# Methods

## Subjects

Five adult males, ages 24--57, with normal or corrected-to-normal vision, participated in the experiments. All subjects participated in two fMRI sessions and a third session to acquire a high-resolution structural anatomy. Informed consent was obtained from all subjects.

## Stimulus

To facilitate designing our virtual environment, the Unreal Developer's Kit developed by Epic Games, Inc. was used (available at <http://www.unrealengine.com/udk>). This development kit is available free of charge for non-commercial applications and uses the same rendering and game engine found in many current and popular video games.

One of the long-term goals of this research is to understand post-traumatic stress disorder. Therefore, we created a virtual town intended to mirror the kinds of real-world settings encountered by many currently deployed military forces (Fig. \ref{fig:stimulus}). Virtual characters representing friendly forces and hostile combatants were situated at a variety of locations throughout the virtual town.

The stimulus was rendered in real-time from the point of view of a camera moving at eye level through the town.

During the stimulus presentation, the camera followed a preset path through the town. The stimulus employed a blocked approach. The camera would be steadily moving for 15 seconds, followed by 15 seconds in which it stopped to view virtual characters at predefined locations. There were four different predefined character-viewing locations, and the number of characters present at each of these locations varied from one to six (Figs. \ref{fig:stimulus-five-soldiers} and \ref{fig:stimulus-three-insurgents}). While the camera was stopped to view the characters, the view slowly panned and rotated and the characters engaged in animated movement sequences, so that the scene was never static. A passive viewing paradigm was employed for this stimulus: there was no fixation dot and no task during the presentation.

Scanning sessions included five to six runs that were six minutes in duration.

Each run contained 12 alternations between moving through the virtual environment and character presentations. Since there were four predefined character viewing locations, the camera made three laps around the town during a single run. While the number of characters presented in each block varied from one to six, the stimulus was controlled such that a presentation with a particular number of characters appeared twice in each run. The ordering of the presentations was also controlled such that presentations with the same number of characters were located at different viewing points. The order of the presentations was generated randomly given the previous constraints, but the same ordering was used for each subject.

A single type of character (friend or foe) was presented in each run, and character type was alternated from run to run.

## MRI protocols

Imaging was performed on a GE Signa Excite HD scanner using the product eight-channel head coil. Whole-brain image volumes were collected using a custom GRAPPA EPI sequence \citep{Griswold2002}. Sequence parameters were g-factor = 2, TE = 25 ms, TR = 2.5 s, and 2.5-mm cubic voxels across a 200 mm field-of-view.

The slice prescription included 40 slices oriented along the AC-PC axis. A high-order shim was performed to improve field homogeneity.

A set of T1-weighted structural images was obtained on the same prescription before the functional acquisition runs using a three-dimensional (3D) fast RF-spoiled GRASS (fSPGR) sequence. These anatomical images were then used to align the functional data to a structural 3D reference volume, which was acquired for each subject in a separate session. The structural reference volume was T1-weighted with good gray-white contrast and was acquired using a 3D inversion-prepared fSPGR sequence (minimum TE and TR, TI = 450 ms, 15$^\circ$ flip angle, isometric voxel size of 0.7 mm, 2 excitations, $\sim$28-minute duration).

## Preprocessing

Preprocessing of the fMRI data was performed using the mrVista software package (available at \url{http://vistalab.stanford.edu/}). The first 15 seconds of data were discarded to reduce transient effects. Within-scan motion was then estimated using a robust intensity-based scheme \citep{Nestares2000}. Between-run motion was corrected using the same scheme, this time applied to the temporal average intensity of the entire scan. The first run of the session was used as the reference.

Because the goal is to learn associations between patterns of activation in the brain and stimulus presentation, it is important that the activation is temporally aligned with the stimulus. Therefore, a Wiener filter deconvolution \citep{Poor1980} was applied using a generic difference-of-gamma hemodynamic response function (HRF; \cite{Glover1999}) as the kernel to the recorded BOLD signal. Mostly, the deconvolution served to shift the peak response in time so that it was aligned with its associated stimulus, but it also provided some amount of noise reduction.

