# Predicting Image Categories Using Brain Decoding

Charles Akin-David, Minymoh Anelone, and Aarush Selvan
{aakindav, aselvan, manelone}@stanford.edu

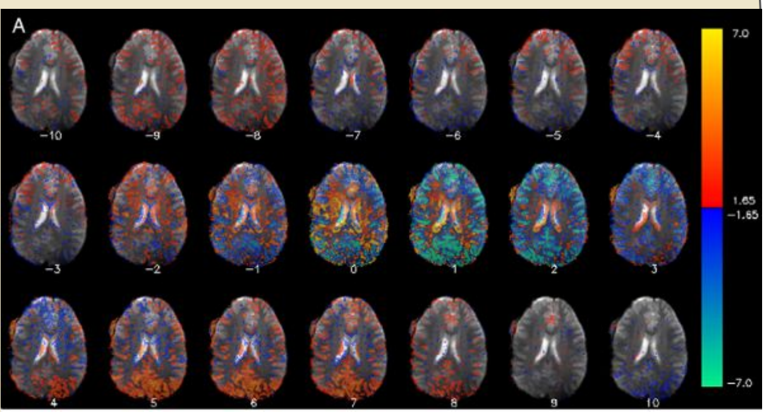
## Introduction

Question: How well can we decode human visual object recognition directly from the fMRI brain data.

Idea: Perform an experiment where subjects are shown a variety of images while their brain responses are recorded. Take the fMRI brain data and apply machine learning to accurately predict what images categories the subjects were being shown.

Goal: To learn about how the visual system of the brain chooses image classifications by decoding brain fMRI stimuli from different parts of the brain.

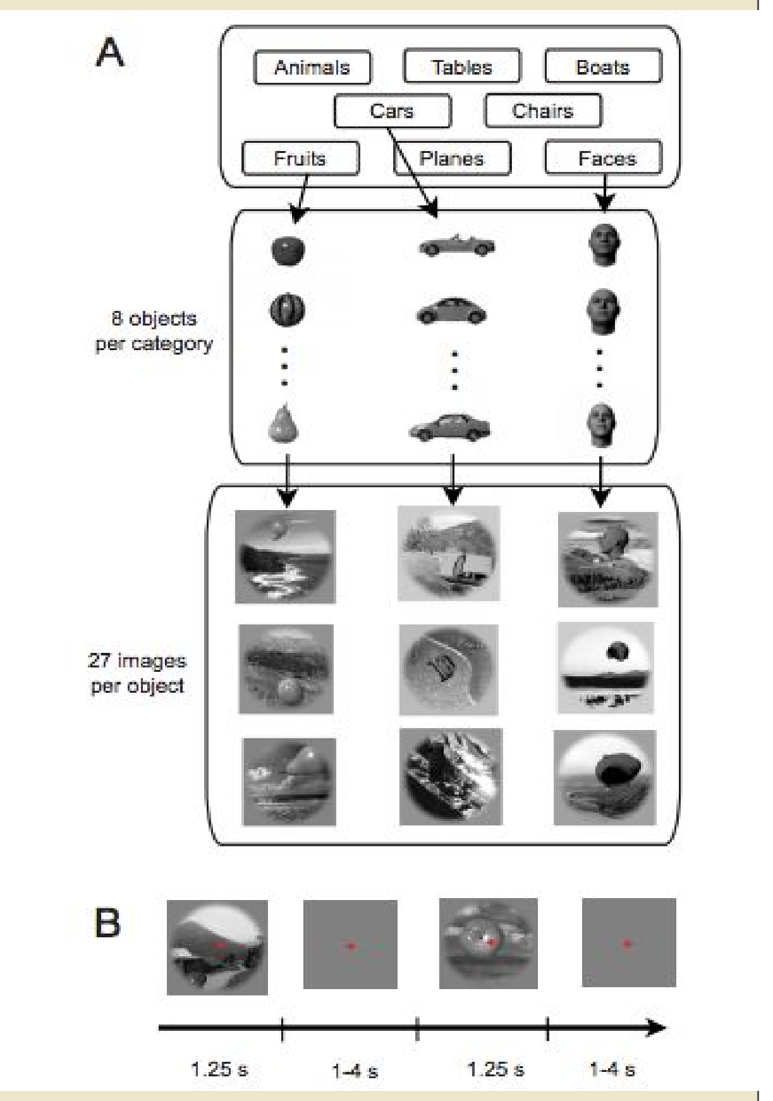
## Background



What is fMRI?
When neurons fire, the require energy and so more blood flows to that region of the brain.

fMRI measures this Blood-Oxygenation Level Dependency (BOLD signal)

Since this is a measure of a secondary effect of neuron firing - has lower spatial resolution and is noisy - hence opportunity for applying machine learning!



