Master thesis

Encoding BOLD responses to video stimuli with K-means-based hierarchical representation learning

submitted by
Katja Müller
Matrikel: 350831

Professor: Prof. Dr. Klaus-Robert Müller, TU Berlin
Supervisor: Dr. Grégoire Montavon, TU Berlin

Technical University of Berlin, Department of Software Engineering and Theoretical Computer Science
Machine Learning Group
Berlin, August 2015

Contents

Authorship

I hereby declare that I have written this thesis without inadmissible help from third parties and that I have not used any other sources or aids than those specified.

Berlin – September 22, 2015

____

Signature

Acknowledgements

This thesis was written at the at the Chair for Machine Learning at the Institute of Software Engineering and Theoretical Computer Science of the Technical University of Berlin. I would like to express my gratitude to my supervisor Dr. Grégoire Montavon and to Prof. Dr. Klaus-Robert Müller for their guidance and for the opportunity to pursue a thesis about this topic in their group. My sincere thanks also goes to Dr. Shinji Nishimoto at the Center for Information and Neural Networks (CiNet) in Osaka, who invited me to work in this area in his group, introduced me to their methodology and allowed me to use both his dataset and their computing resources at NICT. Besides my advisors, I would like to thank Dr. Adam Coates, Dr. Felix Bießmann, Kishore Konda, Dr. Roland Memisevic and Dr. Carsten Bogler for their insightful comments and frequently instant help on modelling specifics, visualization and on best practices in fMRI analysis. Also I would like to thank my friends and family for their support and patience throughout the entire length of the thesis.

Abbreviations

BOLD	Blood-Oxygen-Level Dependent
fMRI	functional Magnetic Resonance Imaging
GFP	Gabor Filter Pyramid
HRF	Haemodynamic Response Function
RDE	Relevant Dimensionality Estimation
RDM	Representational Dissimilarity Matrix
RSA	Representational Similarity Analysis
ROI	Region Of Interest
TCM	Two-Component Model
A1	Primary Auditory Cortex
AC	Auditory Cortex
EBA	Extrastriate Body Area
FBA	Fusiform Body Area
FEF	Frontal Eye Fields
FFA	Fusiform Face Area
FO	Frontal Operculum
IFSFP	Inferior Frontal Sulcus Face Patch
IPS	Intraparietal Sulcus
LO	Lateral Occipital area
FO	Frontal Operculum
MTp	Middle Temporal area
OFA	Occipital Face Area
PPA	Parahippocampal Place Area
RSC	Retrosplenial Cortex
TOS	Transverse Occipital Sulcus
VO	V8 or Ventral Occipital area
pSTS	posterior Superior Temporal Sulcus

Abstract

Recently several publications have analysed whether deep neural networks can be used as an encoding model for brain activity in the hierarchically organized human visual system. These so-called deep encoding models are promising for understanding higher-level visual representations in the brain and for visual perception reconstruction, but have so far primarily been analysed for static image data. In this project we investigated encoding spatio-temporal stimulus data with a purely unsupervised deep encoding model based on a soft K-means hierarchy. The model consists of two learning routines, where the first routine is used only for the raw-pixel based first layer and the second routine is used repeatedly for all higher layers. Based on the acquired feature sets BOLD activity is predicted via ridge regression for evaluating the predictive power of the model. The linear model was verified by voxel-wise estimation of noise with Relevant Dimensionality Estimation. The learned representations were investigated in more detail by using Representional Dissimilarity Matrices. The features learned in the first layer are capable of predicting BOLD activity in early and few intermediate areas, however with lower predictive power than the hand-crafted state-of-the-art Motion Energy model from [Nishimoto et al., 2011]. In comparison to the first layer of the K-means model, its higher layers lead to better signal prediction in a subset of voxels in intermediate and higher level regions of interest, in particular in areas pSTS, IPS, MTp and EBA. In higher visual cortex areas however the K-means model is often outperformed by semantic `word2vec` based features similar to the category model in [Huth et al., 2012]. We conclude that while this completely unsupervised hierarchical learning routine is in principle capable of predicting voxel activity in the lower visual areas in response to spatio-temporal stimuli, it can only explain a few new voxels in intermediate and higher layer regions of interest.

1 Introduction

1.1 Background and rationale

Although a wide range of experimental results exists, computational models of the visual system – including V1 – are still speculative [Carandini et al., 2005]. While the visual features V1 responds to could often be shown to resemble Gabor functions [Hubel and Wiesel, 1959], it has been described as questionable whether these represent the complete feature spectrum covered by V1 or whether the experimental results fall victim to methodological biases [Olshausen and Field, 2005]. Since early studies about the cat's visual cortex it has also been known that feature detectors in the visual cortex do not exist in the brain a-priori, but are *learned* from the experienced visual environment during a critical period after birth [Hubel and Wiesel, 1970], [Blakemore and Cooper, 1970].

Tracing how the BOLD signal develops over the ventral and dorsal streams in reaction to specific visual stimuli is currently the only method of investigating human visual processing over the whole visual cortex. *Visual encoding models* describe how visual information about the world could be *represented* in the brain. [Nishimoto et al., 2011] showed that an encoding model based on *motion energy* in video data – using a hand-crafted dictionary of Gabor filters (a Gabor Filter Pyramid (GFP)) and extracting motion energy (as in [Adelson and Bergen, 1985]) from their responses – can explain the BOLD activity in early visual areas. However defining such a model requires experimentally verified knowledge about which type of feature detectors to expect. Due to this a hand-crafted approach for modelling representations is limited to experimental knowledge about early visual areas. In particular in V1 neurons have been experimentally verified to respond to edges [Hubel and Wiesel, 1959] and edge conjunctions [Ito and Komatsu, 2004]. However the representations in intermediate (such as V4, V7) and higher areas (such as FFA, EBA) remain unexplained with the Motion Energy model.

From [Huth et al., 2012] we know that activity in higher-order areas in reaction to natural video stimuli can be explained with a *category model* based on `WordNet` categories, forming a continuous semantic space over the cortex. However, this model relied on previously created manual video stimulus labels. Desirable would be a computational encoding model leading from simple low-level features to the continuous semantic categories that we saw in this publication. In principle, any evidence for a general learning algorithm for higher order representations would help understanding these transitions. There is reason to assume that such a model of visual processing would be reflected in the BOLD signal in a mostly hierarchical way [Felleman and Van Essen, 1991]. Refer to figure 1.1 for a comparison of the areas that can be predicted by the low-level motion-energy model from [Nishimoto et al., 2011] (*Red*) and the high-level category model from [Huth et al., 2012] (*Green*).

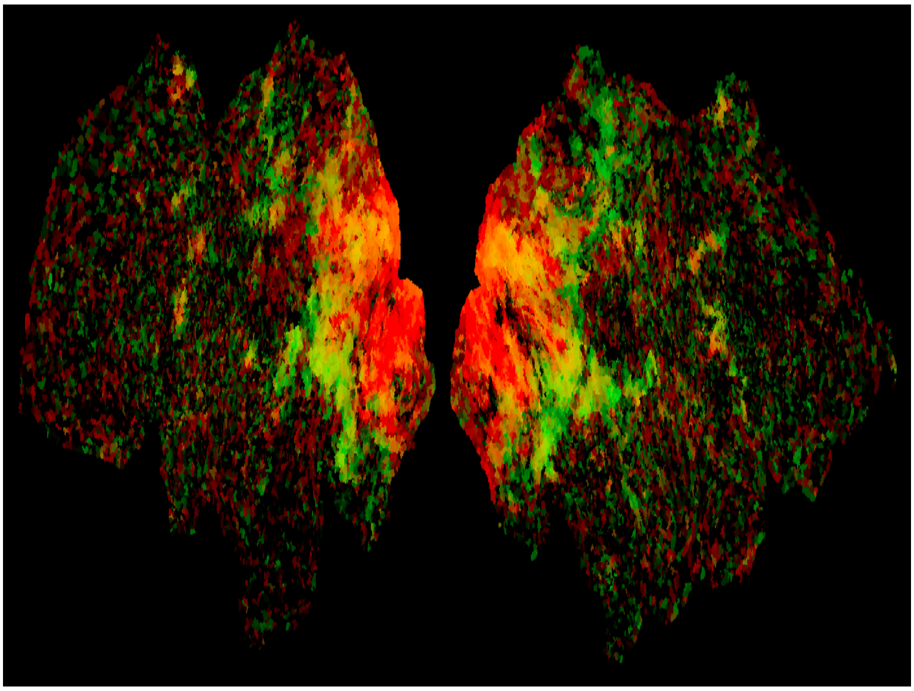


Figure 1.1: Correlations between acquired and predicted BOLD responses for subject SN. Red: Motion energy model, Green: Category model.

Due to the increasing computational resources, recently many hierarchical representation learning models for visual data could be investigated with higher amounts of training data, deep complex architectures and effective architectural optimizations by the machine learning community [Bengio et al., 2013], [LeCun et al., 2015]. These models are currently leading to promising results for predicting object types or categories on top of a hierarchy of learned low-level representations. These low-level representations usually include, but are not limited to the Gabor filter-like edge structures that had been studied intensely in V1 experiments.

While these deep neural networks are not intended to be meaningful models of biological visual processing, convolutional unsupervised learning strategies are indeed inspired by our knowledge about the visual cortex. It is worth investigating and it is being investigated whether a deep representation learning model that has been proven to work well for a specific domain (visual object recognition, audio) does reflect brain activity in response to the same domain. This idea – to investigate *deep encoding models* – is the general framework of this thesis project.

1.2 Related publications

1.2.1 Deep Encoding Models

The idea to encode processing stages in the brain with hierarchical representation learning models has already been explored a few years ago: An early example is [Gerven and Lange, 2010], who used a hierarchical conditional restricted Boltzmann machine to encode BOLD brain activity for characters taken from the MNIST database and also reconstructed the perception. However due to the current resurrection of neural networks – and due to the availability of both deep learning libraries and large training data sets suitable for studying encoding – the idea has recently been more enthusiastically explored by several groups, for instance [Güçlü and van Gerven, 2014], [Agrawal et al., 2014] and [Cadieu et al., 2014].

[Agrawal et al., 2014] and [Gerven, 2014] have used the pre-trained [Krizhevsky et al., 2012] ImageNet deep neural network (implemented in the `Caffe` framework) to encode static images. Their fMRI datasets had originally been published in [Kay et al., 2008] and [Naselaris et al., 2009] for studying hand-crafted encoding models. They both found that the voxel activity in response to the purely spatial stimuli covered different voxel regions, in the case of [Gerven, 2014] a feature gradient along the ventral pathway. [Gerven, 2014] have also derived receptive fields for voxels, and could show that different layers cover mutually exclusive voxel areas. [Agrawal et al., 2014] compared the same deep neural network against Fisher vectors. There has also been an evaluation for static images with a large selection of several object recognition models (both deep and shallow models) based on Representational Similarity Analysis (RSA) in [Khaligh-Razavi et al., 2014].

All of these publications have been working with *static spatial stimuli*. There still have been few encoding model publications working with *spatio-temporal* data. [Häusler et al., 2013] introduces a temporal restricted Boltzmann machine to encode videos, and derives a spike response model from them to study its properties. [Ramakrishnan et al., 2015] study two single-layer object recognition models from computer vision in response to short video sequences. To our knowledge there is no published attempt to use *hierarchically learned spatio-temporal representations* to encode BOLD brain activity in response to video signals in *large cortical areas.*

1.2.2 K-Means based deep encoding

A soft K-means model trained on whitened prototypes had been shown to lead to similar low-level representations and prediction results as more tuning-intensive deep neural networks in [Coates et al., 2011] and [Coates and Ng, 2012], compared to the 2011 state-of-the-art deep networks. Also, K-Means representations in lower levels have been shown to be largely equivalent to methods with solid information-theoretical foundation such as ICA if conditions such as input patch whitening are met [Vinnikov and Shalev-Shwartz, 2014].

Bag of Words models in general are a common basis for object recognition in computer vision applications. Furthermore, it is possible to build a hierarchical model based on them, as described in [Coates and Ng, 2012].

This thesis project studies whether a completely unsupervised hierarchical representation learning model based on soft K-means can encode spatio-temporal stimuli: After an initial convolutional layer as described in [Coates et al., 2011] it builds each successive layer using the process described in [Coates and Ng, 2012]. We use the same stimuli and whole-cortex dataset as in [Huth et al., 2012] to be able to use the Motion Energy and Semantic features as reference models. The derived feature maps are evaluated for functional and anatomical ROI-wise voxel coverage, noise (derived from Relevant Dimensionality Estimation) and Representational Dissimilarity in comparison to the reference models.

The basis for deciding to study this soft K-means model was the observation of a performance increase when encoding purely *static features* over a hand-crafted GFP based model with this soft K-means variant. Furthermore, K-means-based representation learning keeps the number of intrinsic parameters low, allowing to shift the tuning effort to highly influential side parameters like the receptive field size or even K [Coates et al., 2011]. Also, early in this project we assumed that from a biological point of view it could be plausible that the brain uses similar mechanisms (see section 4.1), having the idea of a general neural learning algorithm in mind.

