THE ALGONAUTS PROJECT 2023 CHALLENGE: UARK-UALBANY TEAM SOLUTION

TECHNICAL REPORT

Xuan Bac Nguyen
CVIU Lab, EECS Department
University of Arkansas, USA
xnguyen@uark.edu

Xudong Liu
cosinexd@gmail.com

Xin Li
University at Albany, SUNY
xli48@albany.edu

Khoa Luu
CVIU Lab, EECS Department
University of Arkansas, USA
khoaluu@uark.edu

ABSTRACT

This work presents our solutions to the Algonauts Project 2023 Challenge. The primary objective of the challenge revolves around employing computational models to anticipate brain responses captured during participants' observation of intricate natural visual scenes. The goal is to predict brain responses across the entire visual brain, as it is the region where the most reliable responses to images have been observed. We constructed an image-based brain encoder through a two-step training process to tackle this challenge. Initially, we created a pretrained encoder using data from all subjects. Next, we proceeded to fine-tune individual subjects. Each step employed different training strategies, such as different loss functions and objectives, to introduce diversity. Ultimately, our solution constitutes an ensemble of multiple unique encoders. The code is available at https://github.com/uark-cviu/Algonauts2023

Keywords Vision Brain Challenge · fMRI · Deep Learning

1 Introduction

Over the past decade, the deep learning revolution has significantly impacted scientific research efforts, with profound implications for both artificial and biological intelligence. Initially inspired by the visual system of the mammalian brain, deep learning algorithms have evolved to become cutting-edge AI agents and scientific models for understanding the brain itself. As a result, research on artificial and biological intelligence is becoming increasingly intertwined.

To improve this research direction, the 2023 edition of the Algonauts Project has been proposed [4]. It follows the same objective as its predecessors in predicting human visual brain responses through computational models. However, it stands out from the previous 2021 edition of challenges [2] using the Natural Scenes Dataset (NSD) [1], the most extensive and data-rich collection of neural responses to natural scenes. The primary focus of the challenge lies in visual scene understanding, as vision remains an unsolved problem in both artificial and biological intelligence. The collaboration between these two fields has been particularly impactful in this domain, making it a promising area for further exploration.

2 Dataset and Evaluation Protocol

.

2.1 Dataset

The competition exploits the NSD dataset, an extensive collection comprising responses from 8 subjects, recorded using high-quality 7T fMRI. At the same time, they were exposed to approximately 73,000 different natural scenes to build the encoding models for the visual brain. From subjects 1 to 8, each has 9841, 9841, 9082, 8779, 9841, 9082, 9841, and 8779 unique images, respectively. In addition, the corresponding fMRI visual responses of each image are also provided. These signals contain the left hemisphere (LH) and right hemisphere (RH) that consist of 19,004 and 20,544 vertices, respectively, except for subject 6, which has 18,978 LH vertices and 20,220 RH vertices, and subject 8, which has 18,981 LH vertices and 20,530 RH vertices. These variations in the number of vertices are due to missing data for the specified subjects.

2.2 Evaluation Metric

The evaluation metric is measured by taking the mean noise-normalized encoding accuracy across all the vertices of all subjects and hemispheres.

$$m = \frac{1}{v}\sum_{i}^{v}\frac{R_i^2}{NC_i} \tag{1}$$

where R_i is the Pearson correlation coefficient between predicted response P_i and ground truth G_i and NC_i is the noise ceiling.

3 Methods

In this section, we first present the details of the pre-training stage using the data from all subjects. The fine-tuning stage will be discussed in the next section.

3.1 All Subject Pretraining

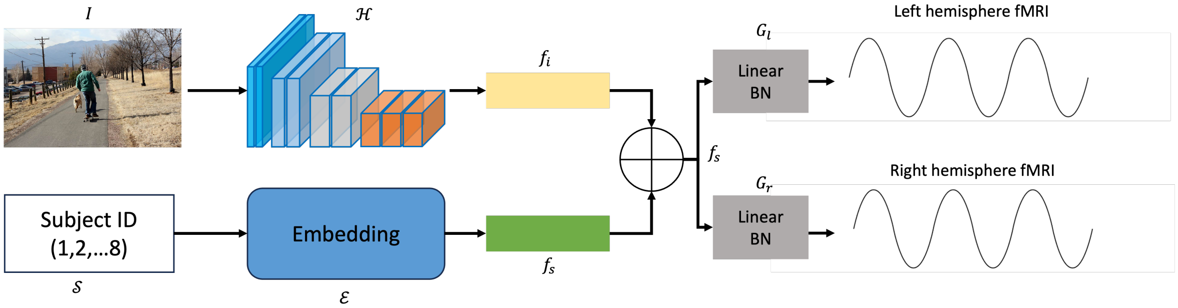

Figure 1: An overview of pretaining stage.

