting the differences between the predicted voxel values and the ground truth values, residual plots enable the identification of any systematic patterns or biases in the model's predictions.

Pearson correlation coefficient was computed to measure the linear relationship between the predicted and actual voxel values. This metric quantifies the strength and direction of the linear association, providing an indication of how well the model captures the underlying relationship between the image features and brain voxels.

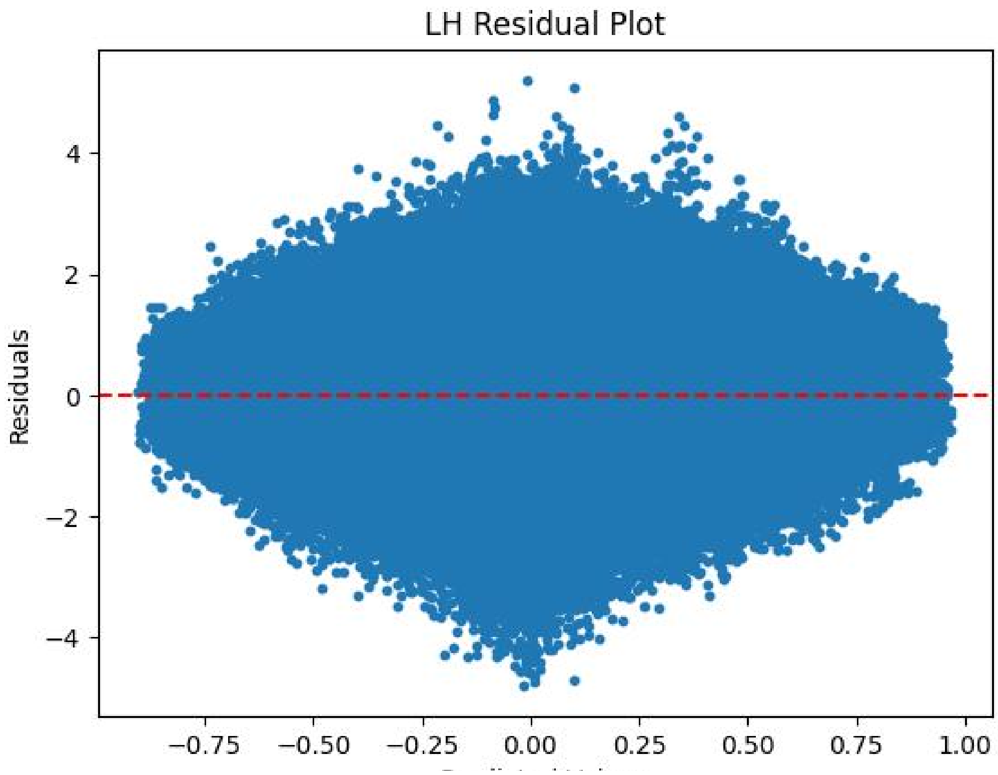
To visualize the model's predictions on brain surface maps, a heatmap was generated using the Nilearn library. The heatmap represents the voxel values as colors on a brain surface, allowing for a spatial representation of the predicted values. This visualization aids in identifying the regions where the model performs well and areas where there may be discrepancies.

