

Mapping the Human Connectome with Image Semantics via Topographic Factor Analysis

Seungwook Han
Columbia University
New York, USA
sh3264@columbia.edu

Hyun Dong Lee
Columbia University
New York, USA
hl2787@columbia.edu

Daniel Jeong
Columbia University
New York, USA
dpj2108@columbia.edu

Yong Hyun Cho
Columbia University
New York, USA
yc3135@columbia.edu

Abstract—We use the event-related fMRI dataset for natural scene imagery and other samples from standard computer vision datasets (COCO and ImageNet datasets) to investigate which regions/areas of the brain are responsible for responding to certain features in the images. We discovered a 5,000-image-based fMRI dataset from this paper CMU professors published recently this year [1]. By using clustering algorithms, we intend to first cluster the images into different groups and heuristically assess whether the different clusters do share key features. Then, we intend to apply topographic factor analysis (TFA) [2] to attempt to infer the brain networks that become stimulated from these clusters of images. By comparing and contrasting the brain networks within clusters and inter-clusters, we hope to answer some questions as to which key features different regions of the visual cortex may be responding to and confirm whether the different kinds of clustering can lead to a better approach to discovering the features that specific parts of our brain do.

I. INTRODUCTION

Avi Widgerson’s “Mathematics and Computation” points out that one of the greatest challenges in neuroscience is connectomics. It says that the task of mapping out the human brain and finding synaptic connections is computationally and algorithmically complex. Jeff Lichtman’s talk on “Mapping Neural Connections and Their Development” also highlights this issue by noting that one of the biggest challenges in the field of connectomics is not only the size of the brain imaging data necessary to map out every cell but also the time it takes to image a mm^3 segment of the brain. Despite the evident difficulties, many neuroscientists believe that connectomics will prove to be valuable in understanding how the brain works. Prof. Sebastian Seung, one of the leading neuroscientists in the area of connectomics today, once claimed he believes in the hypothesis that “we are our connectome.” Inspired by these comments, we would like to investigate the correlation between different areas of the brain and certain external stimuli; more specifically, we want to find the parts of the brain that are involved in processing image semantics. To infer the significant neural patterns and map them to image semantics, we apply TFA to the BOLD5000 fMRI dataset.

II. RELATED WORK

A. Topographic Factor Analysis

In 2014, Manning., et al. published [Topographic Factor Analysis: A Bayesian Model for Inferring Brain Networks

from Neural Data]. This manuscript describes a novel learning method using Bayesian framework to infer the network of interaction between different activity ‘hubs’ in the brain during an fMRI experiment. It is believed that fMRI data often exhibit spatial correlation and thus topographical factor analysis exploits the inherent characteristics to infer the latent structures of fMRI data. In essence the topographical factor analysis framework looks for an area of highest activity of BOLD activity. Then the algorithm initializes the source centers, widths and weights. Subsequent sources are found by looking for peak activity area in the resulting images with the previous source subtracted. The sources are parameterized using Gaussian radial functions and thus, sources are modeled as sphere volumes.

B. BOLD5000

The BOLD5000 dataset compromises of image data and fMRI data. The images consisted of scene images, COCO images and ImageNet images. The images in the BOLD5000 dataset were used as stimuli during fMRI acquisition. Subjects were shown a random set of images in randomly predetermined order for one second and shown a blank fixation cross for nine seconds. For each run of the experiment, subjects were given a total of 37 stimuli and each subject underwent at least nine sessions. At the beginning of every run, there was a six second calibration period where the subject was given a blank fixation cross and the same fixation cross was shown for twelve seconds at the end of each run.

C. Feature Extraction from Image Data

For feature extraction, we mainly took three different approaches: generating filters using deep convolutional neural networks that are well-known for their performance in object detection/image classification tasks, reconstructing the given images from phase only, and gathering statistics from the distribution of the RGB values. For the final approach, we referred to the paper “Image Clustering Using Color and Texture” by Maheshwari et al., where they used color features consisting of the mean, standard deviation, and skewness of the raw RGB values and texture features generated using Gabor filters to perform K-Means clustering.

D. Phase

In the field of digital image processing, phase response is considered to be an integral part in determining structure in an image. However, phase response is often neglected due to its non-linear nature which makes it particularly difficult to analyze. In this preliminary experiment, we show why and to what degree phase response is significant in determining a structure of an image. In doing so, we swapped the phase response of two images. The phase swapped images show resemblance to the image from which the phase response was calculated from. Thus confirming the importance of phase information in an image. The results of this experiment can be found in Figure 1.

