

Final Project for PDF

April 11, 2018

Table of Contents

- 1 Background
- 2 Process
 - 2.1 Mapping Eccentricity
 - 2.2 ROI Masking Based on Eccentricity
 - 2.3 Accuracy Calculation
- 3 Data
 - 3.1 All Subject Data
 - 3.2 Plots
 - 3.3 V1 - V4 Accuracies
 - 3.4 Higher Level Visual Region Accuracies
- 4 Analysis
 - 4.1 Effect Size
 - 4.2 Spatial Arrangement
- 5 Discussion
- 6 References

PSY 5212: Final Project

Amin Banihashemi, Hause Lin & Daniel J Wilson

Note: If possible, please view this via the Binder link. This will allow the interactive running of the Jupyter notebook (you do not need to download anything), so you can recreate plots to explore the data yourself. Just click the link "launch binder" below. Note that it will take a few minutes to launch (it is building a custom Python environment using Docker).

1 Background

While our eye has a wide range of vision, the 1.5mm wide fovea is by far the most sensitive area, where fine detail and color may be distinguished ("Fovea centralis," 2018).

The central fovea in the image below (Figure 1) is shown to cover 5 degrees, however we found general agreement that the foveal region could be assumed to cover 2 degrees - roughly the size of a thumb held at arms length (Fairchild, 2013). We used this 2 degree number to define the fovea for our project.

Figure 1. Peripheral vision of the human eye

The goal of our project was to look at areas of the visual regions that responded differentially to foveal and non-foveal (peripheral) stimuli, and examine how these areas, when analyzed on their own, might predict the scene categories used in the experiment (faces, objects, scenes, scrambled).

2 Process

2.1 Mapping Eccentricity

The fact that the visual cortex is retinotopically organized means that adjacent neurons tend to have similar receptive fields. This can be extended to the voxel level, leading to the idea that each voxel, and the neurons it contains, will have at least somewhat similar receptive field properties. This is referred to as a population receptive field, or pRF (Victor et al., 1994).

We can calculate the pRF size based on an understanding of how the pRF size influences the fMRI time course (Wandell and Doumoulin, 2008). We can also calculate eccentricity, a measurement of distance from a central fixation point, as there is a systematic relationship between duty cycle and visual eccentricity (Smith et al., 2001).

We computed pRFs for all three subjects in matlab using the `runPRFanalysis.m` script, written by Dirk Bernhardt-Walther.

Converting Eccentricity to Visual Angle

This step first required entering the measurements from the experimental setup, in our case these were:

```
stim_distance_cm = 228.3;
```

```
stim_diameter_cm = 27.8;
```

```
stim_diameter_pixels = 768;
```

With these data we were able to run the following commands in matlab to get the eccentricity of each PRF in degrees:

- 1) Look at the eccentricity results for the voxels of a given ROI

```
ROI_eccentricity = results.ecc(roiIdx);
```

- 2) Calculate the eccentricity in centimeters

```
ecc_cm = ROI_eccentricity / stim_diameter_pixels * stim_diameter_cm;
```

- 3) Confirm that the eccentricity value in cm is valid (i.e. the value is equal to or smaller than the actual size of the stimulus)

```
ecc_valid = (ecc_cm <= stim_diameter_cm);
```

- 4) Use the inverse tangent function calculate the degrees of eccentricity (distance from the center of the stimulus).

```
ecc_degrees = atand(ecc_cm/stim_distance_cm);
```

2.2 ROI Masking Based on Eccentricity

With eccentricity converted to visual angle, we used AFNI's `3dcalc` function to separate voxels representing the two eccentricity categories of interest.

To create a mask isolating the eccentricities of 2 degrees and below we ran the following code:

```
3dcalc -a ../masks/0003_V1_clean_mask+orig -b 0003_PRFresults_bucket+orig[6] -expr 'step(a) * st
```

Where `a` is the original ROI mask and `b` is the eccentricity results in degrees for the whole brain. The `step` function then used the math $2-b$ to remove any voxels from the original ROI mask which were greater than 2.