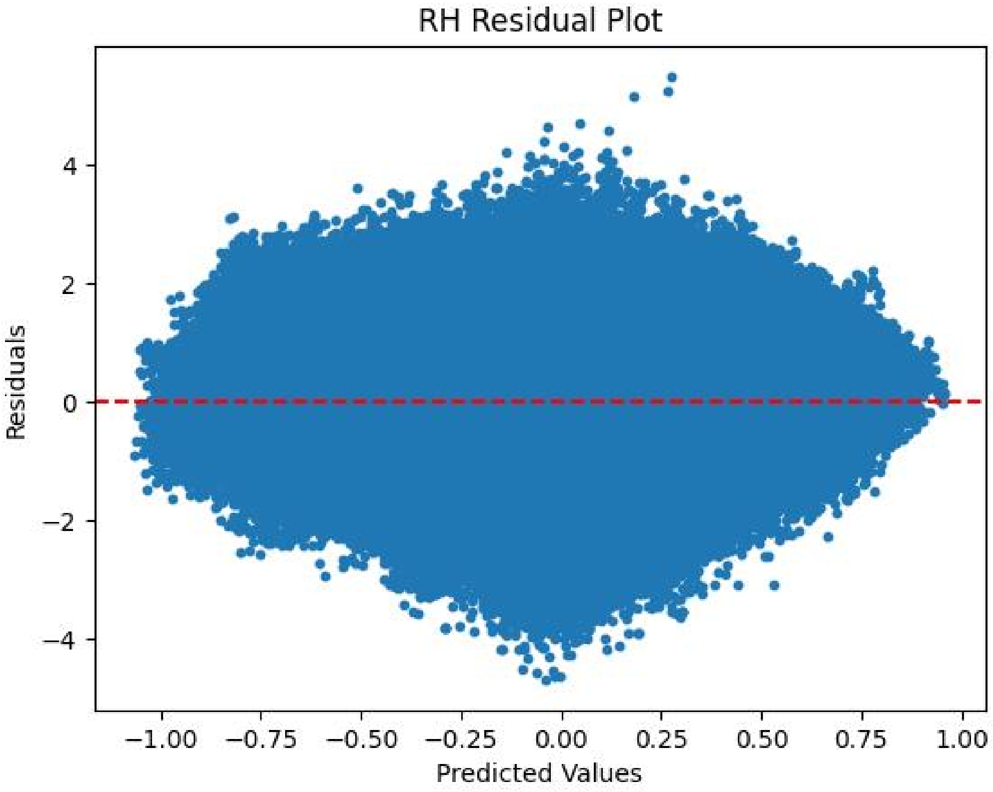
Furthermore, several evaluation metrics were calculated, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared (R2), and Smooth L1 Loss. These metrics provide quantitative assessments of the model's predictive accuracy, precision, and goodness of fit, allowing for a comprehensive evaluation of its performance.

In addition to overall performance, a detailed analysis was conducted to assess the model's performance on specific brain regions. Pearson correlation plots were generated for each brain region of interest (ROI) in both the left and right hemisphere. This analysis aimed to evaluate how the model performs on each specific region of the brain, enabling insights into the model's regional performance variations. By examining these plots, potential strengths and weaknesses in the model's predictions across different brain regions could be identified.