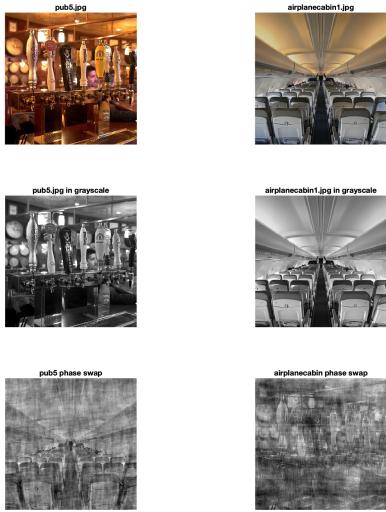


Fig. 1. Phase Swap Experiment

III. EXPERIMENTS & METHODS

A. Source Code: GitHub Repository

<https://github.com/hanseungwook/tfabold5000>

B. Phase Only Image Reconstruction

Image reconstruction using exclusively phase response information was carried out using MATLAB. Each color scene image in the BOLD5000 image dataset was converted to a grayscale image. Then Fourier transformation algorithm (fft2) was applied to decompose the image into phase and magnitude responses. Using only the phase response, inverse Fourier transform was applied to reconstruct the image. Because the resulting reconstructed image had poor contrast and low intensity, a histogram equalization algorithm was applied to better capture the features significant in the image. An example of the resulting image is shown in Figure 2 below.

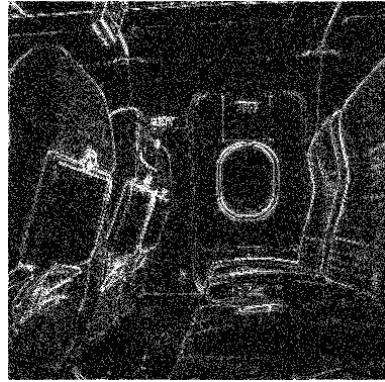


Fig. 2. Phase Reconstructed Image of airplanecabin6.jpg

C. Phase Spectrum Image Generation

Phase spectrum image was generated from all scene images in the BOLD5000 dataset. All data processing was performed using MATLAB. Each color scene image in the BOLD5000 image dataset was converted to a grayscale image. Fast Fourier transform algorithm (fft2) was applied to decompose the image into phase and magnitude responses. Rather than taking the inverse Fourier transform to reconstruct an image, the absolute value of the phase response was taken. Then, the minimum and the maximum value was calculated to determine pixel intensity on a scale from 0 to 255. An example of a phase spectrum image is shown in Figure 3.

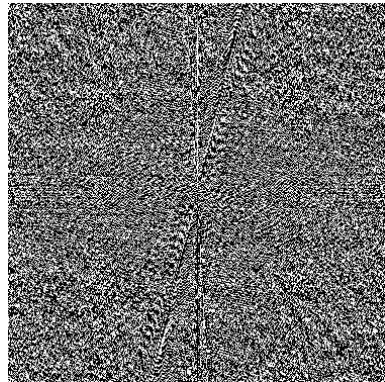


Fig. 3. Phase Spectrum Image of track3.jpg

D. fMRI Preprocessing

All fMRI data were converted from DICOM format into NIfTI format (NIfTI <https://nifti.nimh.nih.gov/>) using the dcm2nii tool [<https://www.ncbi.nlm.nih.gov/pubmed/26945974>].

Results included in this manuscript come from preprocessing performed using FMRIB Software Library (FSL) v6.0 [<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki>]. Standard preprocessing steps, bias field correction, brain extraction, motion correction, registration and standardization, was applied to all experimental runs. Bias field correction of both structural and functional data were performed using the FAST tool in FSL. During

skull stripping and brain extraction using BET tool, a binary brain mask of valid brain region was also created. And to ensure that voxels outside the brain region have precisely the value zero, brain extracted functional data was once more multiplied by the binary brain mask using FSLMATHS. Then, functional data was motion corrected using the MCFLIRT tool. Registration of functional data to high resolution structural data was performed using FEAT in FSL. Standardization was performed by subtracting the temporal mean of each voxel from the functional data and dividing the result by a temporal standard deviation of each voxel. Voxelwise calculation was carried out using FSLMATHS.