How the Data was Gathered
Each subject was presented 1785 gray-scale images of objects a median of six times across multiple sessions to each subject. Objects were drawn from 8 categories (animals, tables, boats, cars, chairs, fruits, planes, and faces) containing 8 exemplars. Each object was shown from 27 or 28 different viewpoints against a random natural background (circular vignette, radius 8◦ centered on fixation) to increase object recognition difficulty.

Whenever a frame was shown the brain's response was recorded as a stimulus. The results of the fMRI experiment were shown in a 3-D image that is constructed in units called voxels. Each voxel represents a tiny cube (3mm^3) of brain tissue which can represent about a million or so brain cells.

## Prediction

Motivation - Being able to predict what a person was looking at based solely off data from their brain responses.

Inputs - 'voxel' x 'stimuli' fMRI matrices per brain region

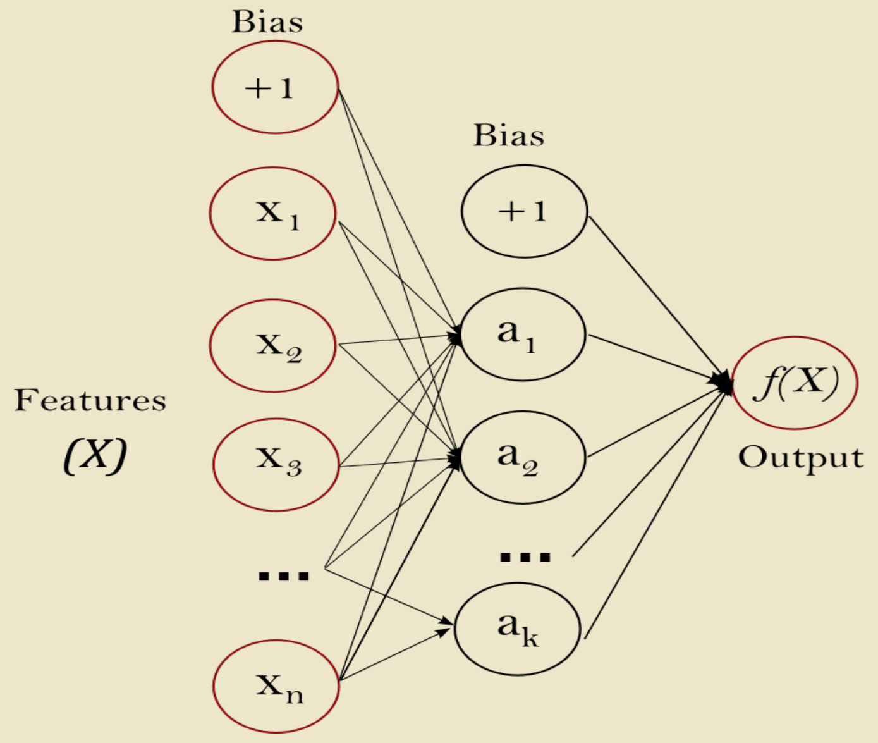
Outputs - predicted image categories

## Model & Features

Features: 3224 voxels representing 12 areas of the brain.
- All features come from raw_input data.
- No derived features since correlations between the raw_input features were unknown

Models:
Multi-layered Perceptron trains using Backpropagation (Neural Network)
- Adam Solver
- SGD with Nesterov's Momentum

Logistic Regression



Parameters for Adam:
- Activation: relu
- alpha: 1e-05
- beta1: 0.9
- beta2: 0.999
- epsilon: 1e-08
- initial learning rate: 0.001
- hidden layers: 180

Adam Update (simplified):

```
m = beta1*m + (1-beta1)*dx
v = beta2*v + (1-beta2)*(dx**2)
x += - learning_rate * m / (np.sqrt(v) + eps)
```

where m and v are initialized to be zero-vectors

Nesterov's Momentum (simplified):

$$v_t = \mu_{t-1}v_{t-1} - \epsilon_{t-1}\nabla f(\theta_{t-1} + \mu_{t-1}v_{t-1})$$
$$\theta_t = \theta_{t-1} + v_t$$

where θt are the model parameters, vt the velocity, $\mu_t \in [0, 1]$ the momentum (decay) coefficient and $\epsilon_t > 0$ the learning rate at iteration t, f(θ) is the objective function and $\nabla f(\theta')$ is a shorthand notation for the gradient: $\partial f(\theta)/\partial\theta \,|_{\theta=\theta'}$