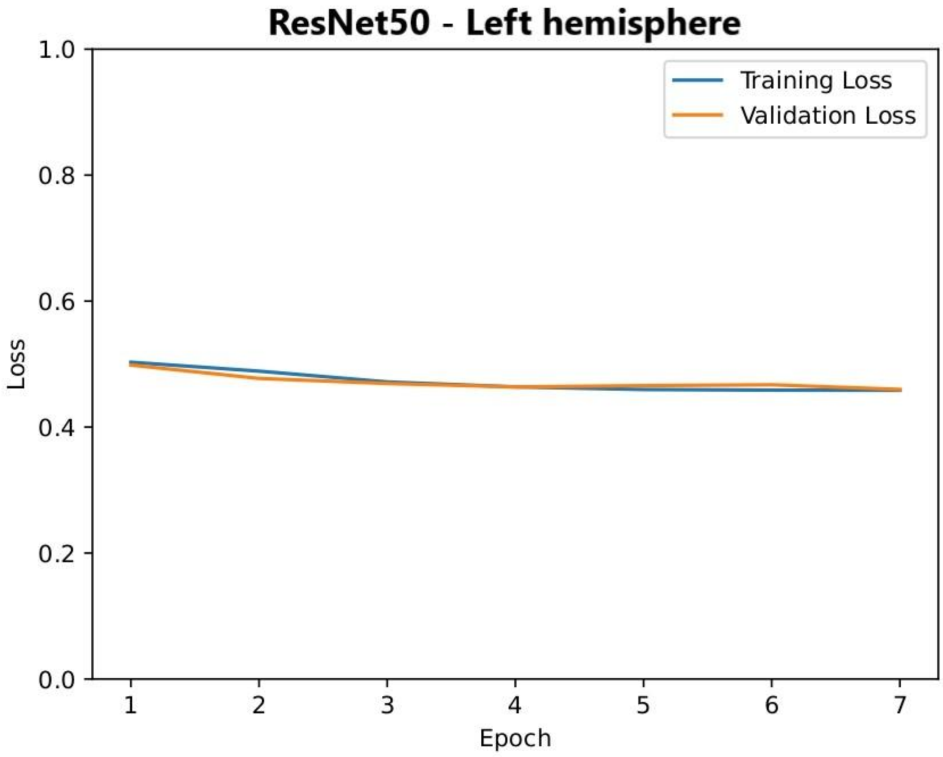
ResNet50

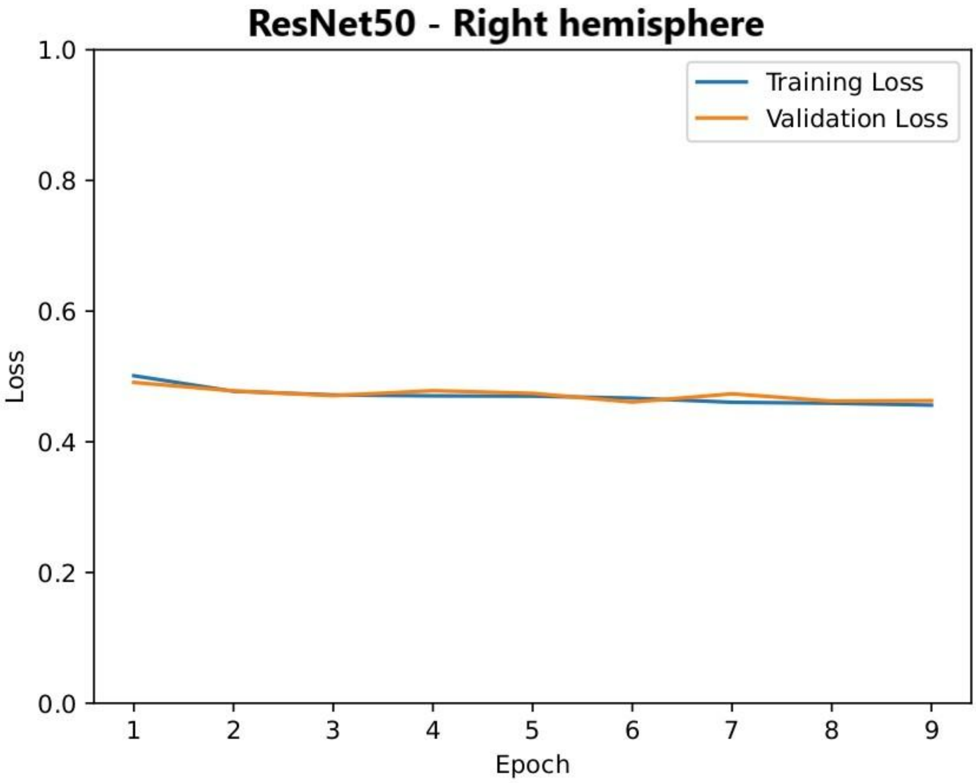
LH RMSE	0.6769
LH R2	0.0799
LH MAE	0.5354
LH Smooth L1 Loss	0.2162
RH RMSE	0.6789
RH R2	0.0690
RH MAE	0.5373
RH Smooth L1 Loss	0.2175



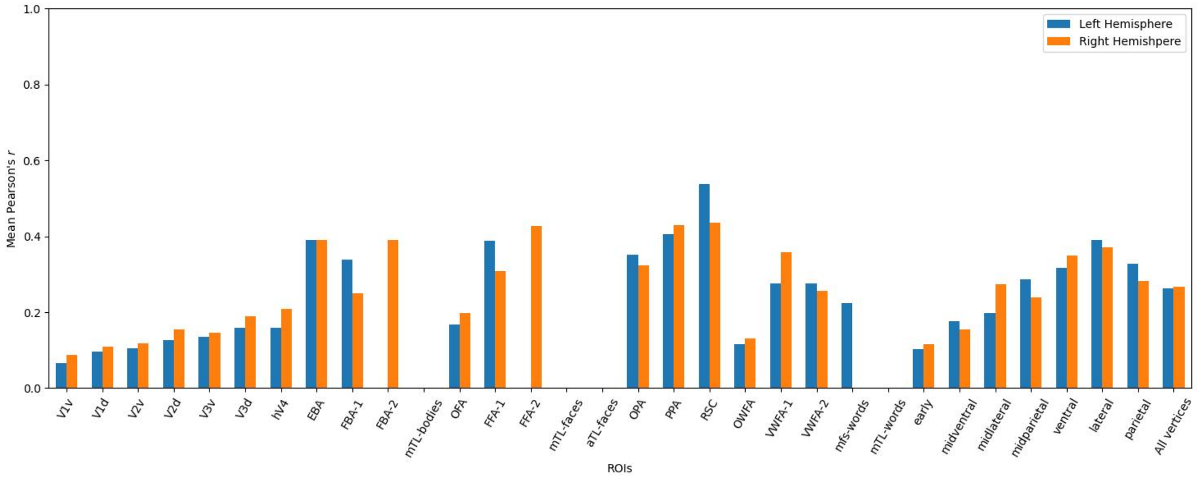


The residual plots for the implemented model using ResNet50 as a pretrained component, exhibits a noticeable degree of symmetry. This indicates that the model is generating balanced errors across the range of predicted values. The absence of a systematic bias towards underfitting or overfitting suggests that our model achieves descently accurate predictions. Furthermore, the observed symmetry in the residual plots suggests that the linearity assumption is reasonably met.