In Wiener filter deconvolution, the deconvolution kernel $g(t)$ is most easily expressed in the Fourier domain as

\begin{equation}

g(t) \xrightarrow{\mathcal{F}} \frac{H^{\*}(f)}{\left|H(f)^{2}\right| + \frac{\left| N(f) \right|}{\left| X(f) \right|}},

\end{equation}

where $h(t)$ is the blurring kernel, $x(t)$ is the signal of interest, and $n(t)$ is additive noise; capital letters denote the Fourier transform of each quantity, e.g., $h(t) \xrightarrow{\mathcal{F}} H(f)$.

In fMRI, $h(t)$ is the hemodynamic response function and $x(t)$ is the neural response. Calculating $g(t)$ requires estimates of the power spectral density of the signal of interest as well as the noise. However, the noise $n(t)$ corresponds not only to scanner noise but other nuisance factors such as pulse and respiration.

These factors make modeling the noise, and its power spectral density, very difficult.

Therefore, $\frac{\left| N(f) \right|}{\left| X(f) \right|}$ was set to $1.0$, a compromise value that provided a satisfactory combination of temporal alignment and noise reduction for all subjects.

The high-resolution reference anatomies were segmented using the Freesurfer image analysis suite (\url{http://surfer.nmr.mgh.harvard.edu/}) to create approximate parcellations of the gray matter in each subject, as well as a surface model useful for visualization of the results.

## Dimensionality reduction

When performing MVPA, each voxel can be thought of as corresponding to a separate dimension of a very high-dimensional space. For this experiment's acquisition parameters, the number of voxels/dimensions was $80 \times 80 \times 40 = 256,000$. Because the number of examples necessary to train a classifier increases rapidly with the number of dimensions, it was useful to reduce this dimensionality. This reduction not only speeded up both training and classification but also improved the performance of the resulting classifiers.

Principal component analysis (PCA; \cite{Hotelling1933}) is a common tool for dimensionality reduction. However, PCA only selects the orthogonal dimensions with the highest variance, which are likely to include physiological nuisance and such approaches are therefore not well suited for fMRI analysis. Univariate statistical methods, as typically used in fMRI analysis, can also be good candidates for selecting voxels \citep{Norman2006,Pereira2009}. While effective, these methods were not designed for feature selection and many of them make a number of assumptions about the data such as a Gaussian statistical distribution.

A novel method of dimensionality reduction was developed in this research. This method, called harmonic analysis, selects voxels that are driven most strongly by the 30-second duty cycle of the block design. This method is similar to other univariate statistics, but the only assumption is that the BOLD response is linear with respect to the stimulus. The primary advantage is that it can detect voxels that covary with the stimulus regardless of their detailed temporal relationship. Thus, it includes voxels that respond positively to either the character presentation or motional phases of the stimulus alternation, or to more complex patterns such as a brief strong response to both phases. All repetitive responses that follow the period of the stimulus alternation are included by this method, minimizing any bias in the dimension reduction.

Harmonic analysis takes advantage of the fact that the response of any linear system to a blocked alternation at frequency $f$ will contain power only at $f$ and its harmonics. Under a linear response assumption, we can therefore form an unbiased estimate of the response power by summing the power at these frequencies.

Let $y(t)$ be the recorded discrete time series at some voxel. Then let $Y(f)$ be the discrete Fourier transform of $y(t)$. The fractional harmonic power of that time series is defined as

\begin{equation}

P\_h = \frac{\sum\_{i = 1}^{M}{\left|Y(i \cdot N)\right|^{2}}}{\sum\_{f}{\left|Y(f)\right|^{2}}},

\end{equation}

where $P\_h$ is the fractional harmonic power, $M$ is the number of harmonics, and $N$ is the frequency of interest, in our case the period of the block alternations.

Because the BOLD response has a predominantly low-pass temporal frequency response, we chose $M = 4$ as sufficient. Using $P\_h$, a particular number, $N = 2000$, voxels with the greatest power are selected.