There have been few publications attempting to encode spatial or spatio-temporal stimuli with a Bag of Words-type models: In [Hu et al., 2014] it was shown that a two-layer multiple-firing hard K-means hierarchy could predict voxel activity in response to static images up to V2. This was done on static artificial stimuli, with electrophysiological measurements of neuron unit responses, but not over the whole cortex as it is possible in fMRI. The publication also noted that K-means does fit into the framework of competitive Hebbian learning. Another publication, [Ramakrishnan et al., 2015], compared the biologically plausible HMAX model to a SIFT-features based Bag of Words model with histogram features. However while they encoded 10-minute video stimuli, they did not make the attempt to form a hierarchy.

Including the temporal dimension into Bag of Words-type models is considered as non-trivial in literature [Konda et al., 2013], [Hyvärinen et al., 2009]. At the beginning of this thesis project, the benchmark model for video feature learning was still the hierarchical ISA model proposed in [Le et al., 2011]. An important downside of this model is its computational effort, which would have made it difficult to vary side parameters. In [Konda et al., 2013], this ISA model had been compared to K-means based approaches as well to reduce the computational effort. Their *synchrony K-means* approach was evaluated at the beginning of the thesis through communication with its authors, however it has lead to less satisfying results in comparison to the hierarchy proposed by Adam Coates.

1.3 Structure of the document

The thesis document is structured as follows:

Chapter 2 Materials and Methods first describes the data sets and the encoding model framework. We introduce our K-means model in an overview, and then every step in detail. We also introduce our main analysis methods, in particular the linear model, noise analysis based on Relevant Dimensionality Estimation and Representational Dissimilarity Matrices.

Chapter 3 Results and Discussion presents the model performance and all main results and discusses them in detail.

Chapter 4 Further Discussion considers a biological implementation of the K-means model, debates its limitations and generally explores the potential of deep encoding models.

Chapter 5 Conclusion provides a short summary of the findings of this project.

Appendix A Reference Models introduces the Motion Energy model in detail and describes the `word2vec`-based Semantic features.

2 Materials and Methods

This chapter first introduces the feature learning hierarchy and then describes the components of the sequence in detail. Also, the methods used for model evaluation are introduced. For a description of the reference models see Appendix section A.

2.1 Experimental data and protocol

2.1.1 The fMRI dataset

We used the 3T dataset recorded over the full cortex that was first presented in [Huth et al., 2012]. The experimental design and stimulus dataset for this publication were originally introduced in [Nishimoto et al., 2011]. In the original experiment, data collection was only conducted in the occipital lobe with a 4T scanner on 3 subjects in a single-subject study framework. The [Huth et al., 2012] category model uncovered widespread semantically continuous object representations over a large section of the whole cortex. We chose the full cortex 3T data to be able to compare to the results of semantic encoding models. While in total 6 male subjects where recorded for [Huth et al., 2012], within the thesis document we are only allowed to present visual results for the data of a single subject. However we have analysed our models on all subjects within the scope of this project.

A drawback of the data set choice is that the temporal sampling rate of the 3T dataset used here is TR = 2sec, whereas it was TR = 1sec in [Nishimoto et al., 2011]. Another drawback is the smaller voxel size (spatial resolution), which is $2.24{\times}2.24{\times}4.1\text{mm}^3$ in our 3T data and was $2.03{\times}2.03{\times}2.5\text{mm}^3$ in the 4T data over the occipital lobe from [Nishimoto et al., 2011]. Nonetheless we considered the possibility to explore whole cortex data an advantage.

The training set was acquired in 12 scans of 10 minutes, leading to 3600 training samples in total for every subject. The test set consisted of 270 samples. See Table 2.1 for all experimental parameters.

The test set was collected in 9 separate 10-minute scans scattered throughout the training set acquisition. Each of these scans consisted of 10 1-minute test set blocks, which were permuted randomly (fixed sequence for all subjects) within these 10-minute presentations. This means that in total 5400 seconds of test data have been acquired. The test data set

Scanner model	3T Siemens TIM Trio, 32-channel coil
Scanning center	UC Berkeley Brain Imaging Center
Repetition time (TR)	2.0045s
Echo time (TE)	31ms
Flip angle	70°
Voxel size	$2.24\times2.24\times4.1\text{mm}^3$
Axial slices	32 (entire cortex)
Matrix size	100 × 100
Projection screen position	24 × 24 of the visual angle
Field of view (FOV)	$224 \times 224\ \text{mm}^2$
Number of recorded voxels	30662 (single subject)
Training set acquisition	3600 acquired samples (7200 seconds recording)
Test set acquisition	9 equi-stimuli scans, 10min each 270 acquired samples (5400 seconds recording)
Playback framerate	15Hz

Table 2.1: fMRI recording parameters, originally listed in [Huth et al., 2012]. The original publication also includes fMRI data preprocessing details.

was then taken as the average BOLD response to the same stimuli over all these runs, leading to a 270 samples test set at TR = 2sec. This resampling recording routine accounts for effects of adaptation, long-term drifts or other random influences like subject drowsiness or inattention which could corrupt the much shorter test set. This also leads to a clean signal for evaluating the actual representation in the brain, which is preferable for encoding models since their intention is studying representations, not predicting data on-the-fly. The Motion Energy model and the category model were evaluated based on this averaged test dataset. The original individual test set recordings are available as a further resource. Learning was done on the longer and noisy training data, which was acquired in single recording sessions.

2.1.2 The video stimulus dataset

The stimulus video dataset and its preprocessing steps have been described in detail in the experimental procedures and supplemental information of [Nishimoto et al., 2011]. We repeat its main properties in brief: The video data consisted of movie trailers, clips from thematic video libraries and a small set of HD movies from YouTube. The reason for assuring a wide variety was to reduce bias in the training stimulus selection. These video clips where cropped at the sides to form a square-shaped stimulus, and resized to an identical playback resolution of 512 × 512. Clips of a length of 10 to 20 seconds were extracted and concatenated into the training and test set. The individual frames were replayed at slow motion (15Hz) during the experiment due to the low temporal resolution of fMRI recordings.

To reduce the computational load during the feature extraction and evaluation, the movie clips where rescaled to 96 × 96 before entering the feature extraction routine. Any model with raw pixel-based features (Motion Energy and K-means) was evaluated on these rescaled frames.

2.1.3 Framework for voxel-wise encoding of stimuli

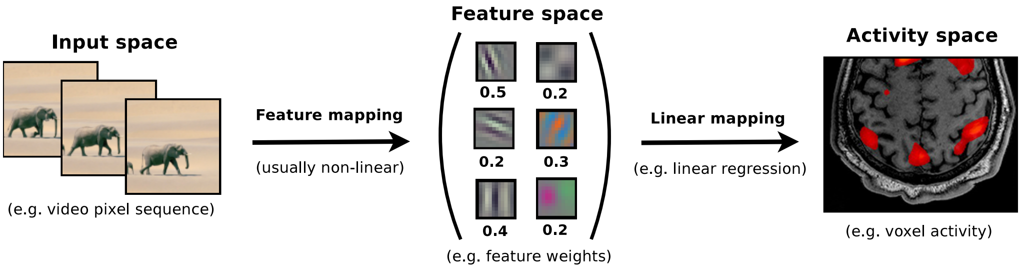


Figure 2.1: Spatio-temporal stimulus encoding for BOLD signals. A encoding model is created for the activity of every voxel (mass-univariate approach).

Next to using one of their datasets, we also follow the same approach for voxel-wise encoding that is used by the Gallant Lab at UC Berkeley, illustrated in Figure 2.1. In this framework they usually study fMRI data during passive perception of natural stimuli. Here these stimuli are natural video stimuli, i.e. spatio-temporal stimuli with natural image statistics as described in section 2.1.2. This input space is transformed into a feature space that is assumed to be the actual cortical input representation. The theory behind input transformation and feature extraction relies on the idea that each cortical area responsible for perception represents information explicitly that is only implicitly contained in the input[1]. This implicit information should be uncovered by choosing an appropriate feature extraction method or encoding model. The newly formed feature vector is then used for a voxel-wise prediction of the BOLD signal at the time when the underlying stimulus occurs. In our case of spatio-temporal stimuli, we use a feature vector that extends into the temporal dimension, i.e. there is one feature vector for every sampling point. This feature vector is used for the prediction of the complete BOLD signal (i.e. a prediction for every sampling point) recorded in one voxel during the training set recording.

For the brain signal prediction, usually a linear model such as ridge regression is used. One practical reason for this is that regularized linear models are better at preventing overfitting than non-linear models. The theoretical reason is that linear models are better suited for the ideas behind encoding models in general (details in section 2.3.2). The Gallant Lab also tries to avoid overfitting the predictive model by recording very long (multiple hours) scanning sessions. They furthermore follow the general model evaluation framework of machine learning by training the regression model on a training set, evaluating their models solely on a separate test set, and searching for ridge regression parameters in a cross-validation routine.

[1] Description of the Gallant lab approach: `http://gallantlab.org/approach.html`

While the first step, the encoding into a new feature space usually relies on a type of feature extraction that is believed to occur in the sensory cortices, in the linear modelling step other models of cortical processing such as sparsity can be introduced.

In a final evaluation, both [Kay et al., 2008] and [Nishimoto et al., 2011] attempted to detect (and effectively reconstruct) the input from the test set.

2.2 Details of the K-means based model

This section first provides a brief overview on how our K-means-based model is trained. Subsequently all steps of this unsupervised model are described in detail.

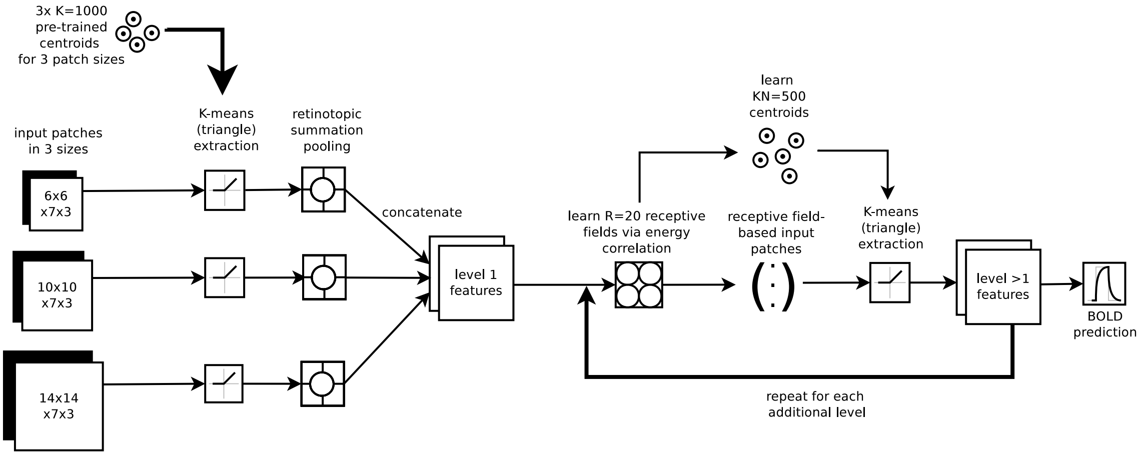


Figure 2.2: The hierarchical K-means model. The initial layer extracts features from 4D spatio-temporal patch blocks using the pre-trained 4D spatio-temporal centroids. Each successive layer then first computes energy correlation on the feature data from the previous level and forms receptive fields. It then extracts receptive-field based patches from the previous level's feature data for learning new receptive-field based centroids. These new centroids are used for the receptive-field based feature extraction from the previous level.

2.2.1 Overview

The prototype extraction and training in the first layer generally follows [Coates et al., 2011], while each successive layer is based on the receptive field learning method using energy correlation from section 5 of [Coates and Ng, 2012]. Since we assume that preserving spatial structure is important for encoding models for the visual system, we decided to learn the receptive fields within 2×2 spatial regions[2].

[2] This idea has briefly been described in section 3.5 of [Coates and Ng, 2011]. Details were then developed in the course of the thesis project.

The complete model is illustrated in Figure 2.2. The initial input data to the K-means learning hierarchy are 4D patch hyper-cuboids (subsequently (spatio-temporal) patch block or simply patch) randomly extracted from the $96 \times 96 \times 3$ training frames. These 4D patches are used to train the centroids for the first layer. In contrast to [Coates et al., 2011] we decided to use three different spatial patch sizes, resulting in centroids of three different sizes. We decided to do so since there is evidence that in the visual system image features of different scales are represented. [Nishimoto et al., 2011] also used a grid structure with different spatial block sizes for this reason.

Successively iterating through all frames of a separate pre-training video dataset, a random patch of the current patch size is extracted from each frame until the prototype training patch dataset for the current patch size contains the desired number of patches. The patches are whitened and then used to learn $K = 1000$ K-means prototypes. The procedure is repeated for each of the three patch sizes in the first layer.