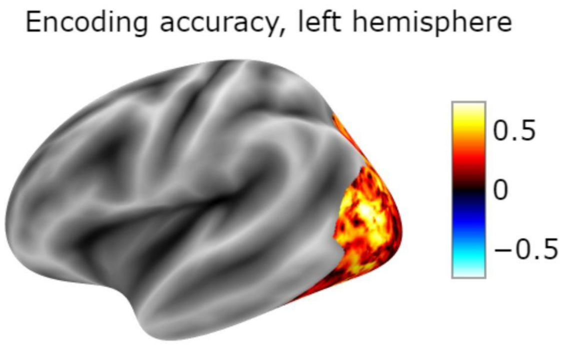


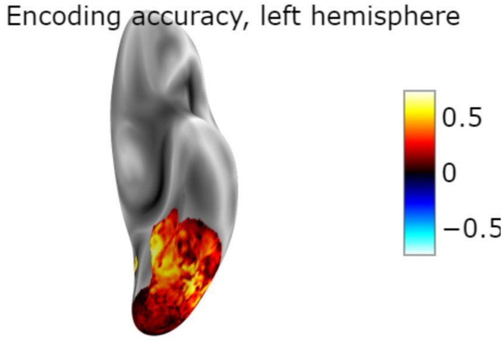
In the case of ResNet, the learning curves show that initially, the training and validation losses decrease together, indicating that the model is learning from the training data and generalizing well to unseen validation data. However, after a certain number of epochs, both losses reach a plateau, suggesting that the model has converged and further training does not significantly improve its performance. The fact that the validation loss closely follows the training loss indicates that the model is not overfitting or underfitting the data. The use of early stopping, which terminates the training process when the validation loss stops improving, suggests that the model has been trained for a sufficient number of epochs, preventing it from overfitting.

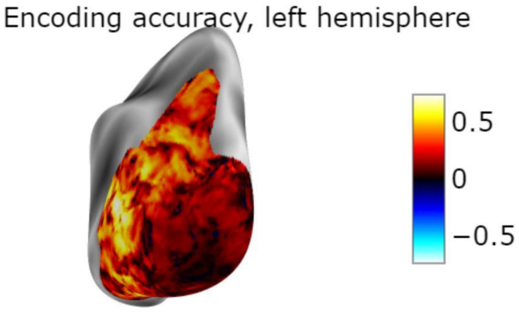


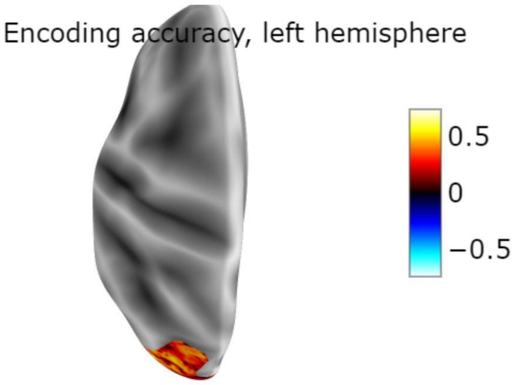
The encoding results for the individual ROI through the metric of Pearson's r correlation, are also provided in the above diagram. Each bar corresponds to the average correlation of all vertices of one ROI. This could be informative if one's interested in knowing whether the model is predicting certain ROIs better than others. As evident from the above diagram, our ResNet50-based model is better in predicting intermediate ROIs like PPA, RSC which correspond to the place-selective regions of the visual brain.