E. Topographic Factor Analysis

Given clusters of images, one image was randomly selected from each cluster. For each image, BOLD5000 dataset specifies when during the experiment the image was shown to a subject. Using this information, fMRI data relevant to a given image were extracted, and a spatial filtering kernel was applied to the data.

Brain Imaging Analysis Kit (BrainIAK) [<http://brainiak.org/>] supports visualization and Topographic Factor Analysis (TFA) of fMRI data. BrainIAK was used to visualize the voxel activations and voxel locations, run TFA to locate network hubs that characterize the data in a highly compact form, run Hierarchical TFA (HTFA) to find network hubs that reflect multiple subjects data, and find connectivity among the network hubs.

For the experiment, RGB distribution method was used for clustering with $K = 3$. Then, we compared and contrasted the following:

1) Brain networks inferred from two subjects' responses to the same image (Case I)

This comparison will be the baseline for other experiments involving multiple subjects. The two subjects were Subject 2 and Subject 3, and the image was *rep_homeoffice9.jpg* from BOLD5000's Scene images.

2) Brain networks inferred from single subject's responses to two different images from different clusters (Case II)

This comparison will test our hypothesis.

F. Feature Extraction

In order to investigate the relationship between the image stimuli in the BOLD5000 dataset and the different regions of the connectome that are responsible for processing the respective stimuli, we needed to cluster the images in meaningful ways that closely resemble how a human may differentiate images. Deep convolutional networks are considered to be the state-of-the-art models for image classification and have demonstrated performance on par with humans. Therefore, using pre-trained models of VGG16, VGG19, ResNet50, and InceptionV3, we first attempted to extract relevant features from the original image stimuli, later to be used in the clustering algorithm.

However, with over 60,000 features extracted by these deep convolutional neural networks per image, it resulted in poor,

overlapping clusters – to be discussed later in the 'Results' section.

Therefore, we refined the scope of feature extraction to two types: phase-based method and color-based methods (Dominant Color, RGB Histogram, RGB Distribution).

1) VGG16, VGG19, ResNet50, InceptionV3

To extract relevant features through these four deep convolutional neural networks, we used the pre-trained models on the ImageNet database provided by the Keras library.

The code that processes the feature extraction through these four models is *feature_extractor.py* in the GitHub repository.

2) Dominant Color

As a simplistic feature to represent the general color of each image stimuli, we pick out the dominant color for each image. To extract the dominant color, we use K-Means clustering on the raw pixels that consist of the (B, G, R) values with $k = 1$ (number of clusters), as illustrated in Figure 4. Then, we use the centroid point of the cluster to be dominant color representation of the images.

The script extracting the dominant color and constructing the feature array is *color_extractor.py*

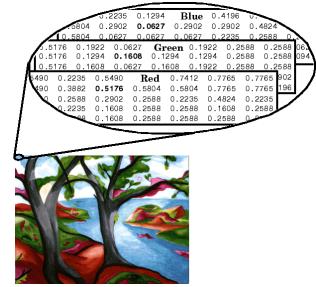


Fig. 4. B,G,R Representation of Image Pixels, adopted from Mathworks

3) RGB Histogram

The Dominant Color method rather simplifies the image significantly because we only choose one dominant color from the (396×396) dimensional image.

With the RGB Histogram, we attempt to obtain a more holistic feature view of the image by constructing a histogram of the RGB values as a matrix. First, before building the histogram matrix, we use histogram equalization in order to more evenly spread out the value distribution in the image and thereby achieve a higher contrast in the images. After performing histogram equalization, we can see from Figure 5 that the distribution of the image evens out. For the histogram matrix, we divide the R, G, and B values with 4, 8, and 4 bins respectively i.e. Bin 1 for values in range $[0, 63]$, Bin 2 for values in range $[64, 127]$, Bin 3 for values in range $[128, 191]$, and Bin 4 for values in range $[192, 255]$. The total range $[0, 255]$ is divided into b equally-sized bins. After the preprocessing step, we iterate through each pixel of the original image and increment the counts for the respective bins the pixel's R, G, and B values fall in, ultimately attaining the final matrix with the count of all pixels that contains values in the respective bins.

The script for extracting and constructing this feature is in `make_histogram.py`.