Using these prototypes, continuous features are extracted for each frame using a soft K-means feature extraction (referred to as *K-means (triangle)* in [Coates and Ng, 2011], see section 2.3). These three feature vectors are then concatenated to learn the next layer, where pair-wise energy correlation is used to uncover feature dependencies. The pair-wise correlations are used to form receptive fields by grouping the output features into RN = 20 groups of $T = 200$ highly correlated features. The receptive fields are used to extract new patch blocks from the feature data of the previous layer, on top of which the next layer's K-means centroids are learned and used for feature extraction. The number of prototypes for all higher levels is KN = 500. Each new higher layer repeats this process. To reduce complexity, we decided to use the same parameters RN and KN for every new layer. We determined these parameters with a simple grid search over the higher layer training routine and the ridge regression prediction.

The *output data* of every hierarchy layer is a feature vector describing every single stimulus frame and its 7 successive frames[3]. This feature vector will then be downsampled to TR and fitted to the BOLD activity with ridge regression as in [Nishimoto et al., 2011].

2.2.2 K-means algorithm

The K-means algorithm is a point-assignment clustering method that iteratively leads to a set of cluster prototypes (centroids). In the first level of our model, the data points are image patches that are extended into the temporal and color dimension, forming 4D spatio-temporal patch blocks. The patch blocks are flattened to a vector before centroid training. Initialized at K random data points, the desired prototypes are adjusted in each iteration using algorithm 1.

[3] The constant temporal length of $T7$ frames was chosen since it was best performing during the previous lab rotation project. At a presentation frame rate of 15Hz 7 frames are 0.25TR.

The *objective function* that is minimized by standard K-means can be defined as in [Bishop et al., 2006]:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\bar{x}_n - \bar{p}_k||^2 \tag{2.1}$$

$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \; ||\bar{x}_n - \bar{p}_j||^2 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

The $\bar{x}$ are the spatio-temporal patch blocks in the training set and the $\bar{p}$ are the prototypes that are optimized in every iteration. r_{nk} describes the assignment of the data points to their closest centroid.

```
Data: $N$ standardized and whitened image patches $\bar{x}$
Result: $K$ cluster prototypes $\bar{p}$
Initialization: $\bar{p} \leftarrow K$ random data points from $\bar{x}$ ;
for 50 iterations do
    for each image patch $\bar{x}_n$ do
        for each patch prototype $\bar{p}_j$ do
            Compute the euclidean distance $d(\bar{x}_n, \bar{p}_j) = \sqrt{||\bar{x}_n - \bar{p}_j||_2}$ ;
            Find the closest patch prototype $\bar{p}_j$ and assign $\bar{x}_n$ to it ;
        end
    end
    for each patch prototype $\bar{p}_j$ do
        Adjust $\bar{p}_j$ to the mean of all images patches $\bar{x}_p$ assigned to it ;
    end
end
```

Algorithm 1: Standard K-means algorithm used in this thesis.

Unsupervised centroid training For training the K-Means centroids in the first layer, a separate RGB video data set was used. This was scaled down to the same spatial resolution as the stimulus dataset ($96 \times 96 \times 3$) during processing, and was randomly extracted from a collection of the same category of videos as the stimulus set. It was made sure that the videos contained in this dataset were not contained in the stimulus data. The higher-level centroids were trained on the stimulus dataset.

For training the first level, from this separate video dataset we extracted 2.500.000 $S6 \times S6 \times T7$ and $S10 \times S10 \times T7$ RGB patches, and due to memory limitations 2.000.000 patches for the largest patch size $S14 \times S14 \times T7$. Since we were using RGB patches, a spatio-temporal patch block actually had the dimension $S \times S \times T \times C$. The color dimension has been included within the flattened patch vectors used for training to avoid learning RGB color information separately from each other.

Furthermore, we decided to enclose the temporal dimension as a component of each RGB color channel. In the implementation this meant combining the color dimension with the temporal dimension by concatenating the $T7$ temporal columns in the patch block within the current color channel. We could not find consensus in literature on the feasibility and process of combining the temporal and the color dimension in Bag of Words settings. We however observed that whitening the raw pixels contained in the patch blocks in this arrangement effectively leads to temporal decorrelation. Analogous to the importance of spatial whitening in [Coates et al., 2011] we expect that the correct application of temporal whitening is important as well.

Before entering the K-means unsupervised learning routine, the training patches were flattened to a vector[4], centred and PCA-whitened with $\epsilon = 0.1$. In image data, ϵ leads to slight smoothing and noise reduction. Note that [Coates et al., 2011] identified the whitening step as crucial for a well-performing K-means image classification model; and [Vinnikov and Shalev-Shwartz, 2014] as one crucial processing step leading to ICA-like filters.

To align the resulting feature vector with the BOLD sampling rate TR = 2sec, the feature vectors needed to be downsampled. For this reason and for comparability, we again adopted the routine from [Nishimoto et al., 2011], where the feature vectors sampled at 15Hz where downsampled by taking the average over TR = 2sec.

Due to the concatenation of the stimulus video clips, specific sequence blending prototypes are learned, usually switching color filters. We did not remove patches with such transitions from the randomly selected training set since we regard them as a visual feature that could be reflected in the BOLD signal.

2.2.3 Feature extraction and pooling

The prototypes learned for the first K-means layer are the basis for extracting features from the actual stimulus data. For feature extraction, each possible spatio-temporal patch with a step size (stride) of $s = 2$ pixels is extracted from every frame for each of the 3 patch scales. $K = 1000$ continuous scalar feature values are extracted from each of these patches by applying the soft K-means feature extraction described in section 2.2.4. Summation pooling is applied to the resulting patch-wise feature vectors in 5 circular pooling regions (described in section 2.2.5), still individually for each scale. This results in 5000 features for each of the 3 scales, which are concatenated to form the feature vectors in the first layer for each frame. To avoid overfitting due to the high feature dimension in the first layer, predictions are done separately for the three scales, and correlation values are combined[5]. The complete feature extraction process is illustrated in Figure 2.3.

[4] Flattening was done in `MATLAB`'s column major order.

[5] This combination was based on the training set prediction results: For predicting a specific voxel i we took the 5000 features of the single scale that resulted in the best correlation in the training set. However, we also found that the correlation results of the *scale-wise predictions* and the prediction based on the *full (concatenated) Layer 1 feature vector* of 15000 features were effectively equal. In section 3.5.3 we present results that indicate a spatial distribution of the different scales on the cortex, which is a possible explanation for this equality.

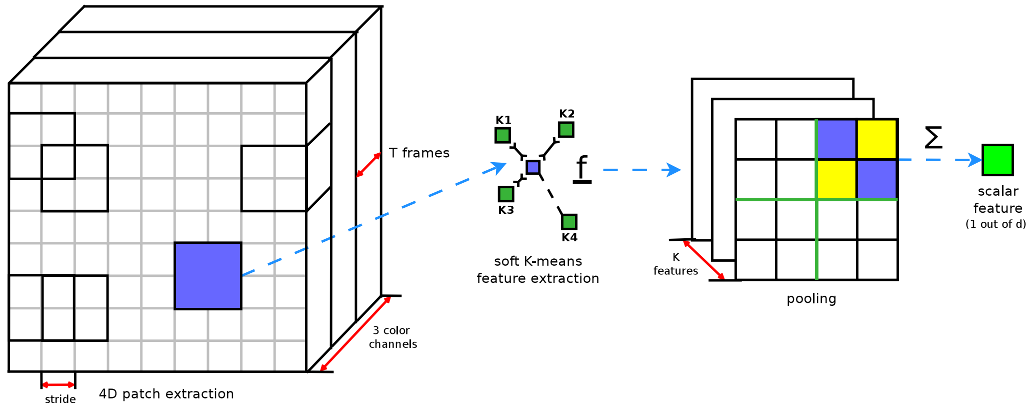


Figure 2.3: Feature extraction and pooling in the first layer. From the raw pixel input all possible patches are used for feature extraction, forming one $N_{dim} = 1000$ feature vector for every patch. Subsequently, spatial pooling over these feature vectors reduces the feature dimension and introduces invariance and smoothing.

2.2.4 K-means (triangle) feature extraction

A *hard K-means cluster assignment* would assign the closest centroid to each of the patch blocks, leading to a binary representation. Here we apply the same *soft K-means* feature extraction that was used in [Coates et al., 2011]. This feature extraction is performed for every centroid and leads to K real-valued features for every patch block.

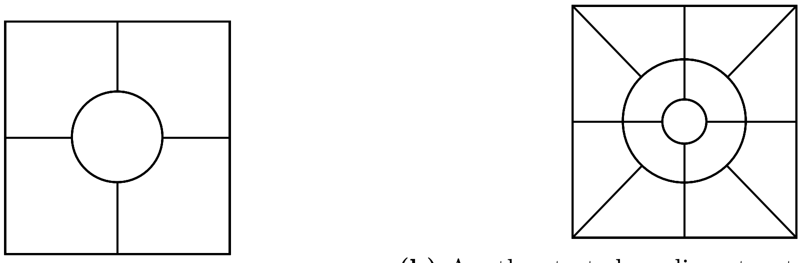
$$f_k(x) = max(0, \bar{d} - d_k) \tag{2.3}$$

$\bar{d}$ is the mean distance of the patch block x to all K prototypes p_k. $d_k = ||x - p_k||_2$ is the quadratic euclidean distance used by the standard K-means algorithm. The result of this feature extraction method is one feature vector for every patch. Every feature vector consists of K continuous values, i.e. one distance to every prototype. If the distance between the considered patch and one of the K prototypes is larger than the average distance of the patch to all prototypes, the distance is not considered by setting the feature value to 0.

In [Coates et al., 2011], this method was called *K-means (triangle)* feature extraction. Next to whitening, the fact that this feature extraction method is leading to continuous feature values was their second reason for the applicability of this feature representation to their classification problem. In comparison, hard K-means assignment has been leading to lower label prediction performance.

2.2.5 Retinotopic summation pooling

Spatial pooling on a regular grid is likely not applicable to biological encoding models due to the retinotopical structure in the early visual system. The mapping of the visual field to



(a) The pooling structure used in this thesis.

(b) Another tested pooling structure that would have lead to too many features.

Figure 2.4: Pooling regions used to approximate retinotopy. After the first layer feature extraction, the feature responses are summed over each of the areas shown in this illustration.

the retinotopically organized cortex area is contorted; the fovea region being magnified on the cortex in comparison to the peripheral visual regions (see Figure 2.5). The grid used for pooling should reflect this structure and either single out or over-represent the small foveal region.

While fixation instructions were given during the experiment, it is likely that the subjects could not follow them over the whole course of an recording session. This however will lead to noise with any encoding model based on the raw pixel stimulus input.

The pooling field structure used for this project is illustrated in Figure 2.4a. We have attempted to use more detailed pooling regions (e.g. the one in 2.4b), however given that every additional pooling region increases the feature dimension by K the possible total number of pooling regions is limited. We additionally account for the overrepresentation of the foveal region by including it in every subfield in Layer 1 receptive field learning (see section 2.2.6).

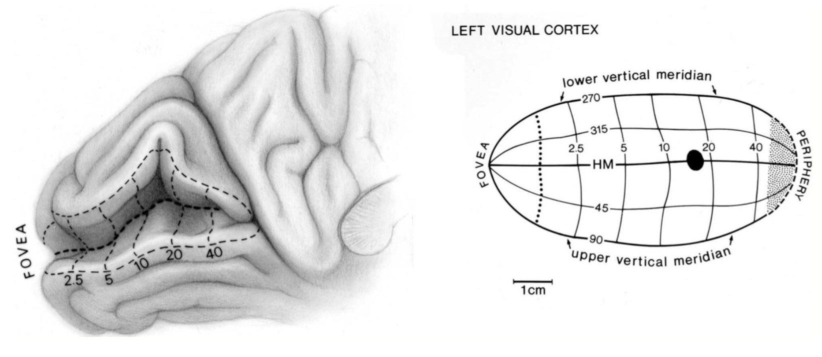


Figure 2.5: Estimation of the representation of the human visual field in the visual cortex from [Horton, 2006]. The left image shows the striate cortex folded up at the calcarine sulcus. The right image shows the location of the coordinates from the circular visual field (represented as in coordinates from a stereographic projection). The black circle is the blind spot. What we can learn from this illustration is that the foveal region is overrepresented in the cortex in comparison to the peripheral areas.

The summed filter responses for all pooling fields and patch sizes are concatenated to form the first layer feature vector for each frame. Within this concatenated feature vector it is possible to identify the original pooling regions, which allows preserving spatial information in the higher layers.

2.2.6 Receptive field learning

For every round of unsupervised prototype learning we need to extract data patches from the training data as a basis. The prototype learning in the first layer is based on pixel patches, which employ temporal and spatial coherence. Feature patches in each higher layer however can not be defined as to have spatial coherence because – due to prototype learning and pooling – they are randomly arranged in the K fields of the feature vector. To solve the problem of extracting new coherent data patches, we used *receptive field learning* as it was presented in [Coates and Ng, 2011] and [Coates and Ng, 2012]:

Receptive field learning will greedily form R sets of T features with *pairwise similar activity* from the previous feature vectors. Here each new receptive field is based on one single feature *randomly* chosen (and not chosen before) from the feature vector[6].