The overview of the pretraining stage is illustrated in Fig 1.

Let $I \in \mathbb{R}^{H\times W\times C}$ be the input image where H, W, and C are the height, width, and number of channels. We use a deep neural network (DNN) denoted $\mathcal{H}$ to extract the features of this image denoted f_i.

$$f_i = \mathcal{H}(I) \in \mathbb{R}^{d_i} \tag{2}$$

In our experiment, a unique set of 1000 images was shared among eight subjects. However, training these images during the pre-training stage might encounter noise-related problems, as the brain responses to the same input image can differ between subjects. In order to effectively tackle this issue, the proposed network needs to incorporate the subject's ID as a crucial factor in learning and specify the corresponding fMRI signals accurately. Let $\mathcal{S}$ be the subject ID whose value ranges from 0 to 7, indicating the subject from 1 to 8. We design an embedding module $\mathcal{E}$ to learn the features of the subject f_s.

$$f_s = \mathcal{E}(S) \in \mathbb{R}^{d_s} \tag{3}$$

Next, we concatenate the image and subject features to construct the features f.

$$f = concat(f_i, f_s) \in \mathbb{R}^{d_i+d_s} \tag{4}$$

These features will be passed into two dependent blocks of linear and batch norm layers denoted as G_l and G_r.

$$y_l = G_l(f) \in \mathbb{R}^{L_l} \tag{5}$$

$$y_r = G_r(f) \in \mathbb{R}^{L_r} \tag{6}$$

where y_l and y_r are the predicted fMRI signals of the left and right hemispheres, respectively.

3.2 Fine-tuning on Individual Subject

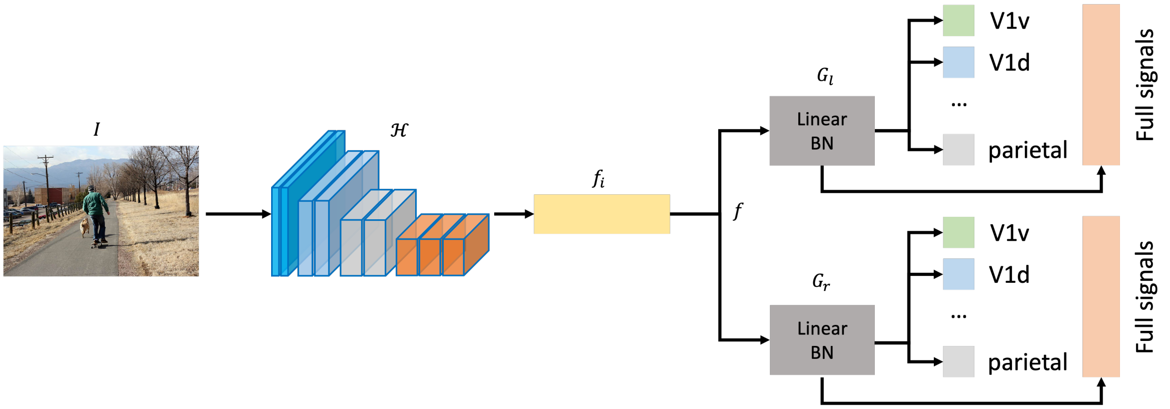

Figure 2: An overview of the fine-tuning stage.

The overview of the fine-tuning stage is illustrated in Fig 2. At this stage, we used the pre-trained weight of DNN $\mathcal{H}$ and fine-tuned the individual data. Apart from the previous stage and predicting the full fMRI signals, we also add multiple heads, i.e., fully connected layers, to predict each signal of each region-of-interests (ROIs), e.g., V1v, V1d, etc. The final prediction is an average of each individual vertices and the full signals.

3.3 Loss function

We utilize several loss functions in both stages. Along with the Smooth L1 loss function, we also implement the new loss functions as follows.

Mean Normalized Pearson Correlation (MNNPC) Loss. We implement the MNNPC loss function based on the evaluation metric defined in Eq. (1). The detailed implementation in PyTorch is described as in Algorithm (1).

Pearson Correlation Loss. We also implement the Pearson Correlation Loss which aims to optimize the challenge metric indirectly. The details of the implementation are shown in algorithm 2

4 Experiments and Results

4.1 Implementation Details

For each subject, we split the database into 5 folds where there are approximately 3900 and 1900 samples in training and validation. In both the pre-training and fine-tuning stages, the images are resized to 384 × 384. We select the embedding vector dimension of the subject $d_s = 512$ while the image feature dimension d_i depends on the DNN $\mathcal{H}$. All the code is easily implemented in PyTorch framework and trained by an A100 GPU. The learning rate is initially set to 0.0001 and then gradually reduced to zero under the ConsineLinear [7] policy. The batch size is set to 8/GPU. The model is optimized by AdamW [8] for 12 epochs or until the network meets the early stopping. The pertaining and fine-tuning are completed within 8 hours for each subject approximately.