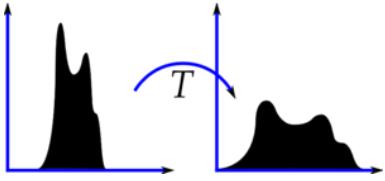


Fig. 5. Histogram Equalization, adopted from OpenCV

4) RGB Distribution

The RGB Distribution method uses color moments as measures to differentiate images based on their features of color. The underlying assumption in this feature is that the color of the image can be described as a probability distribution. With this assumption, we use moments of the distribution (mean, standard deviation, and skew) to identify the image in terms of color. The three moments, mean, standard deviation, and skew, that we have decided as the features are inspired by Stricker and Orengo [3].

In the following equations, c represents the color component (R, G, and B), p_{ij} represents the pixel of the image at height i and width j , and P and Q represent the width and height of the image.

MOMENT 1 - Mean

$$E_c = \frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q p_{ij}^c$$

Mean is the measure of the mean of the distribution – mean color of the image.

MOMENT 2 - Standard Deviation

$$SD_c = \sqrt{\frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q (p_{ij}^c - E_c)^2}$$

Standard deviation is the measure of variance of the distribution.

MOMENT 3 - Skew

$$S_c = \left(\frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q (p_{ij}^c - E_c)^3 \right)^{\frac{1}{3}}$$

Skew is the measure of the degree of asymmetry in the distribution.

Before these features were extracted, the same histogram equalization preprocessing step explained in the RGB Histogram method was also executed for this feature extraction method.

G. Clustering

Using the features from above, we performed K-Means clustering to find semantically (e.g. RGB color distribution, etc.) similar clusters of images. To find the “optimal” (i.e. best performing) k value, we designed the experiments to cluster the given features for multiple values of K and calculated the silhouette scores, a widely-used clustering performance metric which ranges between -1 and +1 (inclusive). Silhouette coefficients near +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample

is on or very close to the decision boundary between two neighboring clusters, and negative values indicate that those samples might have been assigned to the wrong cluster. Given a path to the images in the dataset, a path to the features to use, and the set of k values to experiment with, the script `kmeans.py` outputs the cluster assignments made for each k value and generates a scatter plot of silhouette scores vs. k values.

To visually inspect and analyze the quality of our clusters, we also used principal component analysis (PCA) and t-SNE (t-distributed stochastic neighbor embedding), a nonlinear dimensionality reduction technique that, for each pair of data points, finds the “optimal” lower dimensional probability distribution that minimizes the Kullback-Leibler divergence between it and the original probability distribution in high dimensional space [4]. Running the `cluster_visualizer.py` script, with the appropriate parameters (see documentation in GitHub repository), allows one to plot the clusters generated.

IV. RESULTS

A. Clustering

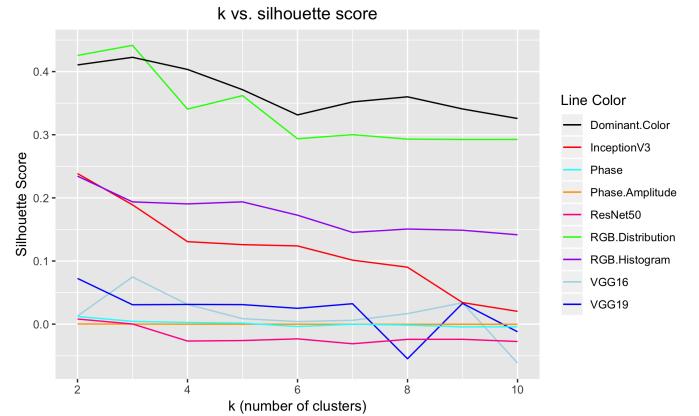


Fig. 6. k vs. Silhouette Score for each clustering based on different feature models

Figure 6 is a plot of k (number of clusters) vs. silhouette score, a metric to estimate the validity of consistency and similarity of data points within its respective cluster, in the different feature models that we explained above.

Figure 7 is the plot of the clusters that we produced through K-Means clustering using the RGB Distribution as the feature model. At $k = 3$, this feature attained the highest silhouette score amongst all other feature models.

Figure 8 is the plot of the clusters that we produced through K-Means clustering using ResNet50 (deep CNN) as the feature model. At $k = 2$, this feature attained its highest silhouette score, but demonstrated the lowest silhouette score amongst all other feature models.

Figures 9, 10, and 11 below are samples of images from each of the $k = 3$ clusters when contrast in the scene images was increased and RGB distribution features were used (one of the best performing cases).