This harmonic-power selection was based on the alternation between characters present and characters absent, without regard to the number of characters presented. Therefore, classifier accuracy estimates will only be presented for character count, and not for the presence or absence of characters, in order to avoid overlap between dimension-reduction and classification criteria that would result in inflated classifier performance estimates \citep{Pereira2009}. However, the classifiers were also trained to distinguish between time points with and without characters as a check, and their performance was consistently above 95\%, confirming that the machine-learning algorithms were working correctly.

## Classification

Using the time series from the voxels selected by the harmonic analysis, a one-versus-one multi-class linear support vector machine (SVM) \citep{Cortes1995,Weston1999}, a feedforward neural network (NN) \citep{Hornik1989,Hagan1994}, a Gaussian naive Bayes classifier (GNB) \citep{Duda1973}, and a k-nearest neighbor classifier (KNN) \citep{Cover1967} were trained to identify the number of characters presented in each scene, regardless of character type (friend or foe). To maximize temporal resolution, each 2.5-second frame (time point) was treated as a separate example, rather than averaging across the 15-second blocks. Similar methods were also used to classify scenes as friends or foes, but these classifiers did not perform significantly above chance and will not be mentioned further.

The performance of machine learning algorithms is generally defined to be the expected accuracy of the classifier on previously unseen examples \citep{Bishop2006}. In practice, this measure can only be estimated.

A typical approach is to split the available examples into training and test sets.

The classifier is first trained on the training set, and its performance on the test set is then taken as the estimate. The process of splitting all of the available examples into training and test sets is performed multiple ($n$; typically 10) times to reduce the variance of the performance estimate. This procedure is known as $n$-fold cross-validation \citep{Kohavi1995}. Previous studies have discussed issues with optimistically biased performance estimates due to temporal correlations that violate independence assumptions between training and test set samples \citep{Pereira2009}. The slow speed of the fMRI hemodynamic response clearly introduces temporal correlations on time scales less than 10 seconds. Therefore, training and test sets were constructed by randomly drawing 15 second blocks rather than individual frames after which 10-fold cross-validation was performed.

Furthermore, the null distribution for the cross-validated performance estimates was generated by randomly permuting the labels on the examples 2000 times and repeating the training and cross-validation procedure. That is, the distribution of performance estimates was generated under the assumption that the labels and data were independent. Using this distribution, $p$ values were calculated for the performance estimates. The high-performance computing resources of the Texas Advanced Computing Center (TACC) at The University of Texas at Austin were utilized to perform this computation.

## Sensitivity Analysis

MVPA can tell us whether the time-series data from a subset of human brain voxels can be used to discriminate character number from the stimuli. However, these results do not show which voxels in the large group were actually important for that discrimination. This information is important for localizing functions in the brain.

One existing technique is to train machine learning classifiers on small localized areas in the brain and use their performance as a measure of the strength of the function in question in that area; this is known as the ``searchlight'' technique \citep{Kriegeskorte2006}. While this technique is effective for brain functions already known to be highly localized, the results are less clear when the function is sparsely distributed over the brain. A single region may not contain enough information for accurate classification or a region may only contain relavant information only when considered in conjunction with a spatially disparate region.

To overcome this limitation, a sensitivity analysis technique originally used to minimize the input data dimension of feedforward neural networks \citep{Zurada1994} was adapted to display the spatially distributed set of voxels that are most important for identifying the classes. Specifically, the sensitivity, or magnitude of change, of the neural network output was calculated with respect to a change in each voxel. Let $\mathbf{o}$ be the vector of outputs and $\mathbf{x}$ be the vector of inputs. Then the sensitivity of output $k$ to input $i$ is defined by

\begin{equation}

S\_{ki} = \frac{\delta o\_{k}}{\delta x\_{i}},

\end{equation}

which is the partial derivative of the output with respect to the input. Let $\mathbf{w}$ be the weight matrix from the hidden layer to the output layer and $\mathbf{v}$ be the weight matrix from the input layer to the hidden layer. Then the partial derivative can be expressed as

\begin{equation}

\frac{\delta o\_{k}}{\delta x\_{i}} = o'\_{k} \sum^{J}\_{j=1}{w\_{kj}y'\_{j}v\_{ji}},

\end{equation}

where $J$ is the total number of hidden units in that layer of the neural network, $o'\_{k}$ is the value of the derivative of the activation function at output $k$, and $y'\_{j}$ is the value of the derivative of the activation function at hidden neuron $j$.