4.2 Experiment Results

We have chosen `convnext_xlarge` [6] as the baseline for our experiment and `nn.Embedding` as the embedding $\mathcal{E}$. The results are presented in Table 4.3. The baseline, which is based solely on the Smooth L1 loss function without any fine-tuning, achieves a submission score of 54.21%. However, by implementing a pre-training strategy, we observe an

Algorithm 1 Mean Noise-Normalized Pearson Correlation Loss

```
class MNNPC(nn.Module):
    def __init__(self):
        super().__init__()

    def forward(self, pred, gt, nc=None):
        pred_mean = torch.mean(pred, axis=0)
        gt_mean = torch.mean(gt, axis=0)

        gt_t = gt.T
        pred_t = pred.T

        ts = (gt_t - gt_mean.view(-1, 1)) * (pred_t - pred_mean.view(-1, 1))
        ts = ts.sum(axis=1)

        ms1 = (gt_t - gt_mean.view(-1, 1)) ** 2
        ms1 = ms1.sum(axis=1)

        ms2 = (pred_t - pred_mean.view(-1, 1)) ** 2
        ms2 = ms2.sum(axis=1)
        ms = (ms1 * ms2) ** 0.5

        rv = ts / (ms + 1e-8)
        rv = 1 - (rv + 1) / 2

        if nc is not None:
            nc[nc == 0] = 1
            rv = rv**2 / torch.from_numpy(nc).to(rv.device)

        return rv.mean()
```

Algorithm 2 Pearson Correlation Loss

```
class PCLoss(nn.Module):
    def __init__(self):
        super().__init__()
        self.cos = nn.CosineSimilarity(dim=1, eps=1e-6)

    def forward(self, pred, gt, nc=None):
        pearson = self.cos(
            x1 - x1.mean(dim=1, keepdim=True), x2 - x2.mean(dim=1, keepdim=True)
        )
        pearson = pearson.mean()
        pearson = (pearson + 1) / 2
        return 1 - pearson
```

improvement of approximately 2.3%, bringing the score to 56.5%. Further enhancements are achieved by introducing two new loss functions, PCLoss and MNPCLoss, resulting in a submission score of 57.01%.

4.3 Ensemble

In addition to training `convnext_xlarge` as the baseline, we experimented with several other backbones, including `seresnext101d_32x8d` [5], `convnext_base`[6], `vit_small_patch16_224` [3], `efficientnet_b7` [9], `seresnet152d`[5] and `seresnextaa101d_32x8d`[5]. Each backbone underwent various training settings, including different combinations of loss functions, to further increase the diversity of the ensemble. Finally, a heuristic approach is used to compute the weighted average of all models. Consequently, the final submission achieves a submission score of 61.56%. Details of the submission results are illustrated in Fig. 3

Backbone $\mathcal{H}$	Pretraining	L1 Loss	PCLoss	MNPCLoss	Score (%)
`convnext_xlarge`	✗	✓	✗	✗	54.21
`convnext_xlarge`	✓	✓	✗	✗	56.50
`convnext_xlarge`	✓	✓	✓	✓	57.01

Table 1: Summary of experiment results on various loss functions and pretraining strategy

5 Conclusion and Discussion

Our solution makes significant contributions to the challenge in several key aspects. First, we introduce a pretraining stage that greatly enhances performance. Additionally, we propose two novel loss functions that effectively optimize the