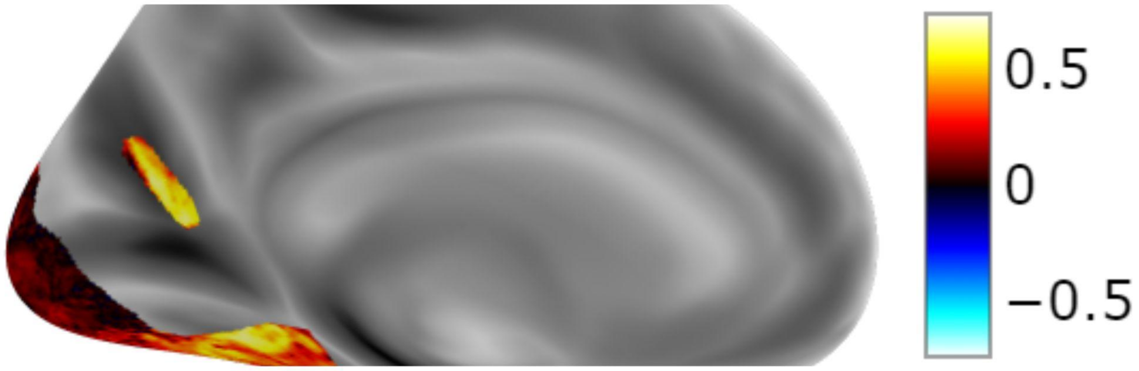
Heatmaps of the visual cortex





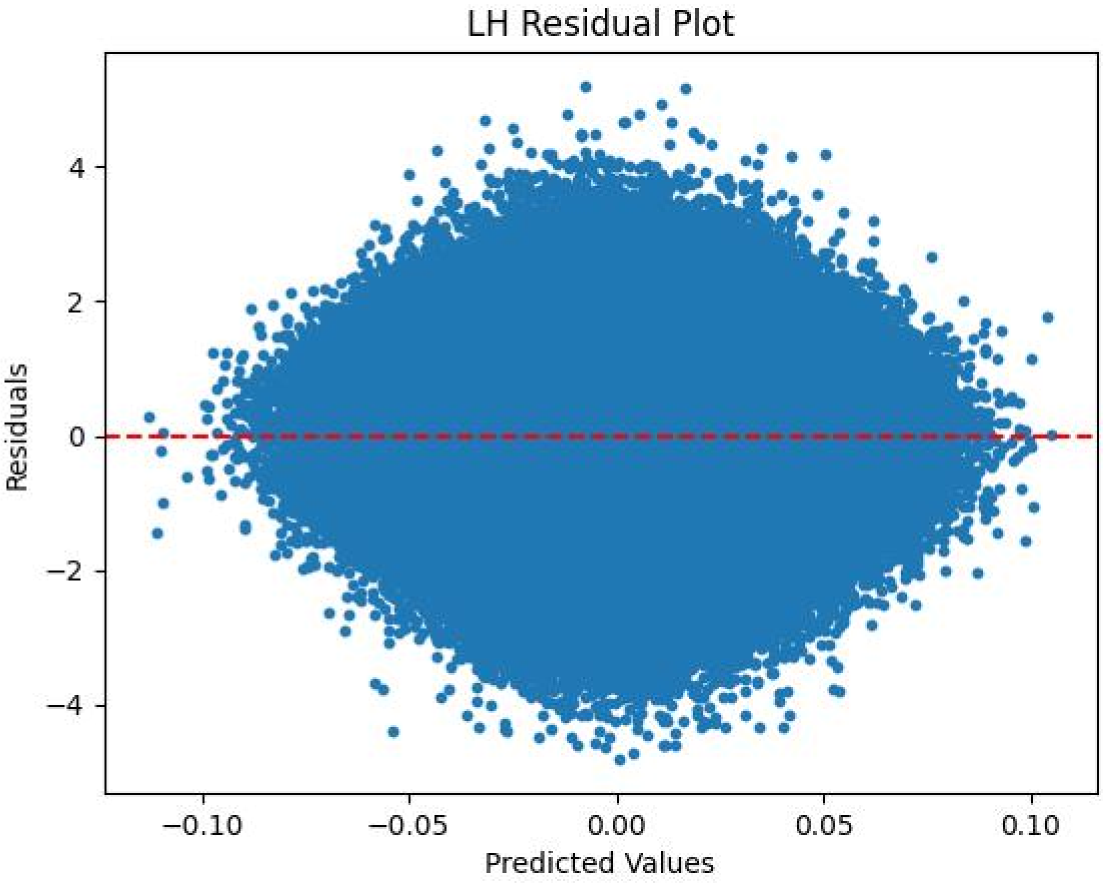


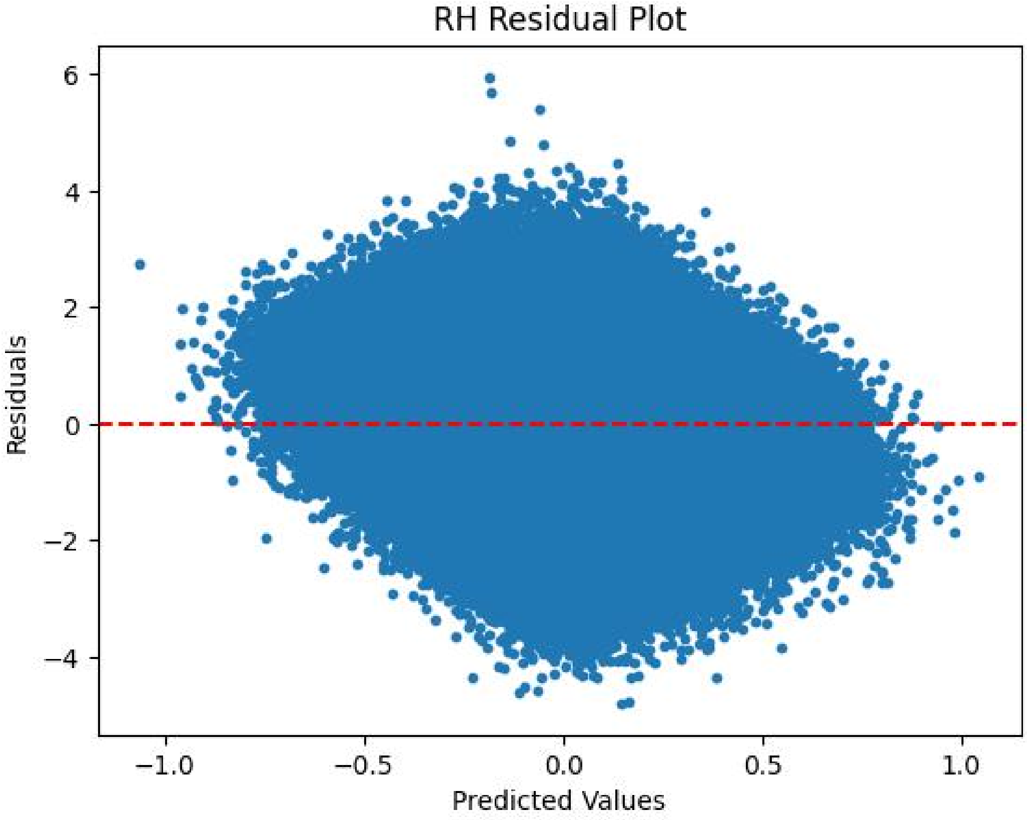


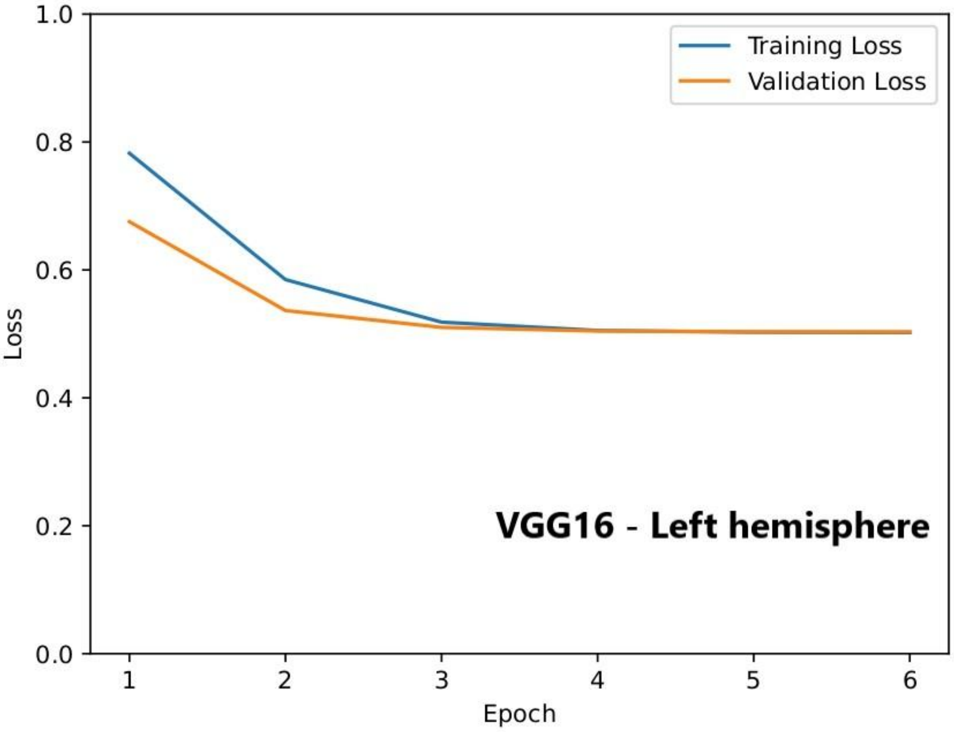