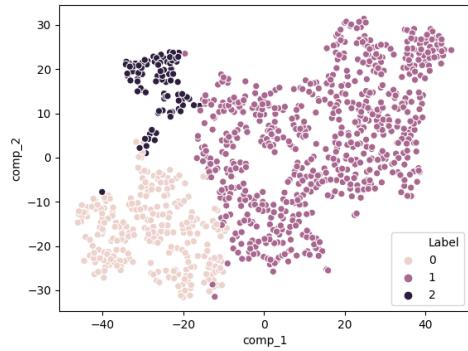


Fig. 7. Clusters ($k=3$) with RGB Distribution as Feature Model

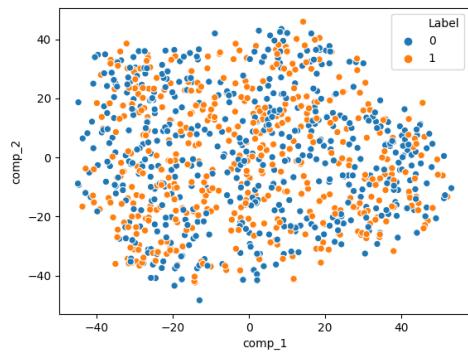


Fig. 8. Clusters ($k=2$) with ResNet50 as Feature Model

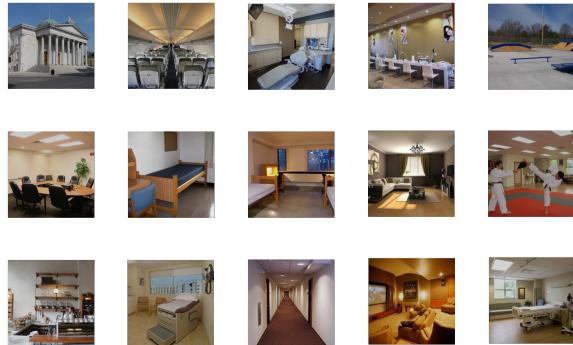


Fig. 9. Sample images from Cluster 1

B. Topographic Factor Analysis

1) Voxel Locations (Case I)

Figure 12 is a visualization of the locations of voxels of Subject 2 during the experiment.

2) Original fMRI data (Case I)

Figure 14 and Figure 15 visualize the difference between fMRI data collected from Subject 2 and those collected from Subject 3 when they were shown the same image, *homeoffice9.jpg*.

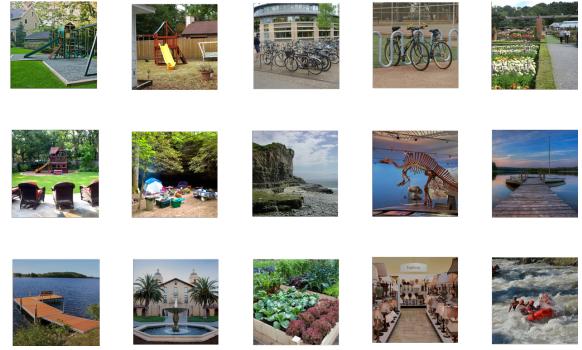


Fig. 10. Sample images from Cluster 2

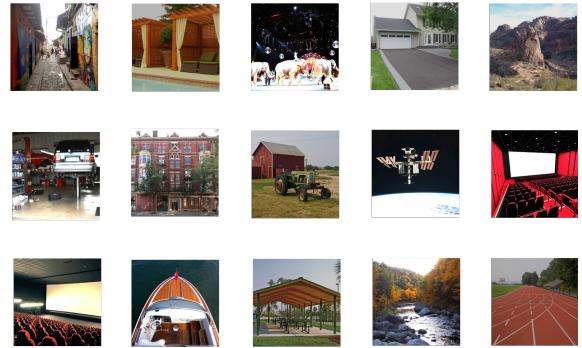


Fig. 11. Sample images from Cluster 3

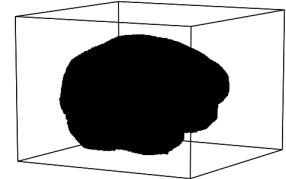


Fig. 12. Voxel Locations of Subject 2 looking at *homeoffice9.jpg*



Fig. 13. Original *homeoffice9.jpg*

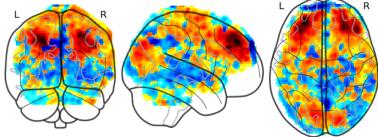


Fig. 14. Original fMRI data from Subject 2 looking at *homeoffice9.jpg*

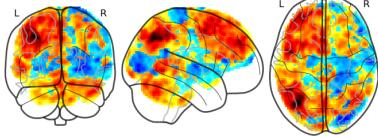


Fig. 15. Original fMRI data from Subject 3 looking at *homeoffice9.jpg*



Fig. 16. HFTA Global and Local Hub Locations of Subject 2 and Subject 3 looking at *homeoffice9.jpg* (Case I)