## Data

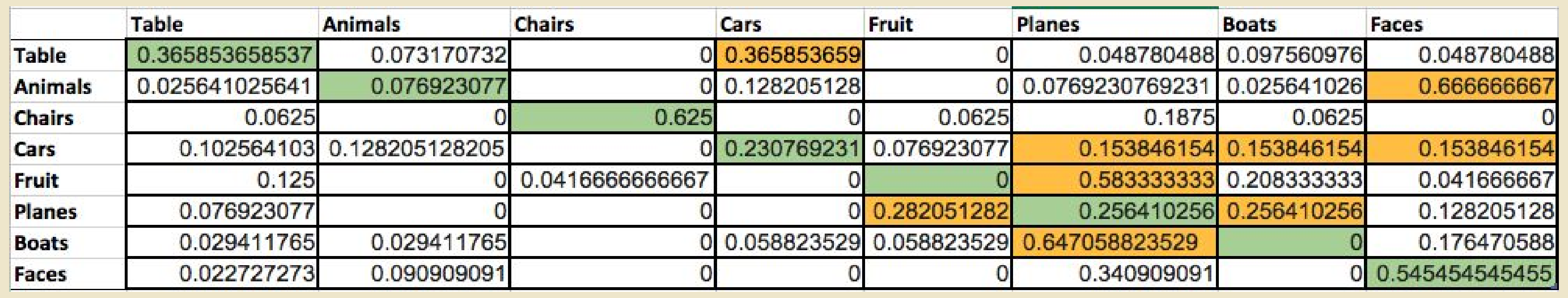
**1 voxel** - 3x3x3mm scan (1 million neurons). For each voxel, recorded **BOLD** response.

**Columns** - each voxel in region. **Rows** - each stimulus presented (**1365** grayscale images, **8** categories)

**Visual stream** split into **12** regions - **V1, V2, V3, V3a, V4, FFA, LO1, LO2, LOC, PPA, OFA, TOS**.

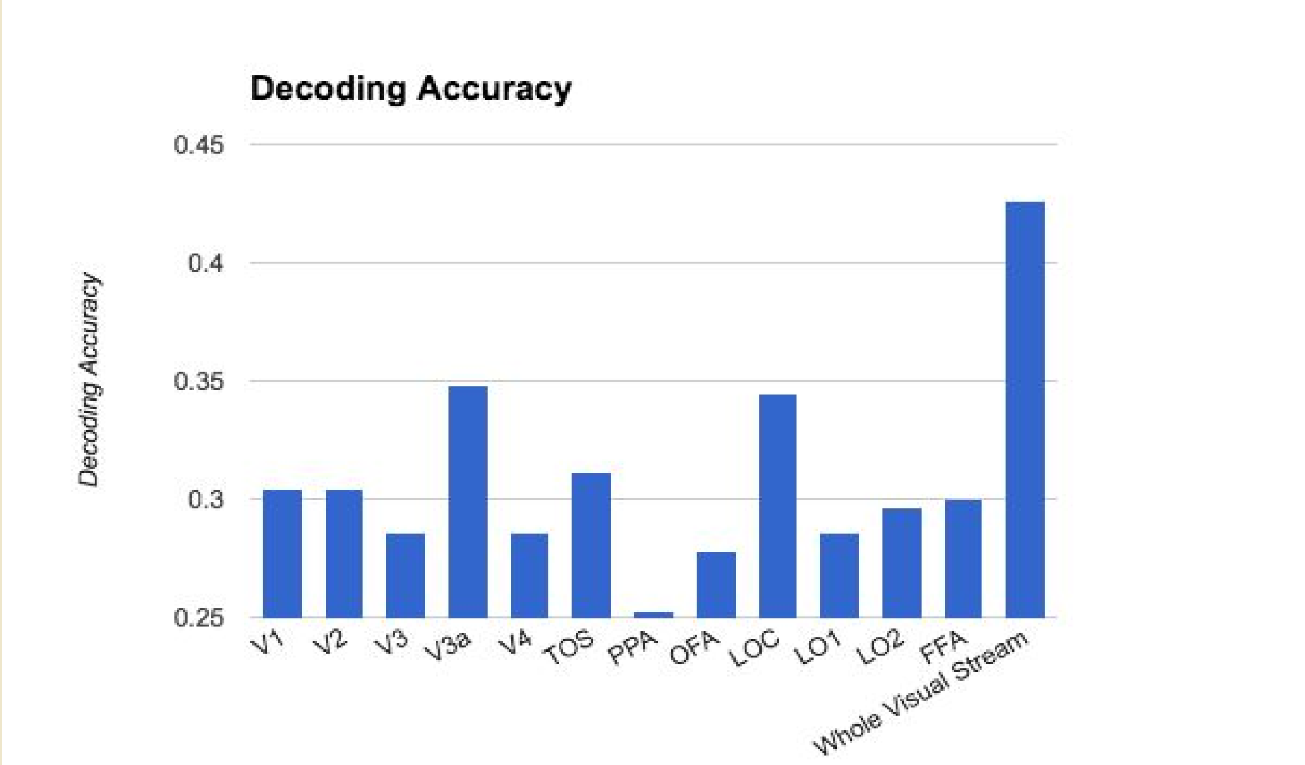
**Data Matrices**: **V1** - 1365x379, **V2** - 1365x601, **V3** - 1365x555, **V3a** - 1365x148, **V4** - 1365x350, **FFA** - 1365x211, **LO1** - 1365x118, **LO2** - 1365x114, **LOC** - 1365x460, **PPA** - 1365x97, **OFA** - 1365x125, **TOS** - 1365x66. **Whole visual stream** - 1365x3224