The opposite math, $b-2$ was used to isolate the non-foveal eccentricities (angles greater than 2 degrees), with the following code:

```
3dcalc -a ../masks/0003_V1_clean_mask+orig -b 0003_PRFresults_bucket+orig[6] -expr 'step(a) * st
```

2.3 Accuracy Calculation

Accuracies were calculated in Matlab.

For classification we used `libsvm`, and the raw data, as in a previous assignment this combination had given the highest classification accuracy.

We ran the following code in Matlab in order to obtain classification accuracy and a confusion matrix:

- 1) Select a subject (e.g. 0003), foveal/peripheral ROI (e.g. V1_fov).

Prediction Reality	Face	Object	Place	Scrambled	Total
Face	3	4	1	0	8
Object	2	4	2	0	8
Place	0	2	6	0	8
Scrambled	2	1	1	4	8

title

```
dsetsV1_fov = loadROIData('0003', 'V1_fov', 'raw');
```

- 2) Check classification accuracy values and create a confusion matrix (see Figure 2) to visualize where errors are being made.

```
[accV1_fov, cmV1_fov]=classifyFPO(dsetsV1_fov);
```

This was repeated for all ROIs for all subjects at both foveal (less than 2 degrees) and peripheral (greater than 2 degrees) eccentricities.

Figure 2. Sample confusion matrix output. The diagonal values are used to calculate the accuracy of predictions. Each diagonal value was divided by 8, which is the total number of predictions, and expressed as percentage to yield the accuracy.

3 Data

Below is a summary of all data.

Import Libraries

```
In [2]: import pandas as pd
import numpy as np
import scipy as sp
import plotly
import plotly.plotly as py
import plotly.figure_factory as ff
```

```
In [3]: plotly.tools.set_credentials_file(username='dapper2733', api_key='sbn0WN1I8CGjjK5MzXae')
```

Subject	Category	Region	Foveal	Accuracy
2	Face	V1	1	25.0
2	Face	V2	1	12.5
2	Face	V3	1	87.5
2	Face	V4	1	75.0
2	Face	OPA	1	75.0
2	Face	PPA	1	75.0
2	Face	RSC	1	25.0
2	Face	FFA	1	100.0
2	Face	LOC	1	100.0
2	Face	V1	0	25.0
2	Face	V2	0	50.0
2	Face	V3	0	37.5
2	Face	V4	0	50.0
2	Face	OPA	0	75.0
2	Face	PPA	0	62.5
2	Face	RSC	0	25.0
2	Face	FFA	0	100.0
2	Face	LOC	0	75.0
3	Face	V1	1	75.0

title

```
In [4]: plotly.offline.init_notebook_mode(connected=True)
```

```
In [5]: ROIs = ['V1', 'V2', 'V3', 'V4', 'OPA', 'PPA', 'RSC', 'FFA', 'LOC']
        categories = ['Face', 'Object', 'Place', 'Scrambled']
```

3.1 All Subject Data

```
In [6]: df = pd.read_csv("http://www.danieljwilson.com/data/Accuracies.csv")
```

```
table = ff.create_table(df)
```

```
plotly.offline.iplot(table)
```

Figure 3. Accuracy for each subject, for each category, for each region, for both foveal and peripheral pRFs.

Create mean of subjects

```
In [7]: Subject = 5
```

```
d = []
```

```
for category in categories:
    for foveal in range(0,2):
        for ROI in ROIs:
```

```
            mean = (float(df.Accuracy[(df.Subject==2) & (df.Foveal==foveal) & (df.Region
```

```

d.append({'Subject': Subject, 'Category': category, 'Region': ROI, 'Foveal':
d = pd.DataFrame(d)

```

Add subject 5 (mean of subjects 2,3,4) to dataframe

```
In [8]: df = pd.concat([df, d])
```