- 3) HFTA Global and Local Hub Locations (Case I)
- 4) Inter-subject Functional Connectivity (Case I)

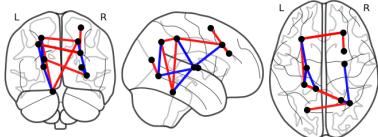


Fig. 17. Inter-subject Functional Connectivity between Subject 2 and Subject 3 looking at *homeoffice9.jpg* (Case I)

Figure 17 shows which parts in the brains are correlated when two subjects are given the same stimulus.

- 5) HFTA Global and Local Hub Locations (Case II)

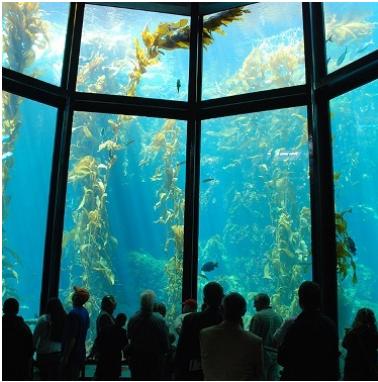


Fig. 18. Original *aquarium6.jpg* (Cluster 1)

- 6) Inter-cluster Functional Connectivity (Case II)

Figure 21 shows which parts in the brain are correlated when given two different stimuli.



Fig. 19. Original *garbagedump4.jpg* (Cluster 2)

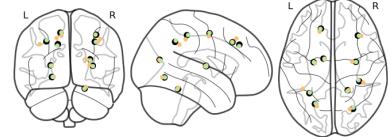


Fig. 20. HFTA Global and Local Hub Locations of Subject 3 looking at *aquarium6.jpg* (Cluster 1) and *garbagedump4.jpg* (Cluster 2) (Case II)

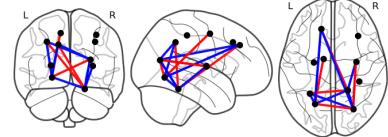


Fig. 21. Inter-cluster Functional Connectivity of Subject 3 looking at *aquarium6.jpg* (Cluster 1) and *garbagedump4.jpg* (Cluster 2) (Case II)

V. DISCUSSION

A. Clustering

When features generated from deep convolutional neural networks were used for clustering the images, the silhouette scores were always close to zero and often negative for some values of k . Note that the closer the silhouette score is to -1, it indicates that the data points are dissimilar to their respective clusters and the clusters are not well-formed; the closer the silhouette score is to +1, it indicates that the data points are very similar to their respective clusters and the clusters are well-formed; a silhouette score near 0 indicates that the clusters overlap with each other significantly.

Despite being powerful representations for object detection/classification tasks, we noticed that using CNN-based features to directly cluster the images are not as useful. As for the poor performance of the CNN filters and methods, we conjectured that it may be due to the wide spectrum of categories from which the images are derived and the relatively small number of samples in each different category. Essentially, the CNN models are extracting necessary features for image classification and our dataset may contain too many different categories of images for the K-Means clustering algorithm to cluster them in respect to the derived features.

Moreover, we observed that the CNN models use between 60,000 and 90,000 features for each image, and we believed that this complex set of features' relationship to the brain activation would eventually be undecipherable. An example of clusters created by a deep neural network (ResNet50, in this case) is shown in Figure 8.

The initial motivation behind using phase-based methods (1. Phase 2. Phase Amplitude) was that studies have found that images can be dissected into two components, phase and amplitude, and that humans respond more significantly to the changes in the phase of the image. Despite this hypothesis, in our own clustering algorithm, phase-based methods had an average silhouette score of 0.0003, indicating that the clusters overlap quite significantly.