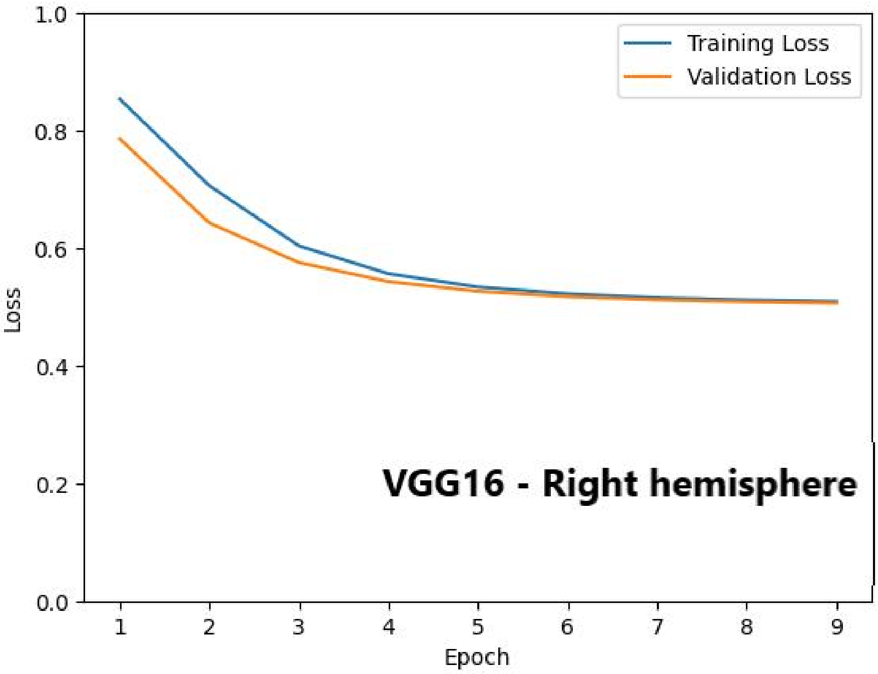
VGG16

LH RMSE	0.7086
LH R2	-0.0014
LH MAE	0.5619
LH Smooth L1 Loss	0.2352
RH RMSE	0.7201
RH R2	-0.0412
RH MAE	0.5719
RH Smooth L1 Loss	0.2424