3.2 Plots

Examine data by subject and region

```
In [9]: from plotly.graph_objs import *

# Plotting function

def ROI_accuracies(Subject: int, Category: str) -> None:
    """
    Draws a plotly plot based on Subject and stimulus Category
    """
    trace_foveal = Bar(x=df.Region[(df.Subject==Subject) & (df.Foveal==1) & (df.Category=
                        y=df.Accuracy[(df.Subject==Subject) & (df.Foveal==1) & (df.Categor
                        name='Foveal',
                        marker=dict(color='#59606D'))

    trace_non_foveal = Bar(x=df.Region[(df.Subject==Subject) & (df.Foveal==0) & (df.Cate
                        y=df.Accuracy[(df.Subject==Subject) & (df.Foveal==0) & (df.Category=
                        name='Non-Foveal',
                        marker=dict(color='#A2D5F2'))

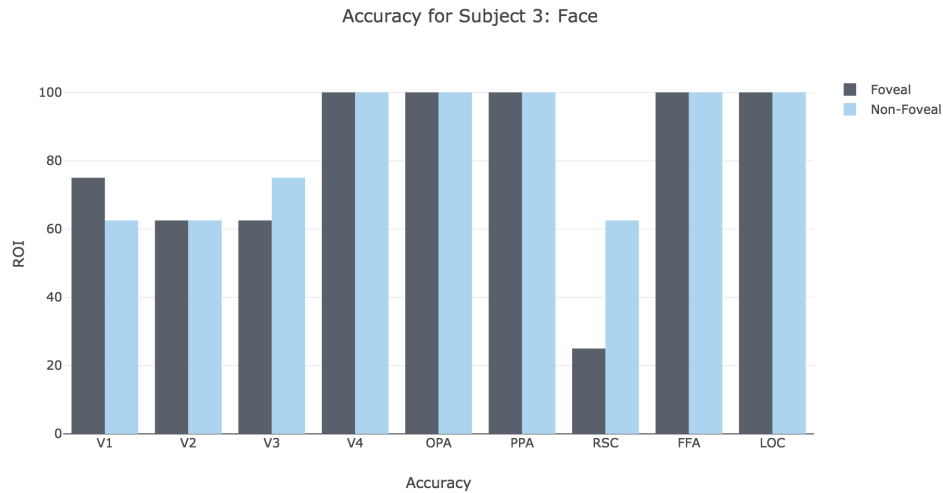
    data = [trace_foveal, trace_non_foveal]

    layout = Layout(title="Accuracy for Subject " + str(Subject) + ": " + str(Category),
                    xaxis=dict(title='Accuracy'),
                    yaxis=dict(title='ROI'))

    fig = Figure(data=data, layout=layout)

    plotly.offline.iplot(fig)

```



title

Interactive Plot

To dynamically view any subject and category you like, just change the Subject and Category variables in the function call below (ROI_accuracies).

Subject: {2, 3, 4, 5} subject 5 = mean of 2, 3 & 4

Category: {'Face', 'Object', 'Place', 'Scrambled'}

```
In [11]: ROI_accuracies(3, 'Face')
```

Figure 4. Foveal and Peripheral categorization accuracy for all regions for a selected subject and category.