Finally, the entire sensitivity matrix can be expressed in matrix notation as,

\begin{equation}

\mathbf{S} = \mathbf{O}' \times \mathbf{W} \times \mathbf{Y}' \times \mathbf{V}

\end{equation}

where

\begin{equation}

\mathbf{O}' = diag(o'\_{1},~o'\_{2},~\cdots,~o'\_{K})

\end{equation}

\begin{equation}

\mathbf{Y}' = diag(y'\_{1},~y'\_{2},~\cdots,~y'\_{K}).

\end{equation}

However, because the transfer functions are nonlinear they can only be evaluated for specific input values. Therefore, we calculated the average sensitivity matrix across all input vectors,

\begin{equation}

\mathbf{S}\_{avg} = \sqrt{ \frac{ \sum\_{n = 1}^{N}{ \left( \mathbf{S}\_{n}\right)^{2} } }{N} },

\label{eqn:sensitivity}

\end{equation}

where $N$ is the number of input vectors. The magnitude is squared so that the effects of both positive and negative sensitivities are included in the average. Eq. \ref{eqn:sensitivity} gives a sensitivity value for each voxel with respect to every output, whereas it is useful to have a measure of the sensitivity of a voxel with respect to any output. Such a metric can be defined by taking the maximum sensitivity of each voxel across all outputs, i.e.

\begin{equation}

\Phi\_{i} = \max\_{k=1 \dots K}{S\_{ki,~avg}}.

\end{equation}

This sensitivity can now be projected back into the volume anatomy space to create a map that reflects the relative incremental importance of each voxel's response to the classification decision. In theory, a similar technique could be introduced for any differentiable classifier such as the SVM. However, only the sensitivity analysis applied to neural networks will be presented and discussed as it has been well studied in the existing literature \citep{Zurada1994}.

We explored the relationship between sensitivity and classifier performance by training the classifier on only a subset of the input voxels as determined by a minimum-sensitivity threshold. The threshold was gradually increased until classifier performance was significantly degraded.

The Freesurfer image analysis suite was employed to construct cortical surfaces for each of the subjects using high-resolution anatomy volumes.

Ten anatomical labels were constructed on each surface based on FreeSurfer's automatically generated labels. Each subject's volume sensitivity map was then projected onto their cortical surface and blurred along the surface using a 5 mm full-width half-maximum (FWHM) Gaussian kernel. Group averaged sensitivity maps were also formed using both affine and non-linear registration algorithms.

However, the quality of the registration between our subject brains was not satisfactory. In particular, group-average maps tended to mislocalize temporal-lobe activity onto the parietal lobe. Therefore, sensitivity maps will be only presented on individual subject brain surfaces. However, the total sensitivity in each anatomical label was averaged across subjects in order to examine group-average sensitivity without requiring registration.

## GLM

GLM analysis was also performed on the data to compare classification performance and the spatial distribution of significant voxels with the MVPA sensitivity analysis maps. A linear activation model was constructed using an explanatory variable for each character count. Processing of fMRI data was carried out using FEAT (FMRI Expert Analysis Tool) Version 5.98, part of FSL (FMRIB's Software Library, \url{www.fmrib.ox.ac.uk/fsl}). Z (Gaussianized T/F) statistic images were thresholded using clusters determined by $Z > 2.3$ and a (corrected) cluster significant threshold of $P = 0.05$ \citep{Worsley2001}. Clusters in these images indicate regions where the magnitude of activation is linearly correlated with the number of characters presented. The thresholded Z statistic images were projected onto the same Freesurfer generated surfaces as the sensitivity analysis.

For each voxel above the significance threshold, this model was used to construct a simple linear estimator for the number of characters presented by fitting a line to the parameter estimates of each explanatory variable versus their corresponding character counts. A continuous estimate of character count was obtained for each frame using the linear fit and the current parameter value. This character count estimate was clipped between one and six, averaged across significant voxels, and then rounded to the nearest whole number. The accuracy of this model was calculated for comparison with the machine-learning techniques.