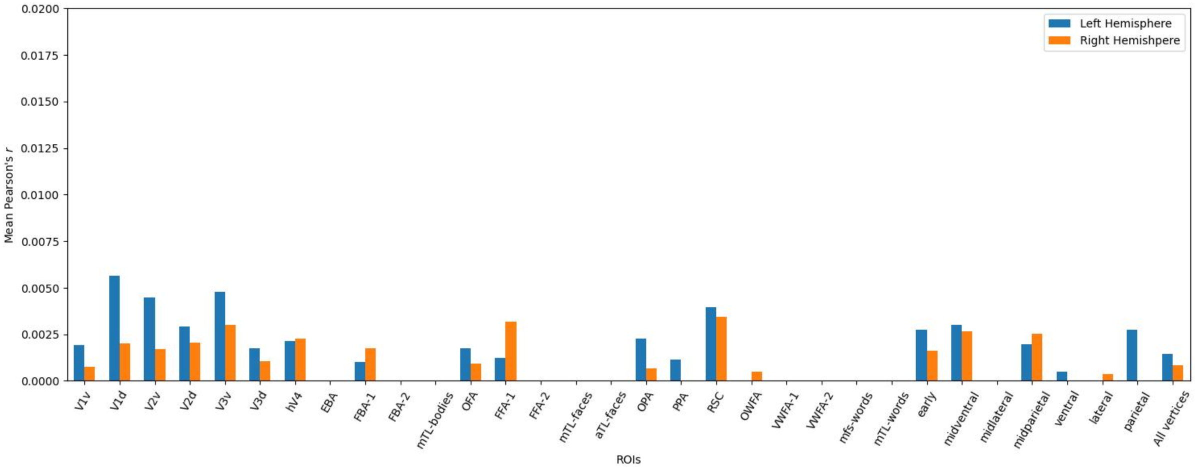



RH Residual Plot
Residuals
6
4
2
0
−2
−4
−1.0
−0.5
0.0
0.5
1.0
Predicted Values


1.0
0.8
0.6
0.4
0.2
0.0
Loss
1
2
3
4
5
6
Epoch
Training Loss
Validation Loss
VGG16 - Left hemisphere