Examine data across subject by category and region

```
In [12]: import plotly.graph_objs as go
```

```
# Plotting function
```

```
def fov_vs_peripheral(Region: str, Category: str) -> None:
```

```
    """
```

```
    Plot change from foveal to peripheral for all subjects as well as a mean
    line using plotly.
```

```
    """
```

```

sub2_y1 = float(df.Accuracy[(df.Subject==2) & (df.Foveal==1) & (df.Category==Catego
sub2_y2 = float(df.Accuracy[(df.Subject==2) & (df.Foveal==0) & (df.Category==Catego
sub3_y1 = float(df.Accuracy[(df.Subject==3) & (df.Foveal==1) & (df.Category==Catego
sub3_y2 = float(df.Accuracy[(df.Subject==3) & (df.Foveal==0) & (df.Category==Catego
sub4_y1 = float(df.Accuracy[(df.Subject==4) & (df.Foveal==1) & (df.Category==Catego
sub4_y2 = float(df.Accuracy[(df.Subject==4) & (df.Foveal==0) & (df.Category==Catego

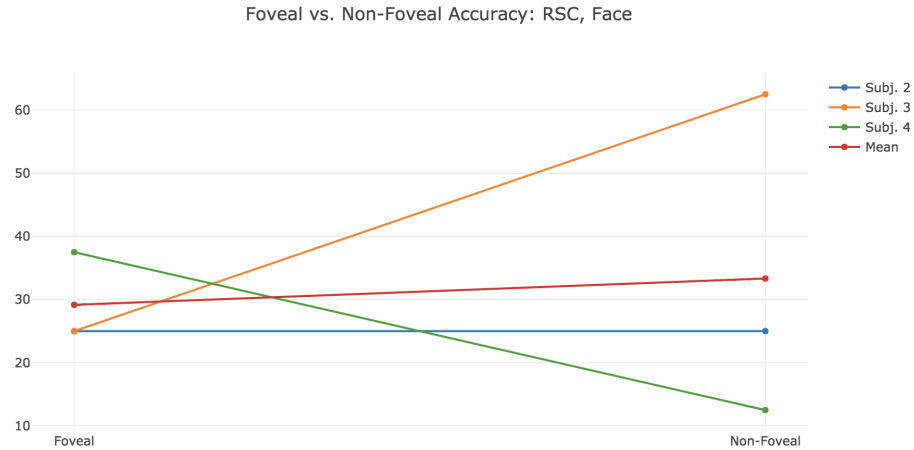
# Create traces
subject_2 = go.Scatter(
    x = ['Foveal', 'Non-Foveal'],
    y = [sub2_y1,
          sub2_y2],
    mode = 'lines+markers',
    name = 'Subj. 2'
)
subject_3 = go.Scatter(
    x = ['Foveal', 'Non-Foveal'],
    y = [sub3_y1,
          sub3_y2],
    mode = 'lines+markers',
    name = 'Subj. 3'
)
subject_4 = go.Scatter(
    x = ['Foveal', 'Non-Foveal'],
    y = [sub4_y1,
          sub4_y2],
    mode = 'lines+markers',
    name = 'Subj. 4'
)
subject_mean = go.Scatter(
    x = ['Foveal', 'Non-Foveal'],
    y = [(sub2_y1 + sub3_y1 + sub4_y1)/3,
          (sub2_y2 + sub3_y2 + sub4_y2)/3],
    mode = 'lines+markers',
    name = 'Mean'
)

data = [subject_2, subject_3, subject_4, subject_mean]

plotly.offline.iplot({
    "data": data,
    "layout": Layout(title="Foveal vs. Non-Foveal Accuracy: " + str(Region) + ", "
})

```

Interactive Plot



title

To dynamically view differences in accuracy for all subjects between foveal and peripheral pRFs just change the Region and Category variables in the function call below (fov_vs_peripheral).

Region: {'V1', 'V2', 'V3', 'V4', 'FFA', 'LOC', 'OPA', 'PPA', 'RSC'}
 Category: {'Face', 'Object', 'Place', 'Scrambled'}

```
In [13]: fov_vs_peripheral('RSC', 'Face')
```

Figure 5. Foveal vs Peripheral categorization accuracy for all subjects for a given ROI and category.

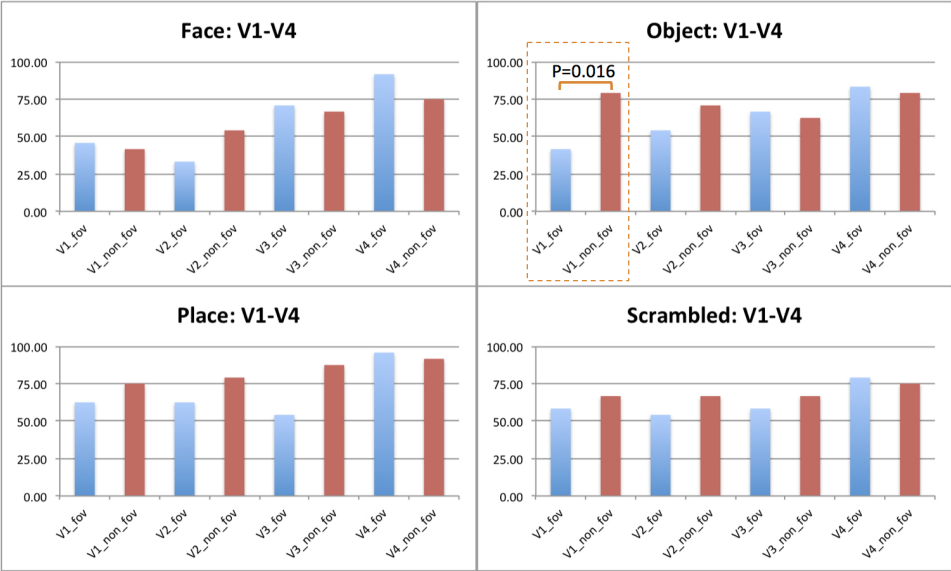
3.3 V1 - V4 Accuracies

Figure 6. Average of the three subjects for each ROI.