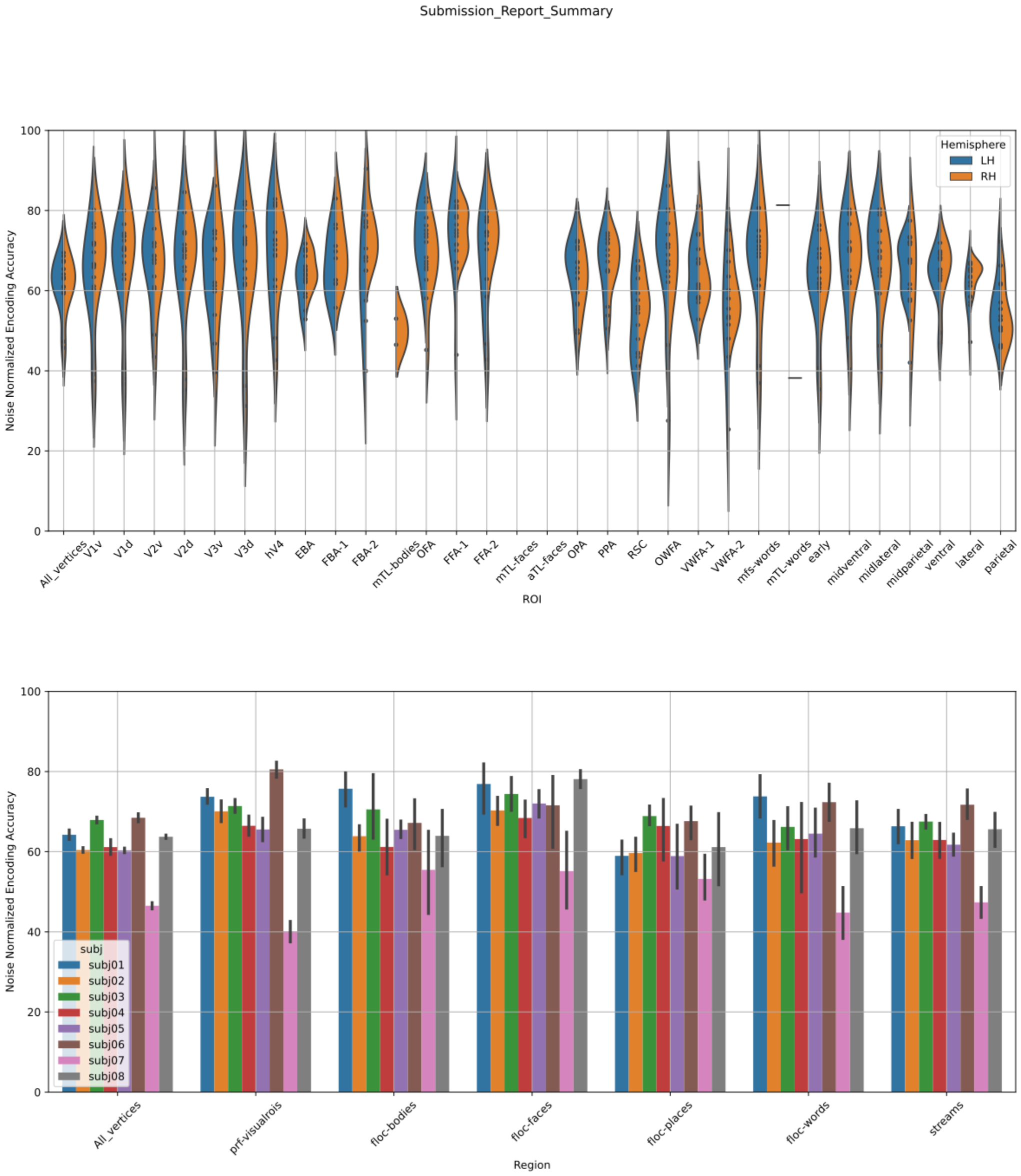

Figure 3: Submission report summary.

evaluation metrics. During the competition, we also identify several intra-challenges that need to be addressed. Notably, we observe that the choice of the backbone has a substantial impact on the final predictions. To improve upon our solution, future work will be delved into exploring network designs specifically tailored to the problem. Furthermore, we are interested in investigating the utilization of additional information from raw data for self-supervised learning, which could open promising research directions.

Acknowledgments

Our sincere gratitude goes to the Arkansas High-Performance Computing Center for generously providing GPUs for this challenge.

References

[1] E. J. Allen, G. St-Yves, Y. Wu, J. L. Breedlove, J. S. Prince, L. T. Dowdle, M. Nau, B. Caron, F. Pestilli, I. Charest, et al. A massive 7t fmri dataset to bridge cognitive neuroscience and artificial intelligence. *Nature neuroscience*, 25(1):116–126, 2022.

[2] R. M. Cichy, K. Dwivedi, B. Lahner, A. Lascelles, P. Iamshchinina, M. Graumann, A. Andonian, N. Murty, K. Kay, G. Roig, et al. The algonauts project 2021 challenge: How the human brain makes sense of a world in motion. *arXiv preprint arXiv:2104.13714*, 2021.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] A. T. Gifford, B. Lahner, S. Saba-Sadiya, M. G. Vilas, A. Lascelles, A. Oliva, K. Kay, G. Roig, and R. M. Cichy. The algonauts project 2023 challenge: How the human brain makes sense of natural scenes. *arXiv preprint arXiv:2301.03198*, 2023.

[5] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[6] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[8] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[9] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.