In contrary to the previous two approaches of using deep neural networks and phase for feature extraction, color-based approach demonstrated success in attaining reasonable silhouette scores. The average silhouette score of the Dominant Color method is approximately 0.37, that of the RGB Distribution method 0.34, and that of the RGB Histogram method 0.17. Maximum silhouette score was attained with $k = 3$ with the RGB Distribution method.

The visualization of the clusters through the t-sne and PCA methods are shown in Figure 7 and the actual images in Figures 9 10 11. We can see from Figure 7 that in comparison with the clusters created through the ResNet50-extracted features, the clusters are clearly distinct and separated. Moreover, with the actual images, we can notice that Cluster 1 gathered white and beige-dominant color images, Cluster 2 gathered green and blue-dominant images, and Cluster 3 gathered red-dominant images. Therefore, we decided to use the RGB Distribution-based clusters to investigate whether they lead to any tangible correlations with some areas of the connectome.

B. Topographic Factor Analysis

The Case I experiment verifies that, when two people are given a single image, their responses to the image are similar. Subject 2 and Subject 3 are shown the same image, *homeoffice9.jpg*, and Figure 16 shows that their local hubs (orange dots for Subject 2 and green dots for Subject 3) are located close to corresponding local hubs.

The Case II experiment, Figure 20, shows that, when a single person is shown two different images, the response for each image is noticeably different. Orange dots are placed in the Subject 3's brain that are activated when shown *aquarium6.jpg*, and green dots are responses to *garbagedump4.jpg*. Compared to Figure 16, the orange and green dots are noticeable apart from each other.

Case I and Case II experiments show that there is a correlation between the color distribution of the visual stimuli and the activations in the brain. According to "Categorial Encoding of Color in the Brain" by Bird et. al [7], the researchers conclude that there is little to no correlation between the color category and the visual cortex and that the visual cortex only responds to color differences. Perhaps, it is true that an individual color itself does not activate different regions of the visual

cortex. Nevertheless, in an image where the visual stimuli consist a distribution of colors, color differences are bound to exist. Because our selected feature extraction model – RGB Distribution – accounts for the distribution of different colors using moments, it seems that our fMRI data demonstrate some tangible differences depending on the cluster.

In fact, "Three cortical stages of colour processing in the human brain" by Zeki and Marina [?] illustrates the three stages of cortical color vision and their color pathways. Comparing their illustrations of activation locations in the brain with our fMRI activations shown in Figures 14 and 15, they seem to significantly overlap with each other.

CONCLUSION & FUTURE WORK

In clustering the images into semantically similar clusters, we experimented multiple approaches spanning from using CNN features to using RGB-distribution-based features and found that the results were better when we increased the contrast in the images and used features based on RGB statistics. While we were able to find reasonably separated clusters by doing so, because the k values for which the silhouette scores were high were rather small, it was often difficult to find and understand the semantics common among the images in a given cluster. In the future, we hope to be able to find better methods to cluster the images such that we are able to infer tangible semantics for the clusters generated and map them to different cortical areas of the brain. We would also like to find methods to measure the similarity or difference between local hubs of two different subjects or two different cluster rather than using visual estimation. In addition, we would like to study the meaning of Inter-subject Functional Connectivity more deeply and investigate how activation locations in the brain are connected to each other and the biological reason behind it.

ACKNOWLEDGMENT

We would like to thank Professor Papadimitriou for a wonderful semester and Daniel Mitropolsky for advising and helping us refine our project.

REFERENCES

- [1] Blei, David M., et al. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, vol. 3, no. 993, ser. 1022, 2003. 1022.
- [2] Chang, Nai Chen, et al. Scaling Up Neural Datasets: A Public FMRI Dataset of 5000 Scenes. *Journal of Vision*, vol. 18, no. 10, 2018, p. 732., doi:10.1167/18.10.732.
- [3] Stricker, M. and Orengo, M., Similarity of color images, Storage and Retrieval for Image and Video Databases III (SPIE) 1995: 381-392.
- [4] van der Maaten et al., "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, 2008.
- [5] Image Types, https://www.mathworks.com/help/matlab/creating_plots/image-types.html, 2018.
- [6] Histogram Equalization, https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html, Dec 18 2015.
- [7] Bird, C. M., Berens, S. C., Horner, A. J. Franklin, A. "Categorical encoding of color in the brain." *Proc Natl Acad Sci USA* 111, 4590, 2014.
- [8] Zeki, S., Marini, L. "Three cortical stages of colour processing in the human brain." *Brain*, 1998.