Equation 2.4 expresses the pairwise similarity heuristic. d_j and d_k are two features in the feature matrix, i.e. distances to a certain prototype across time.

$$c[d_j, d_k] = corr(d_j^2, d_k^2) = \frac{\sum_f d_j^{(f)^2} d_k^{(f)^2} - 1}{\sqrt{\sum_f (d_j^{(f)^4} - 1) \sum_f (d_k^{(f)^4} - 1)}} \tag{2.4}$$

This is in fact the correlation coefficient between the energies (squared responses) of the whitened data, where $\mathrm{E}[z] = 0$ and $cov(z) = \mathrm{E}[zz^T] = \mathbb{I}$. Equation 2.5 is this same correlation expressed with expected values. Due to equality of $\mathbb{I}$ and the expectation of the outer product $\mathrm{E}[zz^T]$, the expectations of the squared responses in equation 2.5 equal 1 as well.

$$corr(d_j^2, d_k^2) = \frac{\mathrm{E}[d_k^2 d_j^2] - \mathrm{E}[d_k^2]\mathrm{E}[d_j^2]}{\sqrt{(\mathrm{E}[d_k^4] - (\mathrm{E}[d_k^2])^2)(\mathrm{E}[d_j^4] - (\mathrm{E}[d_j^2])^2)}} \tag{2.5}$$

The process used for forming the receptive fields is expressed in algorithm 2. We use $T = 200$ as the receptive field size, i.e. the new "patch size". Next to changing this number having no apparent effect during our own test rounds, [Coates and Ng, 2011] stated that this value does not have a large influence on the result.

[6] Due to the random selection of the initial feature there is still the possibility that equal or almost equal receptive fields are extracted. Among the $R = 20$ that were trained in each session this however occurred infrequently (once or twice per training session).

```
Data: Training set feature matrix $X$ with feature columns $d_k = d_{k_1}...d_{k_{3600}}$ (with
      $k \in K$).
Result: $N_R$ receptive fields $R$ (set of feature indices)
Compute the pairwise similarity matrix $c(X)$ ;
for $N_R$ iterations do
    Select a random seed feature $d_r$ from $c$ with $r \in K$ ;
    $d_{\text{sort}} \leftarrow d_r$ sorted in descending order ;
    $R_n = \arg_{d_r}(d_{\text{sort}_1} \; ... \; d_{\text{sort}_T})$;
end
```

Algorithm 2: Extraction of receptive fields. Note that a receptive field R is a collection of feature indices. arg_{d_r} will therefore take the corresponding original indices from d_r.

Receptive field learning in detail We would like to provide more details on the receptive field learning procedure. In order to analyse an encoding model for the visual system, we considered the preservation of spatial information across layers as beneficial. Therefore, in general, receptive fields are learned within the pooling regions, and features are extracted based on the same receptive fields within the respective pooling region. The feature sets from all pooling regions are concatenated to form the final feature vector for one layer.

Layer 1 uses four peripheral and one central foveal pooling region (see the pooling regions in Figure 2.4). All higher levels use four pooling regions. In Layer 1 we decided to concatenate the four peripheral regions with the foveal region for learning each of the four receptive field sets. This increases the impact of the foveal region where the subjects fixated during the experiment. Note that in Layer 1 it was also necessary to account for the three different patch scales.

After receptive fields have been extracted for every pooling region, sets of training patches are created. This is done by randomly selecting a receptive field for every frame and extracting the corresponding feature value patch until the desired number of patches is collected. These receptive-field specific patch vectors are then concatenated, and K-means prototypes are trained over all collected patches. Within this procedure it is possible to use the same K-means prototypes for all pooling regions since the centroid indices underlying the Rs match across receptive fields and pooling regions.

Whitening transformations are applied to the feature data used for receptive field learning and on the training patches that are used in the K-means prototype learning. Leaving out each of these whitening steps has lead to decreasing predictive power of the feature sets. [Coates and Ng, 2011] described the linear decorrelation of the training patch vector as one precondition for the receptive field learning to pick up higher-order dependencies.

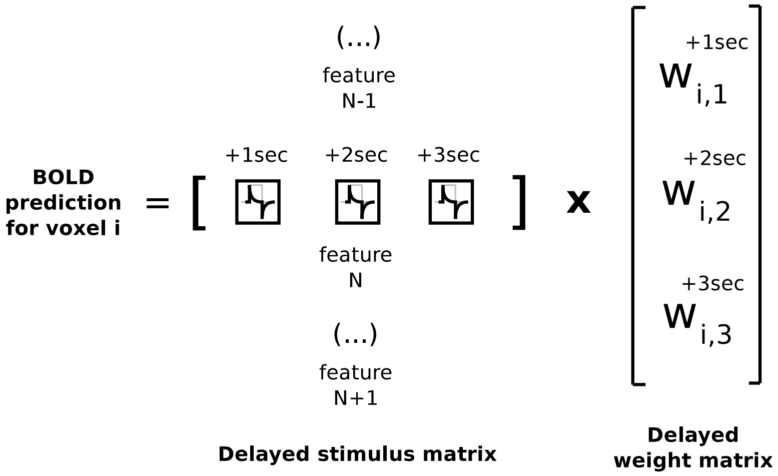


Figure 2.6: Linear model for predicting the BOLD signal in a single voxel i in response to feature vectors extracted from the video stimuli.

2.3 Evaluation methods

2.3.1 Reference models

The K-means-based unsupervised model is partly evaluated by comparing its results to two handcrafted models, which are the Motion-Energy model from [Nishimoto et al., 2011] and a `word2vec`-based feature space similar to the category model from [Huth et al., 2012]. Their detailed descriptions can be found in the Appendix sections A.1 for the Motion Energy model and A.2 for the Semantic features.

2.3.2 Ridge regression model

For linearly predicting the BOLD signal on top of the feature vectors the method described in [Nishimoto et al., 2011] and its supplementary material is used. However L2-regularized ridge regression is used instead of the L1-regularized LASSO regression used in the original publication. Refer to Figure 2.6 for an illustration of the process and the structure of the feature and weight vectors. In words, the process is structured as follows:

1. The temporal sequence of feature vectors is down-sampled to the TR of 2sec by *averaging* over the features in each sampling window.
2. The feature signals are shifted for delays of 1, 2 and 3 seconds and then concatenated to form the delayed stimulus matrix.
3. An equivalent delayed matrix exists for the regression weights. The hemodynamic response function is thus learned implicitly from the BOLD responses within the two delayed matrices instead of using an explicit HRF model. After learning the model via ridge regression, one predictive model (weight set i) exists for the BOLD signal in every voxel i.

4. Ridge regression is applied to learn the weights i that lead to the BOLD training data prediction. The regularization parameter λ is determined via 10-fold cross-validation, i.e. iterating over 10 regularization sets each consisting of 10% of the training data.

The model performance is measured by its *mean prediction accuracy on the test set*, which is defined as the z-transformed (Fisher's Z-transformation[7]) Pearson's linear correlation coefficient between the predicted and observed BOLD signals *within-voxels*. Note that we do not expect a homogeneous distribution of predictability over all voxels. What is expected is that a part of the voxels, e.g. in a subset of a ROI can be predicted very well with a specific feature set (i.e. a set of features matching the cortical representation), while the remaining voxels' activity can not be predicted. We therefore will need to look at the resulting correlation data from different perspectives in the results section.

Linearizing encoding models Using a linear model to fit the representation to the BOLD signal is based on the assumption that there is a simple relationship between the feature maps and the voxel activity. The encoding model should capture all non-linearities in itself and by careful design of the researcher, i.e. by intention. More powerful predictors would introduce new non-linearities that are either difficult to uncover, describe or interpret, hiding potentially crucial information about the encoding process. A powerful predictor might be capable of predicting any relationship between the raw pixels of the input space and the activity space, but could not be analysed or designed well in terms of understanding brain activity. In literature on encoding models this idea is referred to as *forming a linearizing feature space*. Refer to [Naselaris et al., 2011] for detailed information on this framework.

Evaluation is done on a single-subject basis, following the methodologies in [Nishimoto et al., 2011] and [Huth et al., 2012]. Here mass-univariate[8] models are trained for all feature data sets, within different ROIs and using the full training stimulus data. The ROIs are derived from functional localization experiments in [Huth et al., 2012].

Visual ROIs While sensorimotor and auditory ROIs have been mapped during the experiment, we have not used them during most evaluations. We call the following subselection of all defined ROIs *visual ROIs*: EBA, FBA, FEF, FFA, FO, IFSFP, IPS, LO, MTp, OFA, PPA, RSC, TOS, V1, V2, V3, V3A, V3B, V4, V7, VO, pSTS.

[7] We chose to apply Fisher's Z-transformation $f(r) = \frac{1}{2} \ln \frac{1+r}{1-r}$ to the correlations r because Pearson's correlation values are not interval spaced and therefore not additive, which is a prerequisite for comparing them against each other or for taking their mean value. For comparability, the final evaluation score for a ROI is then its Fisher-weighted mean correlation coefficient, including the application of the inverse transformation. Voxel correlations are left in the Fisher-Z-transformed form for comparability however if not specified otherwise. This transformation had not been applied in [Nishimoto et al., 2011], which potentially had lead to a lower influence of the more meaningful higher correlated voxels in their specified average correlations.

[8] Mass-univariate refers to analyzing every voxel separately here, in our case to training one regression model for every voxel.

2.3.3 RDE noise estimation

We use Relevant Dimensionality Estimation (RDE) as described in [Braun et al., 2008] to study the noise of the prediction task in the light of *a specific representation* (feature set or model) without training a ridge regression model. This method is also applicable for small sample sizes, which is beneficial in our case since there is only a limited number of samples recorded for every voxel.

[Braun et al., 2008] could show that the feature dimensions relevant for the learning problem are often contained in the first few principal components, and that a kernel suitable for the learning problem is capable of exploiting the data efficiently. Based on this insight they presented a simple procedure for finding the relevant number of dimensions (i.e. the number of relevant kernel PCA components) for a specific learning problem. This can be applied e.g. to reducing the dimensionality adequately and to studying appropriate kernels. In subsequent publications, RDE was used for studying deep neural network representations in [Montavon, 2013] and – in a similar setting to ours – comparing brain representations with deep neural network layers in [Cadieu et al., 2014].

Here we mainly make use of noise estimation, which is a diagnostic tool derived from RDE. Note that this is performed both *in the light of a specific feature set* (e.g. the Motion Energy features or a single layer of the K-means routine) and in the *in the light of a specific kernel*, which in our case is the *Gaussian Kernel* with an adaptive scale parameter σ. Our assumption when using this method is that when using different feature sets for building the kernel we should still see similar distributions of well-predictable and unpredictable (noisy) voxels that we saw in the regression model.

We now briefly review the RDE process from [Braun et al., 2008] applied to our framework of a mass-univariate learning problem: Our training dataset $Y \in \mathbb{R}$ consist of 3600 samples of the BOLD signal for each individual voxel. For deriving the Kernel matrix K we use the feature data $X \in \mathbb{R}$ with the dimensionality N_{dim}. It is important to note that we do not use the original feature and signal data vectors here, but their `circshifted` version, described previously in section 2.3.2. Here once more this is in order to account for the temporal shifts introduced by the haemodynamic response function.

Kernel PCA refers to an extension of Principal Component Analyses with Kernel methods. We start by describing its typical routine as originally described in [Schölkopf et al., 1998] and [Mika et al., 1998]. We decided to use a Gaussian kernel (Equation 2.6) since this results in a full-rank Kernel matrix.

$$K_{j,k} = \exp\left(\frac{-(d_j - d_k)^2)}{\sigma_{scale}\ \mathrm{E}[(d_j - d_k)^2]}\right) \tag{2.6}$$

As we see above, the scaling parameter $\sigma = \sigma_{scale}\ \mathrm{E}[(d_j - d_k)^2]$ is essentially based on the expected value of the pairwise distances. σ_{scale} is selected from $0.1, 0.5, 0.9$ by taking the σ_{scale} leading to the lowest mean noise $\frac{1}{N_{dim}-d}\sum_{i=d+1}^{N_{dim}} z_i^2$. How to determine Z_i is

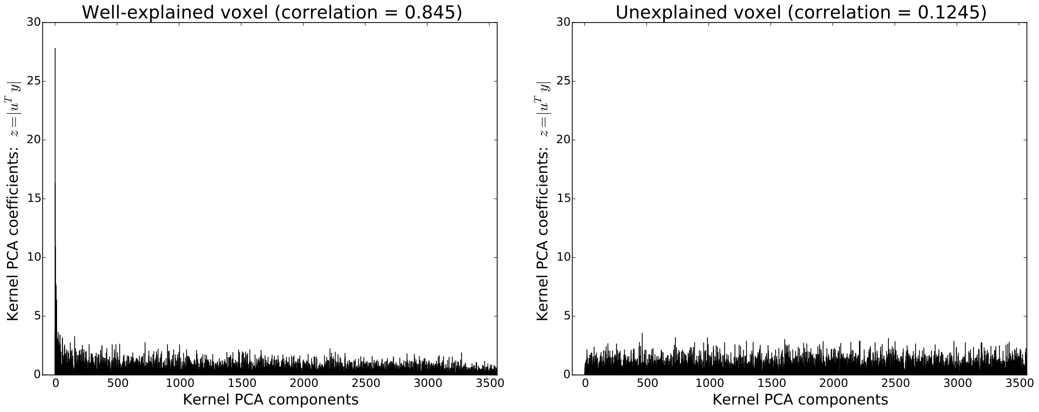


Figure 2.7: Example kPCA coefficients for the Motion Energy features for one well-explained and one unexplained voxel from the mass-univariate RDE. Data of the first recording session is used for this example.

described subsequently. After K has been derived, we apply eigendecomposition to obtain the eigenvectors u and eigenvalues λ.

$$Ku = \lambda u \tag{2.7}$$

[Braun et al., 2008] proceed by estimating the relevant dimensionality with two different methods. For our own analysis we make use of the noise assessment that is obtained with their *parametric model*. For this method we only need the spectrum coefficients z, which are the dot products between the eigenvectors $u_1, \ldots, u_m$ and the BOLD signal of a specific voxel i $y_i \in Y$.

$$z_m = u_m^T y_i \tag{2.8}$$

The coefficients can be understood as to which extend the principal component u_m reflects the measured BOLD signal y_i of a specific voxel. Note that while K only exists for a specific feature set, these coefficient spectra exist for every voxel due to our mass-univariate model. Figure 2.7 shows the kPCA coefficients z_m for one well-explainable and one unexplained example voxel in the light of the Motion Energy features and the Gaussian kernel.