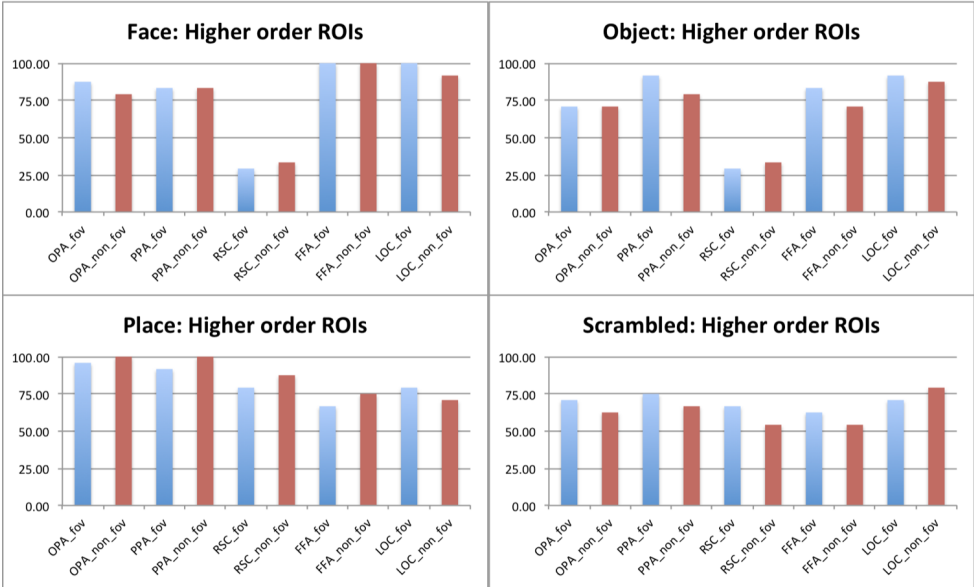
3.4 Higher Level Visual Region Accuracies

Figure 7. Average of the three subjects for each ROI.

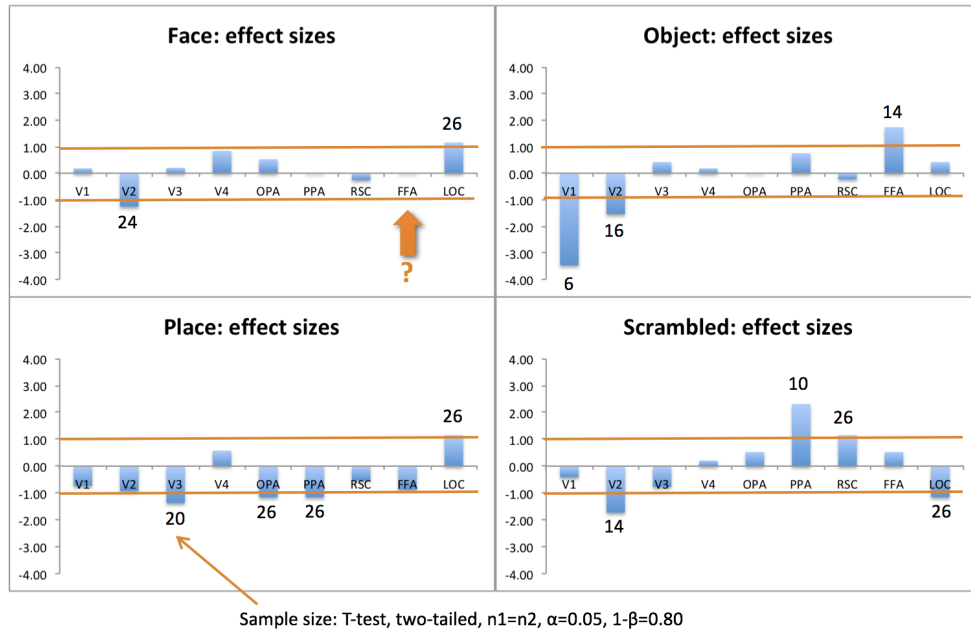
4 Analysis



title



title



title

4.1 Effect Size

Calculations of effect sizes for each category, in each ROI, when comparing foveal to non-foveal accuracy.