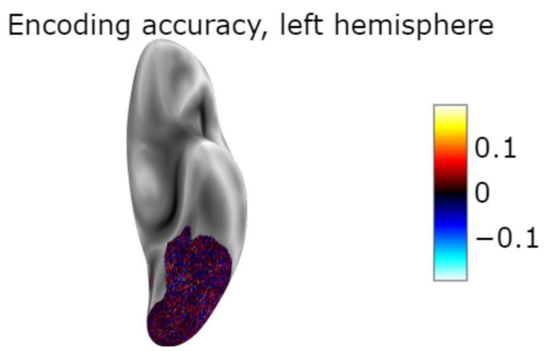


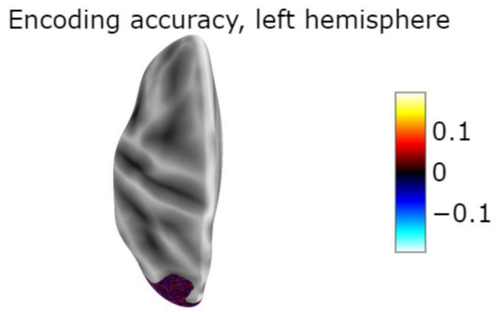
The learning curves for the VGG-based model exhibit a different pattern from the ResNet-based one. The training and validation losses start from different positions, indicating that the model initially performs better on the training data than on the validation data. This suggests that the model might be overfitting to some extent, as it struggles to generalize well to unseen data initially. However, as the training progresses, both losses eventually reach a plateau, indicating that the model's performance stabilizes and further training does not yield significant improvements. The use of early stopping in this case suggests that the model might have been prone to overfitting if training were continued for a longer duration.

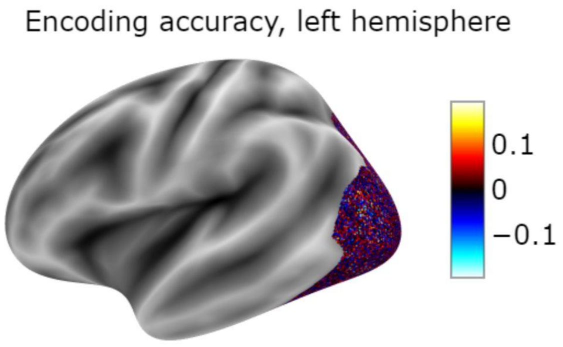


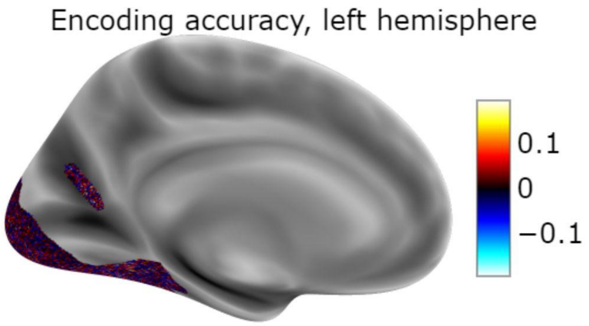
Poor performance of the VGG-based model is evident from the mean Pearson's correlation diagram, failing to predict any ROI in both left and right hemisphere. The linearizing model appears to be inappropriate for the given task.

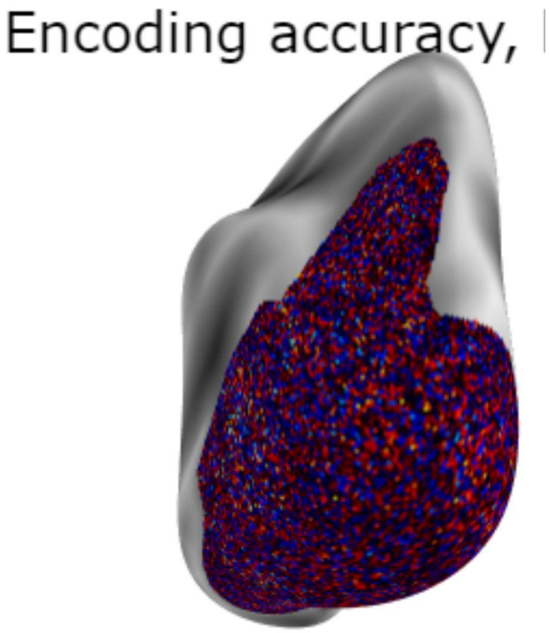
Heatmaps of the visual cortex











The poor performance is also evident from the above heatmaps, where dark regions showcase how bad the model performs in capturing the linear relationship between predicted and actual voxel values.