## Results



| | Table | Animals | Chairs | Cars | Fruit | Planes | Boats | Faces |
|---|---|---|---|---|---|---|---|---|
| Table | 0.365853658537 | 0.073170732 | 0 | 0.365853659 | 0 | 0.048780488 | 0.097560976 | 0.048780488 |
| Animals | 0.025641025641 | 0.076923077 | 0 | 0.128205128 | 0 | 0.0769230769231 | 0.025641026 | 0.666666667 |
| Chairs | 0.0625 | 0 | 0.625 | 0 | 0.0625 | 0.1875 | 0.0625 | 0 |
| Cars | 0.102564103 | 0.128205128205 | 0 | 0.230769231 | 0.076923077 | 0.153846154 | 0.153846154 | 0.153846154 |
| Fruit | 0.125 | 0 | 0.0416666666667 | 0 | 0 | 0.583333333 | 0.208333333 | 0.041666667 |
| Planes | 0.076923077 | 0 | 0 | 0 | 0.282051282 | 0.256410256 | 0.256410256 | 0.128205128 |
| Boats | 0.029411765 | 0.029411765 | 0 | 0.058823529 | 0.058823529 | 0.647058823529 | 0 | 0.176470588 |
| Faces | 0.022727273 | 0.090909091 | 0 | 0 | 0 | 0.340909091 | 0 | 0.545454545455 |

**Confusion Matrix**
The confusion matrix above was built by training the model on a section of the entire dataset, and then only testing it on each specific stimuli category. It helps us understand amongst what categories the model is getting confused. The green boxes illustrate the prediction accuracy - in a perfect model these values would be 1, and all the rest would be 0. The orange boxes illustrate the incorrect stimuli category that the model most often confuses as the actual stimuli category.



**Decoding accuracy by brain region**
Here we trained and tested our model on a region by region basis in order to see which brain regions had the greatest impact in object recognition. We also compared this to testing and training the model on the entire visual stream.

**Model Accuracies**
MLP using Adam's Solver: 43%
MLP using SGD with Nesterov's Momentum: 40%

## Discussion

- We can see that certain stimuli categories are very poorly decoded compared to others - e.g. faces are decoded quite well but fruit is decoded very poorly. We can also see for instance that animals are more often confused for faces than not. Perhaps, since animals and faces are the only two animate categories, the two stimuli categories trigger similar neural responses - explaining why the model gets so confused.
- Similarly the model often confuses cars with planes and boats - all these objects fall under the broader category of 'transportation' and this lends strength to the idea that the brain interprets these objects similar fashion.
- Finally, we can see from the bar graph that while some brain regions are better at decoding object categories than others, decoding from the entire visual stream is better than focusing on one specific region - this tells us that there is no specific region in the visual cortex dedicated to static object recognition.

## Future Work

- Hyperparameter Optimization
  With more time and resources, we would optimize our parameters further following Bergstra and Bengio's *Random Search for Hyperparameters Optimization*
- Derived Features
  We also could have worked further with Seibert et al. to see how we could develop some derived features for our model
- Stronger Neural Networks
  We also would have been able to do more research on different variations of neural networks

## References

- Bengio, Yoshua, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. "Advances in Optimizing Recurrent Networks." Cornell University Library, 14 Dec. 2012. Web. 13 Dec. 2016.
- Kingma, Diederik, and Jimmy Ba. "A Method for Stochastic Optimization." Cornell University Library, 23 July 2015. Web. 13 Dec. 2016. <http://arxiv.org/abs/1412.6980>.
- Scikit-learn Developers. "Multi-layer Perceptron." 1.17. Neural Network Models (supervised) — Scikit-learn 0.18.1 Documentation. N.p., n.d. Web. 13 Dec. 2016.
- Seibert, Darren, Daniel Yamins, Diego Ardila1, Ha Hong, James J. DiCarlo, and Justin L. Gardner. "A Performance-optimized Model of Neural Responses across the Ventral Visual Stream." (2016): n. pag. Web. 13 Dec. 2016. <http://biorxiv.org/content/biorxiv/early/2016/01/12/036475.full.pdf>
- "CS231n Convolutional Neural Networks for Visual Recognition." CS231n Convolutional Neural Networks for Visual Recognition. Stanford University, n.d. Web. 13 Dec. 2016. <http://cs231n.github.io/neural-networks-3/>.