Note that non-foveal is **subtracted** from foveal, so positive values mean that the foveal subregions were more accurate in categorizing.

Figure 8. Bold arrow in Face: effect sizes: The FFA had 100% accuracy in all three subjects for face and as such an effect size could not be calculated.

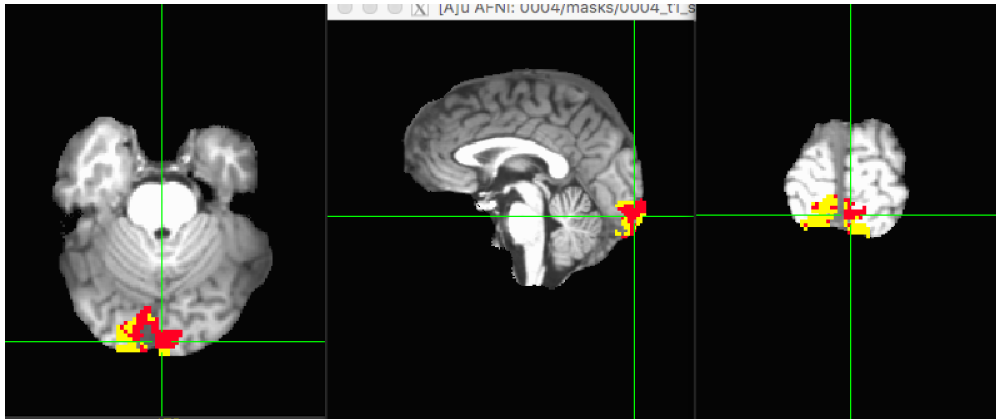
4.2 Spatial Arrangement

We were curious to see whether there might be a clear difference in the arrangement of the ROIs in terms of their receptivity at different eccentricities.

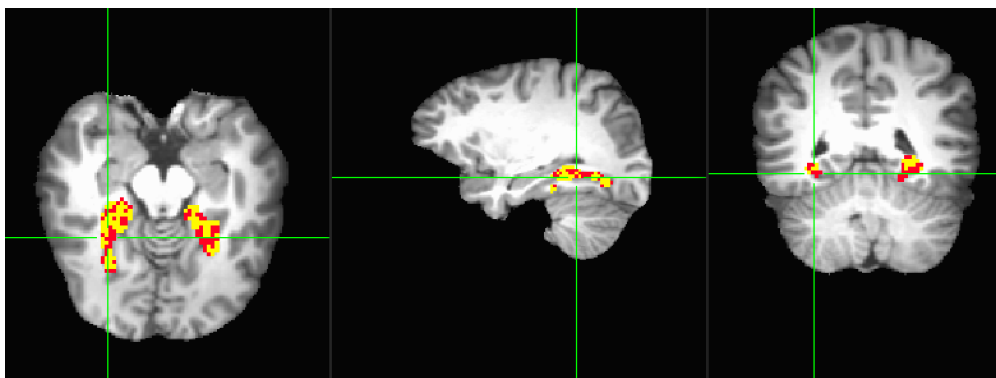
To look at this in a very rough way we loaded the filters for the foveal and peripheral ROIs into AFNI after running the following code in AFNI to combine but separately color the two eccentricities:

```
3dcalc -a 0004_OPA_fov_mask+orig -b 0004_OPA_non_fov_mask+orig -expr 'a+2*b' -prefix 0004_OPA_bo
```

This use of 3dcalc employed the simple math $a + 2*b$ to give the foveal mask a value of 1 and the non-foveal mask a value of 2 for a given ROI. The thought going in was that the lower visual areas (V1-V4) might be more contingently arranged in terms of eccentricity.



title



title

Looking below we found this to be somewhat the case in, for example, V1 and PPA, just based on visual inspection (subject 4).

V1

Figure 9. V1 for subject 4. The two colors represent foveal and peripheral pRFs.