The actual RDE uses the z to derive the relevant dimension and the noise with their *parametrised model*, which is a Two-Component Model (TCM). Here [Braun et al., 2008] observe that Y can be decomposed into a signal and a noise component, i.e. $Y = G + N$. The noise N is always included as an *evenly distributed noise floor* while the signal G only exists in the principal components arising above it. The idea of their TCM is to find a cut-off point d between the coefficients that carry the signal (plus the noise floor) $z_1, \ldots, z_d$

and those that only carry the noise floor $z_{d+1}, \dots, z_{N_{dim}}$. This is done by modelling the two parts of the coefficients with two Gaussians with zero mean and $\sigma_1 \gg \sigma_2$:

$$z_i \sim \begin{cases} \mathcal{N}(0, \sigma_1^2) & \text{if } 1 \leq i \leq d, \\ \mathcal{N}(0, \sigma_2^2) & \text{if } d \leq i \leq n. \end{cases} \tag{2.9}$$

Incorporating our z_i, we have $\sigma_1^2 = \frac{1}{d}\sum_{i=1}^{d} z_i^2$ and $\sigma_2^2 = \frac{1}{n-d}\sum_{i=d+1}^{n} z_i^2$. From these [Braun et al., 2008] find the z_i and the desired cut-off parameter d with a maximum likelihood estimation:

$$d = \underset{1 \leq d \leq n}{\operatorname{argmin}}(-\log l(d)) = \underset{1 \leq d \leq n}{\operatorname{argmin}}(\frac{d}{n} \log \sigma_1^2 + \frac{n-d}{n} \log \sigma_2^2) \tag{2.10}$$

The cut-off point d is used to estimate *voxel-wise noise* with Eq. 2.11.

$$\text{noise} = \frac{1}{N_{dim} - d} \sum_{i=d+1}^{N_{dim}} z_i^2 \tag{2.11}$$

This has been mentioned in [Montavon, 2013] and is derived from a loss function between the real Y and the denoising projection $\sum_{i=1}^{d} u_i u_i^T Y$. The spectrum coefficients can be related to this loss function as in equation 2.12. The noise levels from Equation 2.11 can be understood as average error residuals.

$$e(d) = ||\sum_{i=1}^{d} u_i u_i^T Y - Y||^2 = \sum_{i=d+1}^{N_{dim}} z_i^2 \tag{2.12}$$

As mentioned at the beginning of this section, one advantage of this method is that the estimated dimensionality and noise are robust to the sample size. In brain signal analysis, the number of samples is generally small and therefore likely to lead to an overfitted model. This is essentially also the case for our hours-long single subject recordings, which is why we decided to use this method to verify our ridge regression models.

One possible downside of this method is that the Gaussian Kernel might violate the *linearizing* principle of encoding models (described above in section 2.3.2), i.e. by introducing unintended non-linearities. In the case of our feature and sampling dimensions, a purely linear kernel however would (apart from leading to RDE with normal PCA) not have been full-rank as the Gaussian kernel is. We decided to use the Gaussian Kernel and therefore trade the full-rank property for this possible downside.

2.3.4 Representational dissimilarity

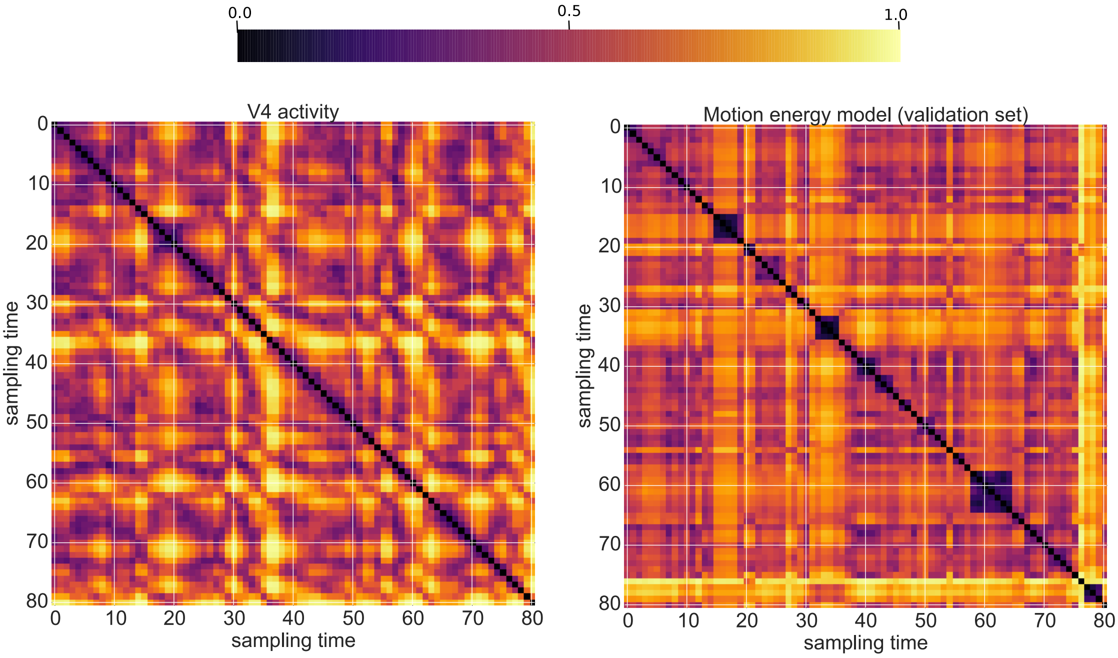
Representational Dissimilarity Analysis is a simple method for comparing representations in different domains introduced in [Kriegeskorte et al., 2008]. Representational Dissimilarity

Matrices (RDMs) visualize which differences in stimulus conditions are detected by a modality. This is done by plotting their differences (measured with any distance metric or correlation) in matrix shape. This is intended to allow comparisons between different measurement modalities, ROIs and with different feature sets.

In the case of this experiment, the RDMs for a ROI or feature set show the sampling points of the test set in time as stimuli on their axes. The difference between the stimulus representations is estimated by applying the *inverted Gaussian Kernel function* without σ-adaptation between each pair of representation vectors of sampling points in the stimulus video:

$$\mathrm{RDM}_{j,k} = 1 - \exp\left(\frac{-(d_j - d_k)^2)}{\mathrm{E}[(d_j - d_k)^2]}\right) \tag{2.13}$$

Figure 2.8 shows two example representational similarity matrices for the *test data set* our single subject. Note that the axes represent the sampling points in time. This evaluation method is not feasible for the training dataset, where the noise inherent in the single recording prevents the emergence of meaningful RDM structures. The resampling of the test data set leads to clean structures arising in the RDMs. Note that *meaningful* RDM structures are only identified by mere *looking at the RDMs*.



(a) Representational Dissimilarity of the test stimulus set in a specific ROI.

(b) Representation Dissimilarity of the test stimulus set in a specific feature set.

Figure 2.8: Example Representational Dissimilarity Matrices for the first recording session of the test set. Note that the RDMs also reveal the slight shift caused by the haemodynamic response function in the BOLD signal (here in V4).

For quantifying the comparison of RDMs, often their triangular forms are simply correlated. We avoid this due to discussions about the uncertainty about what can be claimed by correlating correlations (or distances). We instead use them solely for visualization. Note that since we used the Gaussian Kernel function the RDMs also visually represent the (temporally sorted) Kernel matrices from Relevant Dimensionality Estimation.

3 Results and Discussion

In this section we are evaluating the models visually and quantitatively for a single subject. We focus on the capabilities of the K-means model, and compare the results to the Motion Energy and Semantic features.

3.1 Voxel predictability over the cortex

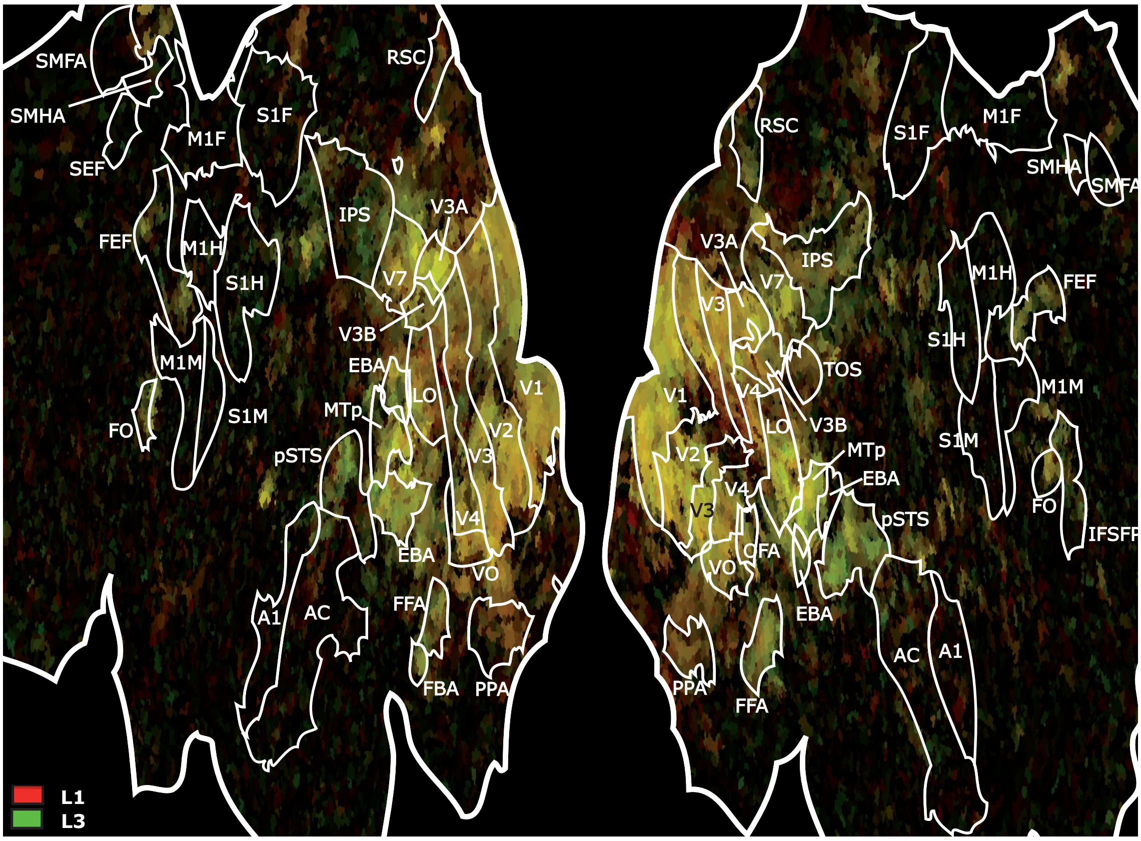


Figure 3.1: Prediction-activity correlations on a cortical flatmap for Layers 1 and 3 of the K-means model. Yellow voxels can be predicted similarly well by each layer's features. These flatmaps show the left and the right hemisphere, where the central regions correspond to the occipital lobes.

The RGB channel figures in this section show the distribution of the voxel predictability over the cortex in proportion for different model choices. The Regions of Interest (ROIs) are identical to the ones defined in [Huth et al., 2012]. What we would expect from a hierarchical encoding model of the visual system is a gradual transition from predicting the lower visual cortex areas with the lower layer encoding model features to predicting higher order cortical areas with the higher layers. The optimum would be a distribution similar to Figure 3.2, which shows a *clear distinction* between the lower visual system covered by the Motion Energy model and the higher order visual areas covered by the Semantic features.

What Figure 3.1 mainly demonstrates is that the magnitude of voxels can be predicted similarly well by both K-means Layer 1 and Layer 3, which is visible from the large proportion of mixed-color (here: yellow) voxels. A tint towards either red or green indicates that a voxel can be predicted best with the respective layer. While each of the layers reaches a slightly better predictability for some of the visual areas, these effect differences are weak. However, they are also consistent for some areas, and not random. Every layer is capable of mainly predicting visual system-related voxels and does not cover random fluctuations over the cortex. This can also be seen in Figure 3.3, which compares the first three K-means Layers. What we also can conclude from Figures 3.1 and 3.3 is that the K-means encoding and prediction routines taken together are generally capable of predicting visual system activity.

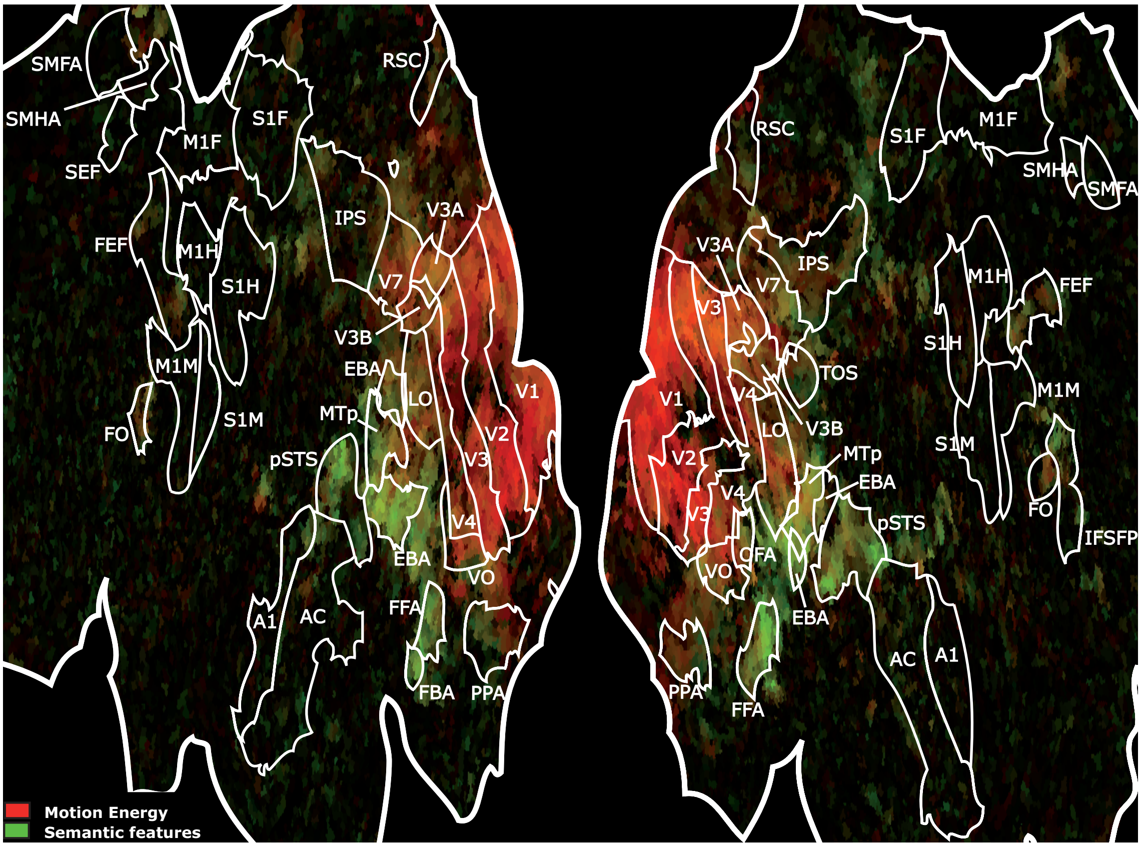


Figure 3.2: Prediction-activity correlations for the Motion Energy features and the word2vec Semantic features.

It is however obvious when comparing to Figure 3.2 that the hand-crafted Motion Energy and Semantic features are clearly superior to our completely unsupervised features. While the early visual cortex coverage of the K-means model is apparently resembling that of the Motion Energy features, the overall effect sizes are smaller. The transition into intermediate and higher visual cortex areas is mediocre. We will explore the predictive power quantitatively in section 3.2.

See Figure 3.3 for an RGB comparison of the K-means Layers 1, 2 and 3. We again see weak tints that indicate that a region can be predicted slightly better by one of the layers. However, considering the whitish areas we also notice that these differences between the predictive power of the three layers are only small.

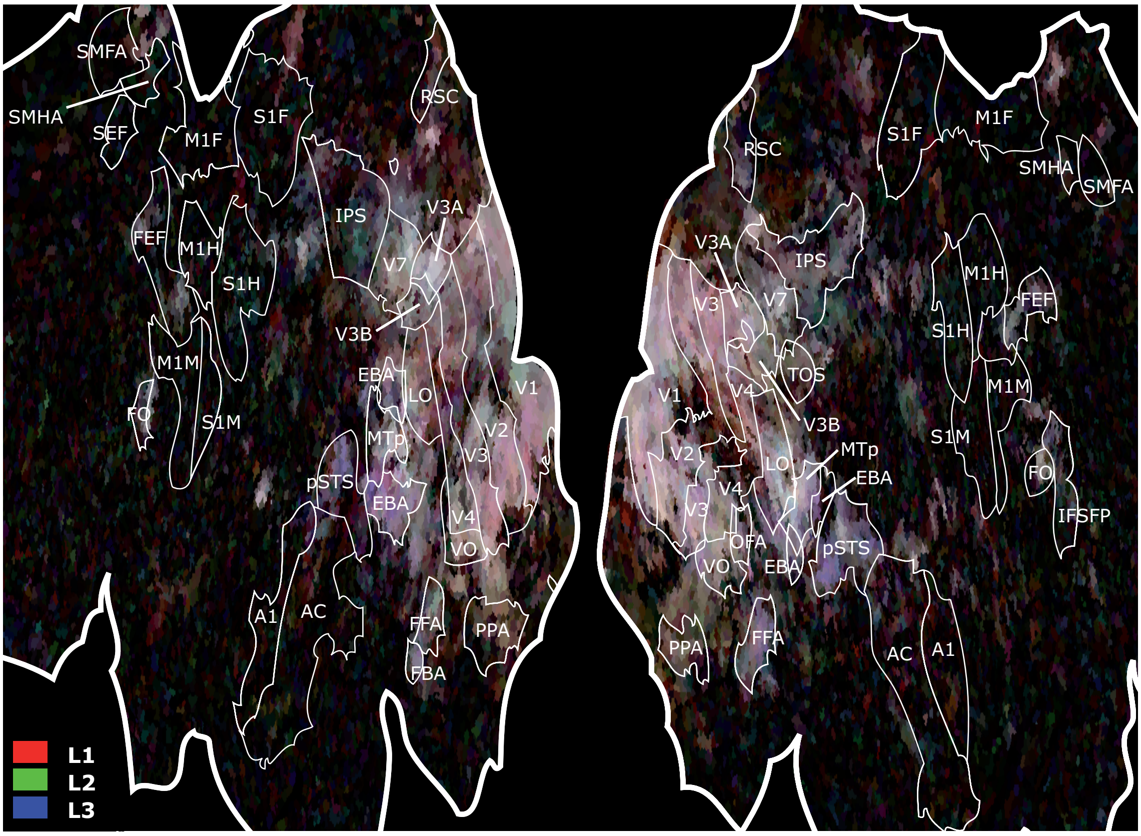


Figure 3.3: Prediction-activity correlations on a cortical flatmap for Layers 1, 2 and 3 of the K-means model. White voxels can be predicted equally well by every layer's features.

Session differences for higher layers and higher visual cortex areas The following section should be viewed as speculative and as a possible starting point for new experiments. We would like to point out that the full data was collected in three recording sessions with one third of the training and test data presented on each day. We could observe that the capability of the Layer 1 feature sets to predict visual activity improves when the full training data from all sessions is used. We could not observe this for the higher layers,

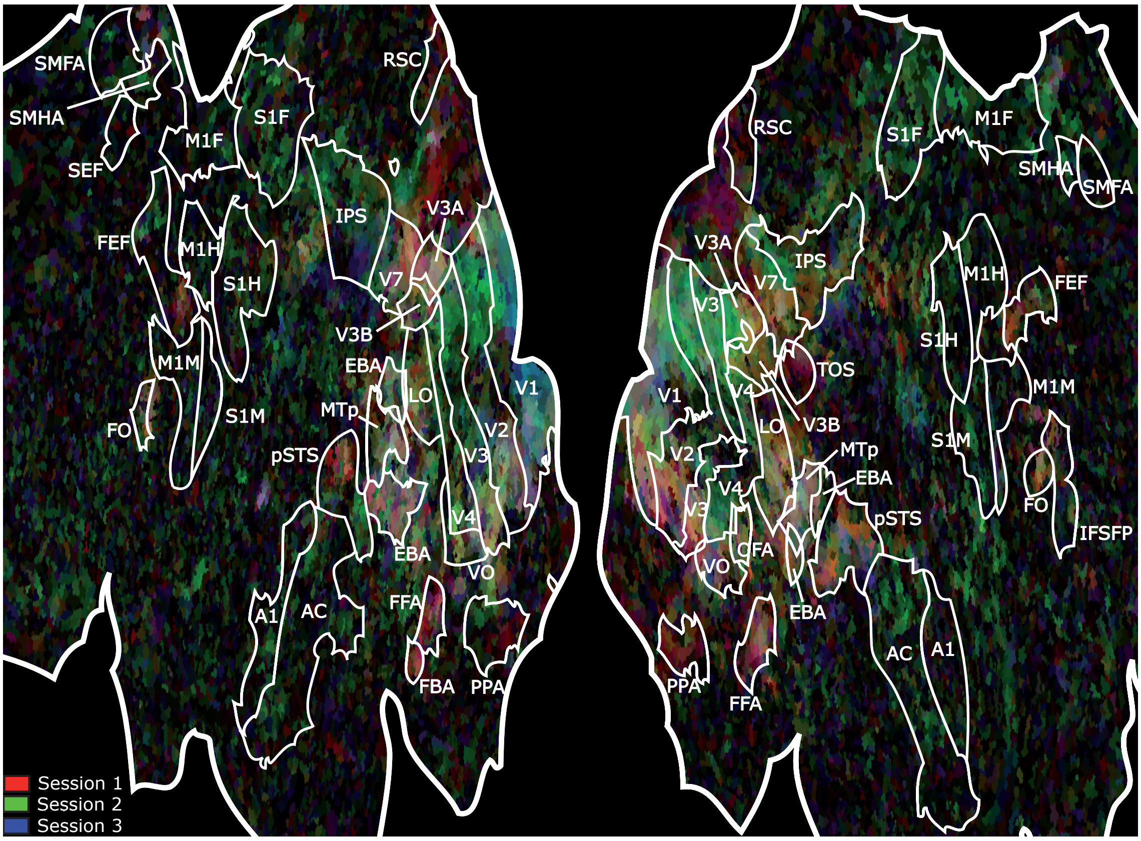


Figure 3.4: Differences between the three sessions when predicting with Layer 3 K-means features. Session 1: Red, Session 2: Green, Session 3: Blue.

however. Here, when the smaller training data set of an individual sessions is used, the predictability and coverage of new voxels improves over diverse regions. Figure 3.4 shows the differences between the models trained on individual sessions for our single subject.

This observation is unusual since less training data is used. The Semantic features lead to the clean (green) distribution that we see in Figure 3.2 when predicting with the training and test data of all sessions, but also show session-wise differences when predicting with the data from each day. Session-wise differences also occur for the other subjects. We have investigated this in-depth, but could not agree on a reason for this observation. Possible explanations include overfitting: When using the data of the *second session* of our single subject, many voxels outside of the visual system, especially in sensorimotor areas can be explained. However in other subject's and session's data this does not occur for sensorimotor activity. It is also possible that the higher K-means levels are only capable of explaining very specific features that dominated certain sessions. Another explanation could be strong attention modulation in the higher visual system: It has often been shown that visual spatial attention strongly influences the response in different areas of the visual system [Bressler et al., 2013]. But we must be careful with these last two assumptions since they would be evidence for partial correctness of our unsupervised K-means model. In our available experimental setting, results from the full data should be considered more reliable.

While these effects appear interesting, we can only conclude that a new experiment would be necessary to find an explanation for these differences. Such an experiment should cover both attention effects and differences in stimulus contents. In our setting we could not provide an answer on why these session differences occur.

3.2 Voxel predictability in specific ROIs

We would like to present quantitative results in this section. Figure 3.5 contains a complete overview of the results for four layers of the K-means model in comparison to the Motion Energy model and the Semantic features. Again, note that this is for a single subject. For comparability the correlations have been made interval-scaled via Fisher's Z-transformation. For an impression of the different amounts of voxels in each ROI refer to Figure 3.6.

What is prominent for all voxel correlation distributions is that they are very broadly scattered. This is due to the fact that not all voxels in a ROI can be explained equally well by any model. Especially in the CORTEX distribution it is noticeable that the meaningfully explainable voxels reside in a smaller distribution of outliers at the top. This however matches our expectation since we would only expect subsets of features to match the function of a subset of voxels. In Figure 3.5 it is also noticeable that the Motion Energy model is capable of explaining some of the early visual cortex voxels very well, with correlations up to 0.85.