PPA

Figure 10. PPA for subject 4. The two colors represent foveal and peripheral pRFs. Additional ROI images for subject 4 can be found in the data folder

However, we also looked at cluster number for each region to see if that gave us more analytically robust information. The idea being that if we set the cluster size to 2 then we would be able to see how many separate clustered regions there were at each eccentricity level. If a given eccentricity was completely contiguous then that would give a cluster number of just 2.

Below are screenshots of the cluster analysis tool for V1 foveal and non-foveal masks (subject 4).

V1 Foveal

Figure 11. Cluster information for V1 foveal. There are a total of 15 clusters (each consisting of 2 or more voxels). There were also 24 single voxels (unclustered), so a total of 15 clusters + 24 single voxels for region V1 foveal.

V1 Non-Foveal

Figure 12. Cluster information for V1 peripheral. There are a total of 8 clusters and 34 single voxels.

All Regions, Cluster Number

Using this technique we can look at the remaining ROIs (Subject 4).

Note that *Clusters + Singles* refers to the number of clusters with two or more voxels and the number of additional individual voxels.

```
In [14]: df2 = pd.read_csv('data/clusters.csv')

        table = ff.create_table(df2)

        plotly.offline.iplot(table)
```

Based on a quick look at the cluster numbers for each region it is not immediately clear that the lower regions are *more* contiguous than the higher level regions.

It could also be the case that the lower regions just look more contiguous to the eye when sliced in AFNI as they are less *complexly* contiguous than the higher regions.

5 Discussion

The goal of this exercise was to investigate how well pRFs of differing eccentricity might, on their own, serve to predict stimulus category. Our initial hypotheses were that categories such as faces and objects might be more accurately categorized by foveal regions, while places/scenes might benefit from peripheral vision, and therefore peripheral pRFs might be more accurate categorizers. While we did not find any significant evidence to support our hypotheses, there were

Region	Clusters + Single
V1_fov	15 + 24
V1_non_fov	8 + 34
V2_fov	28 + 36
V2_non_fov	16 + 33
V3_fov	13 + 29
V3_non_fov	16 + 39
V4_fov	8 + 16
V4_non_fov	17 + 49
FFA_fov	3 + 2
FFA_non_fov	9 + 13
LOC_fov	6 + 11
LOC_non_fov	11 + 40
OPA_fov	4 + 4
OPA_non_fov	10 + 17
PPA_fov	12 + 5
PPA_non_fov	19 + 25
RSC_fov	9 + 10

title

trends that aligned with our premise. For example, in general, peripheral regions were more accurate at classifying scenes, while for object and faces, the foveal pRFs of higher visual regions were slightly more predictive.

That said, our study suffered from two major problems.

Not enough subjects

With only three subjects, power and significance were bound to be hard to come by.

Too narrow a field of view

Due to the nature of the experimental setup, with the subject in a scanner looking at a screen a number of meters away, the maximum angle at which any stimuli could be viewed was well under 10 degrees. This is far below the true peripheral field of view for humans, and ignores large regions of visual receptivity. In order to truly measure the difference between foveal and peripheral eccentricities, and to do so in a more fine grained manner, it would be very helpful to have the stimuli cover a much larger range of the field of view.

6 References

Dumoulin, S. O., & Wandell, B. A. (2008). Population receptive field estimates in human visual cortex. *NeuroImage*, 39(2), 647–660.

Fairchild, M. D. (2013). *Color Appearance Models*. John Wiley & Sons.

Fovea centralis. (n.d.). In *Wikipedia*. Retrieved April 7, 2018, from https://en.wikipedia.org/wiki/Fovea_centralis

Peripheral vision. (n.d.). In *Wikipedia*. Retrieved April 7, 2018, from https://en.wikipedia.org/wiki/Peripheral_vision

Smith, A. T., Singh, K. D., Williams, A. L., & Greenlee, M. W. (2001). Estimating receptive field size from fMRI data in human striate and extrastriate visual cortex. *Cerebral Cortex*, 11(12), 1182–1190.

Victor, J. D., Purpura, K., Katz, E., & Mao, B. (1994). Population encoding of spatial frequency, orientation, and color in macaque V1. *Journal of Neurophysiology*, 72(5), 2151–2166.