We also see that the Motion Energy model generally outperforms the K-means and Semantic features in the early visual cortex. In higher visual areas the K-means layers are often close to the performance of the Motion Energy model, while the Semantic features outperform both in some areas (e.g. FFA and EBA). In all other higher visual areas all models show similar correlation distributions.

In the lower visual areas the distributions of the well-explaining models are generally skewed towards high correlations, i.e. a higher proportion of voxels can be explained very well. This can not be seen in the higher visual areas: While the distributions of EBA, FBA, MTp, OFA and LO are skewed towards moderate correlations, those of e.g. FEF and IPS are skewed towards near meaninglessness.

The progression of the median correlation when going deeper in the K-means hierarchy shows different trends for different ROIs. For instance, there is a pure downwards trend in V3B and VO; a downwards trend from Layer 1 and then equal predictive power of higher layers in V2 and V3; and a Layer 3 upwards trend after a decline in EBA, IPS, MTp, FFA and pSTS. Pure downwards trends could either mean that the predictive power of the higher levels generally declines or that there is a specialization to a smaller subset of voxels in the higher layers.

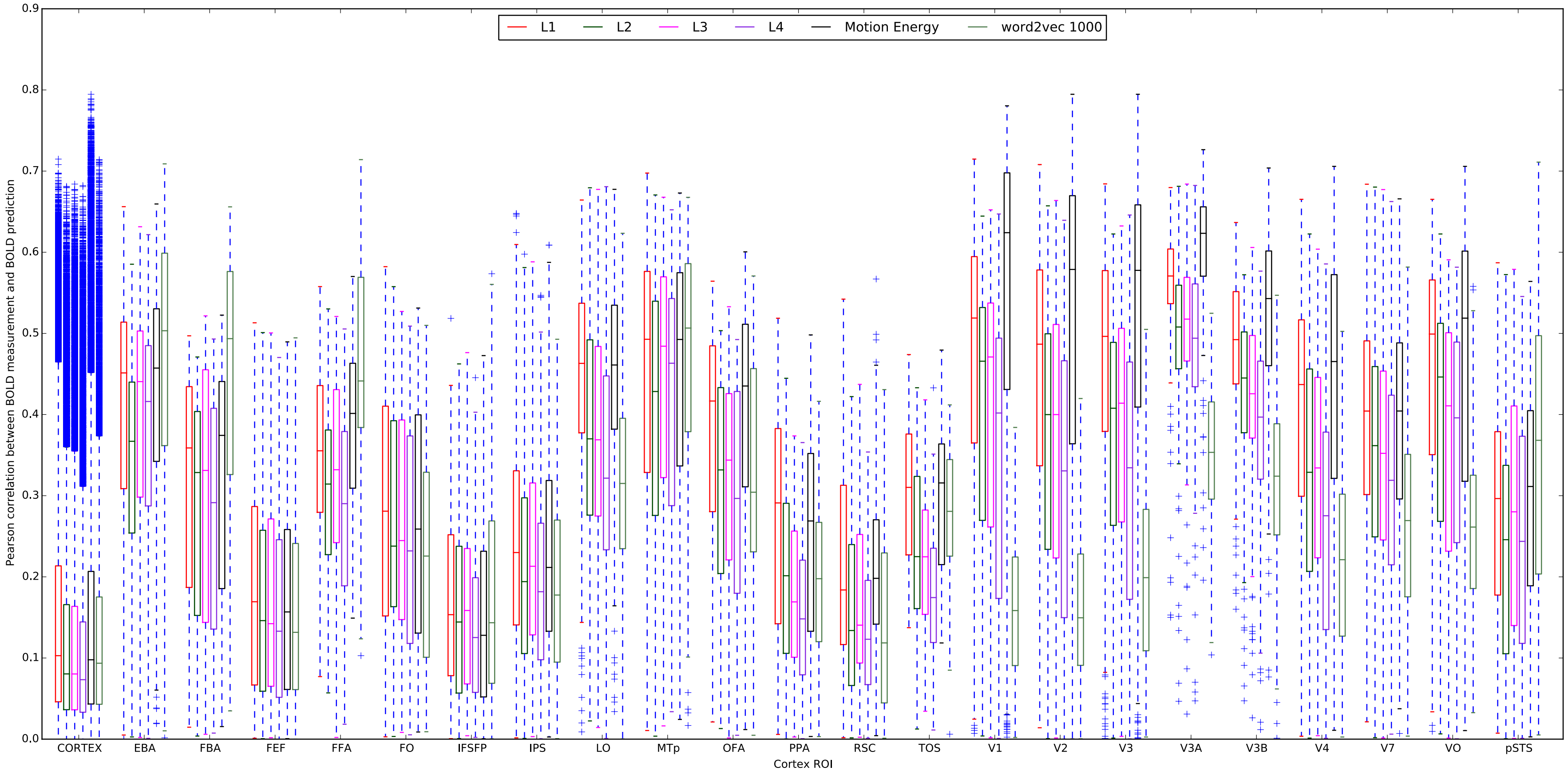


Figure 3.5: Absolute correlation values for the K-means hierarchy layers, the Motion Energy and the Semantic features. For each voxel distribution this shows the median, the interquartile range, the upper and lower quartiles and outliers.

Strong downwards trends occur in the lower visual cortex areas. The upwards trends in Layer 3 occur in intermediate and higher areas. For instance, EBA is possibly involved in the perception of human body parts and form, while pSTS showed activity e.g. during perceiving human body motion and faces [Vangeneugden et al., 2014].

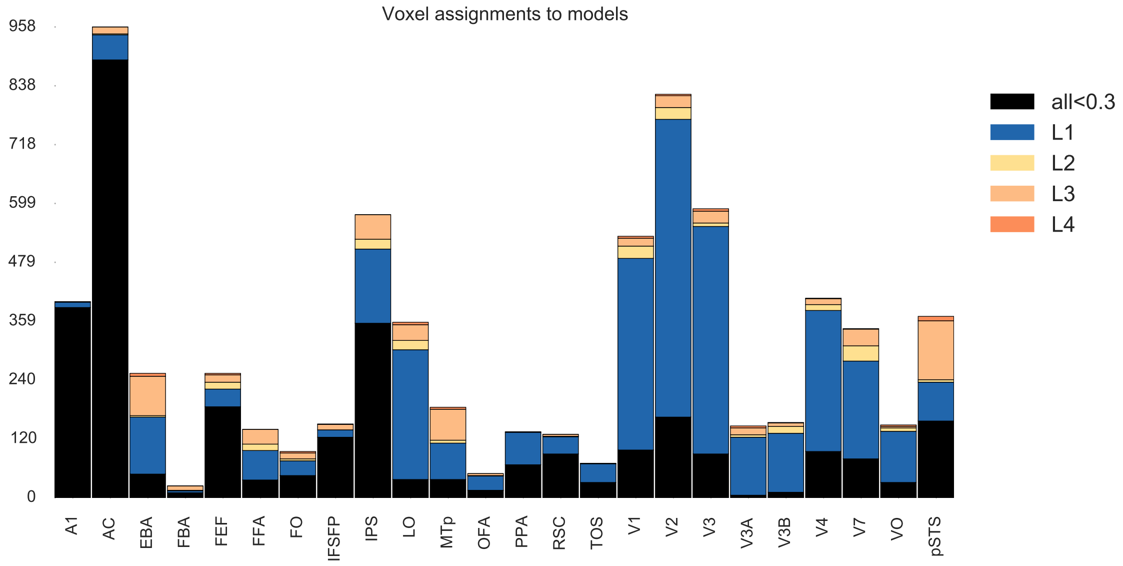


Figure 3.6: Voxel assignments to different layers for moderate and higher correlations, based on which layer's features result in the best prediction-activity correlation. The black area represents the number of voxels where every layer's predictions result in correlations lower than 0.3.

Focusing only on the K-means hierarchy, we would like to know whether higher layers cover previously unexplained voxels, or whether they are capable of explaining certain voxel regions better. Figure 3.6 shows how the voxels in visual system ROIs can be assigned to each of the K-means layers. The differences between the best and second-best correlations have been checked for statistical significance (95% confidence interval). The black area contains voxels with lower than moderate correlations (lower than 0.3). The auditory areas A1 and AC are included to allow comparison to areas not related to the visual system (i.e. unexplainable ROIs). It is obvious that the Layer 1 features always cover most of the voxels in every ROI except pSTS. However for the higher visual cortex areas, especially for pSTS, EBA, MTp and IPS this assignment analysis shows that Layer 3 is capable of explaining a moderate amount of voxels better than Layer 1. These areas also show upwards trends for Layer 3 in Figure 3.5. The improvement in coverage reached by Layer 3 also matches the comparative correlation results in section 3.1.

For an illustration of the actual distributions for some ROIs, refer to Figure 3.7. For comparability these correlations have again been transformed to an interval-scaled distribution via Fisher's Z-transformation. What could be seen in the statistics from Figure 3.5 is also obvious here: Layer 1 is capable of explaining the lower visual cortex area V3 better and with generally higher prediction-activity correlations. The higher visual area pSTS overall has lower correlations, but Layer 3 can explain the higher-correlating voxels slightly better. In area IPS, for which higher K-means layers can predict a larger proportion of the voxels better, the prediction-activity correlation is visibly skewed towards low correlations.

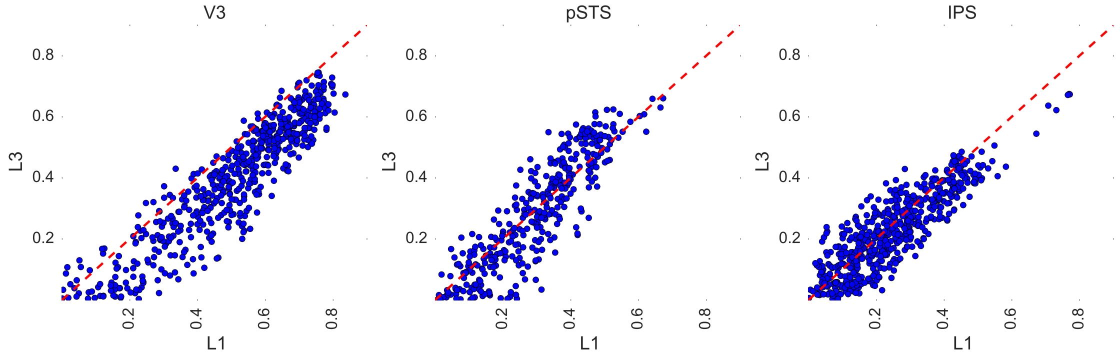


Figure 3.7: Example distributions of prediction-activity correlations for K-means Layers 1 and 3.

3.3 RDE noise for specific feature sets

The univariate ridge regression models are likely influenced by training parameters such as the sampling size of the training data and the actual training procedure. We would like to validate the ridge regression results from sections 3.1 and 3.2 with another procedure that does not include training a model. For this we decided for estimating noise *in the light of a specific feature set* based on Relevant Dimensionality Estimation. Refer to section 2.3.3 for a description of the details. Note that we have preserved the mass-univariate structure of the learning problem.

The noise levels are defined as the squared mean PCA coefficients $z_i \forall i > d_\epsilon$, with $d_\epsilon = d + 5$, where d is the relevant dimensionality for a specific voxel. We decided to add $\epsilon = 5$ in order to stabilize the average noise result by taking the average slightly above the actual relevant dimensionality, avoiding the first coefficients that may still contain signal components not covered by the Two-Component Model.

Figure 3.8 shows the noise values of all voxels in a cortical flatmap, averaged for the first three K-means layers. For a clear visualization, the average noise values were inverted so that voxels with high noise values appear within the blue color spectrum. We can see that for most voxels in the early visual cortex there is a lower noise level than for the higher regions, with small sections showing very low noise. We also can see that some regions that could be predicted well before in the correlation flatmaps show low noise levels in light of the K-means feature sets (e.g. V3A, EBA and pSTS). In general, the previous correlation distribution is reflected in light of the K-means features.

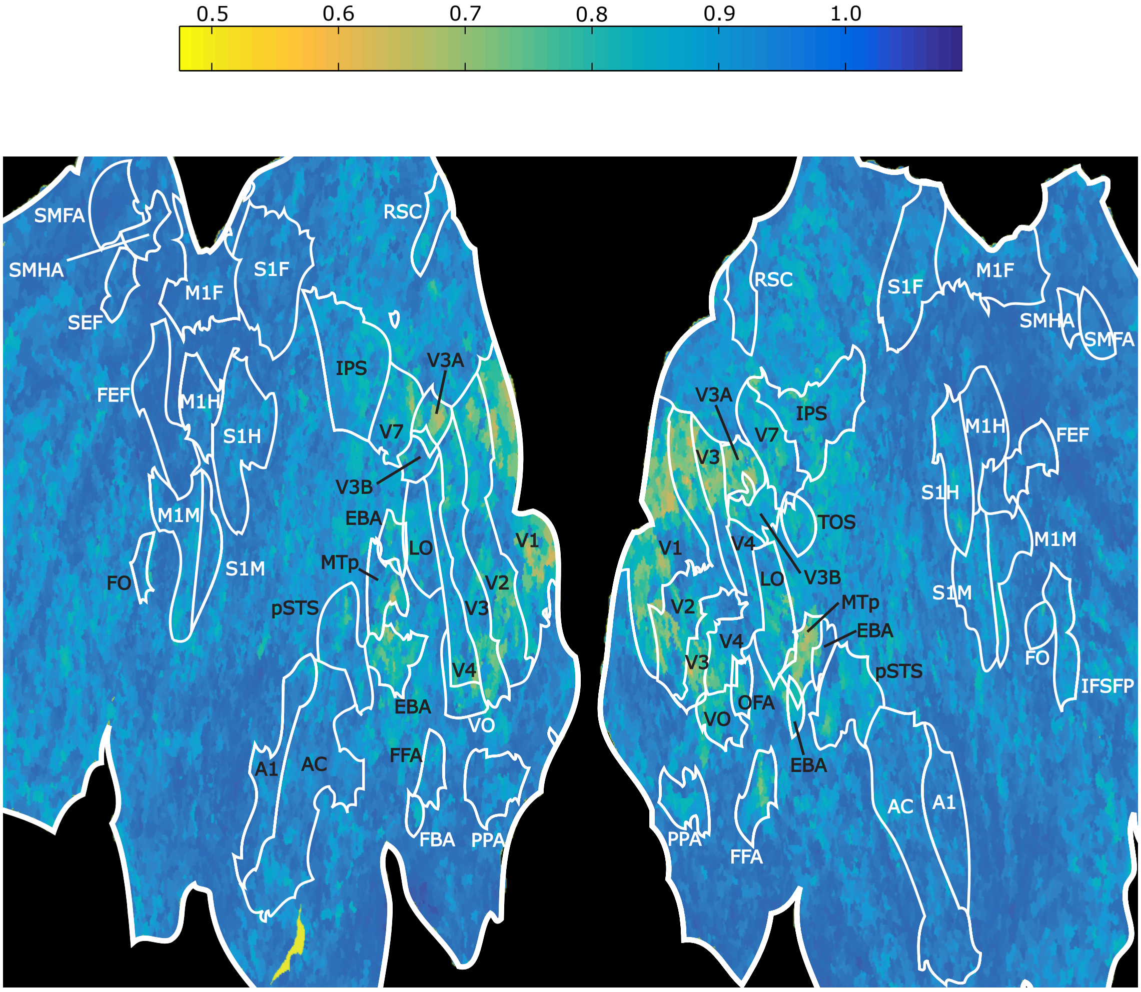


Figure 3.8: Mean inverted noise levels for K-means layers 1-3 on cortical maps. Areas with low noise will appear on the yellow part of the spectrum.

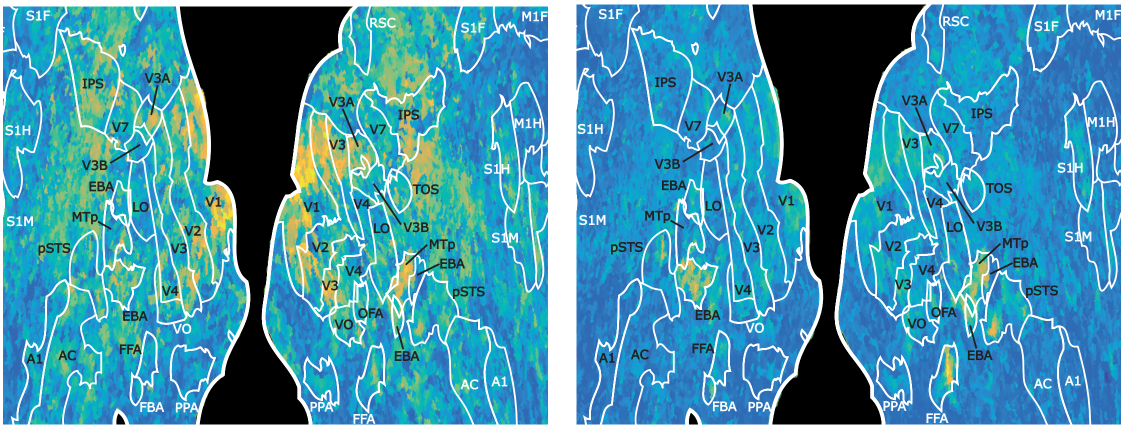


Figure 3.9: Inverted noise levels for motion energy and Semantic features on cortical maps. Areas with low noise will appear on the yellow part of the spectrum.

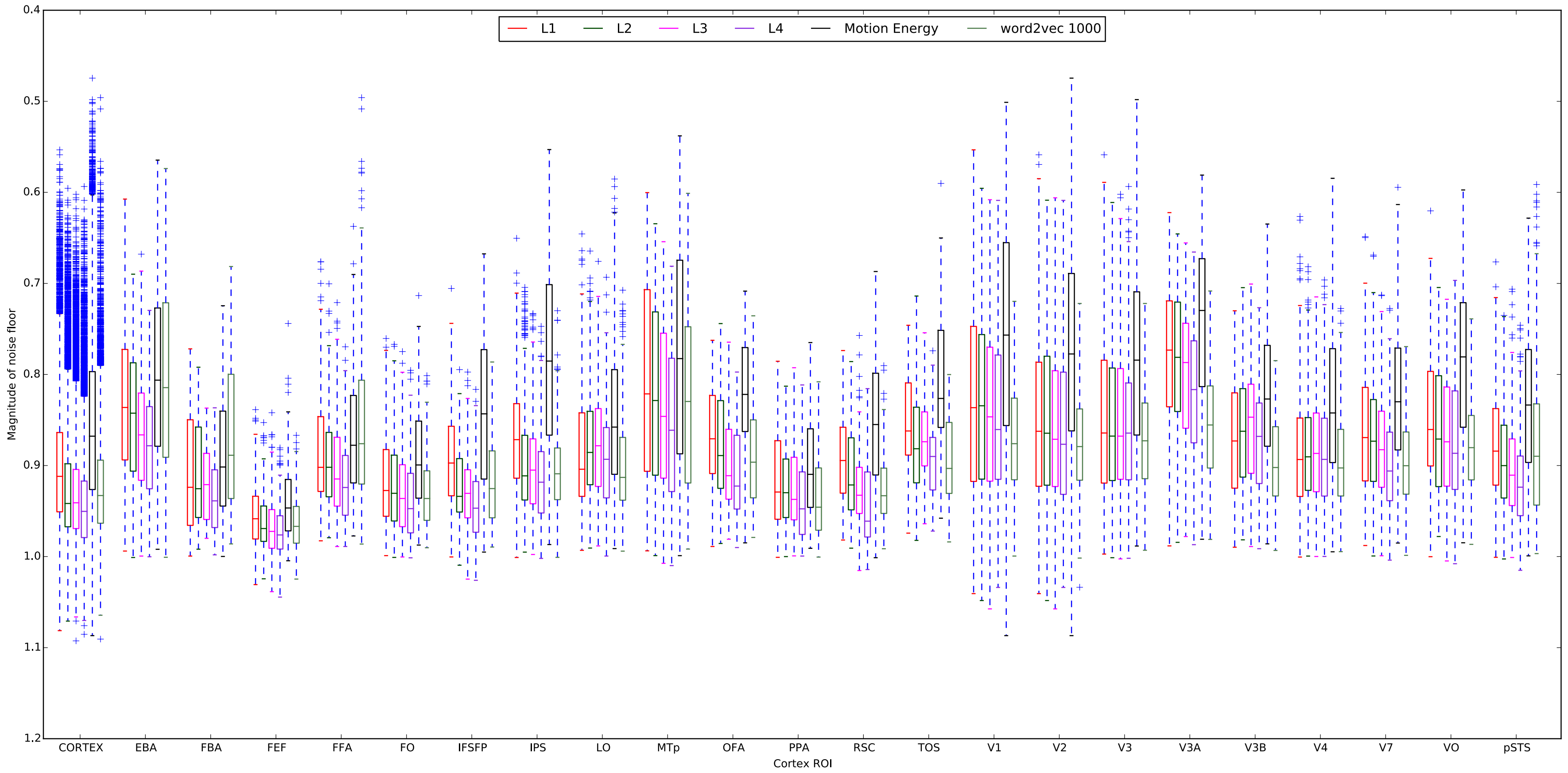


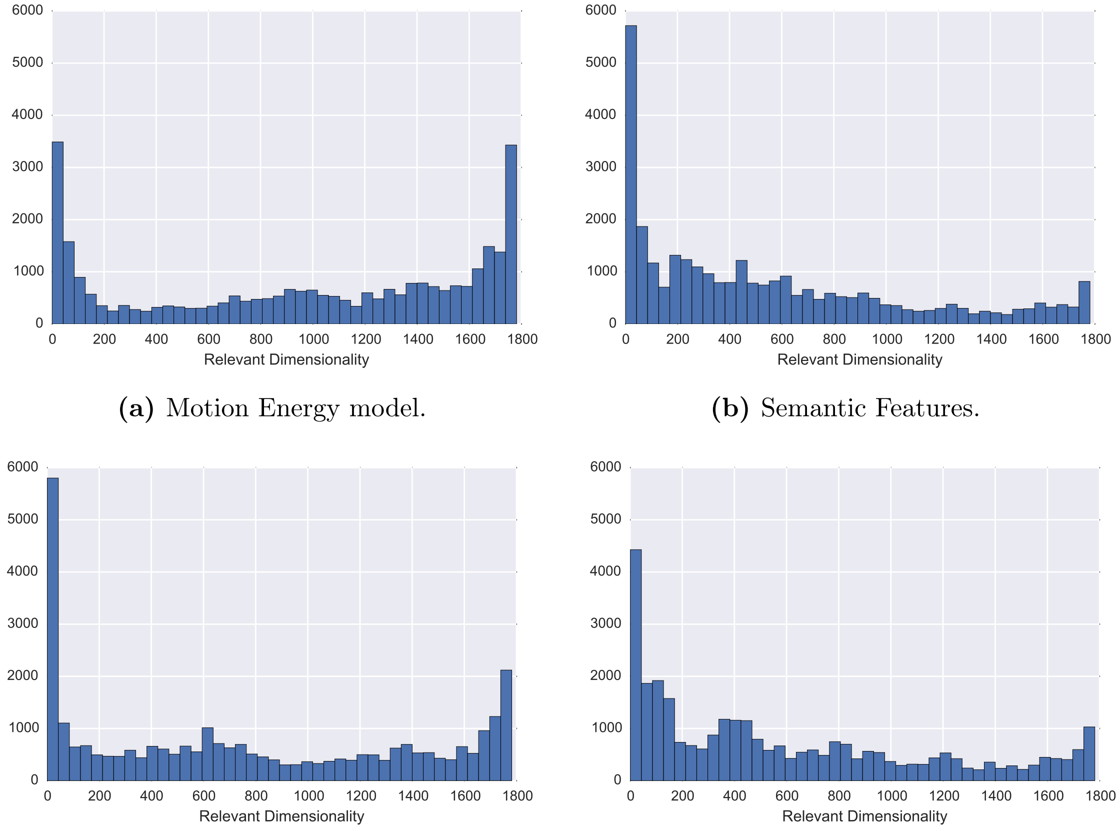
Figure 3.10: Noise levels for all voxels in light of the K-means hierarchy layers, the Motion Energy and Semantic features. For each voxel distribution this shows the median, the interquartile range, the upper and lower quartiles and outliers. Note that the noise axis is inverted.

As a comparison, Figure 3.9 shows the noise values in light of the Motion Energy and Semantic features. Here we notice distinct low noise regions when comparing the flatmaps of the two models. In the early visual cortex it is noticeable how there is less noise in light of the Motion Energy features, while the Semantic features cover higher regions. In comparison to the K-means features, in light of the Motion Energy features substantially less noise is present over the visual system ROIs. Seeing the Motion Energy noise distribution with many low noise voxels we can also assume that the predictive power of the state-of-the art model is superior to that of the K-means features.

In Figure 3.10 we compare the noise levels for specific ROIs. Note that the noise axis has been inverted so that voxels with the lowest noise levels are at the top. The voxels in all ROIs have a similar maximum noise around 1.0.

Similar to the correlations from Figure 3.5, the Motion Energy features generally show their lowest noise in the early visual cortex areas, while the Semantic features show lower noise in some of the intermediate and higher areas. The division of the predictive power into early and intermediate / higher visual cortex areas that we saw in 3.5 are reflected in the noise for the reference models. While this divide (that reflects the correlation results) in principle also exists in light of the K-means features, when following the noise median trends we see different distributions than when following the correlations. Going deeper into the K-means hierarchy, the noise rises or stays at the same level in all areas except V3B, TOS and LO.

Relevant Dimensionalities We also would like to compare the relevant dimensionalities TCM has identified. With their magnitude we can analyse whether, through the lens of a specific feature set, the voxel signal prediction is seen as a complex problem. In Figure 3.11 we can see that both the Motion Energy model and K-means Layer 1 lead to a majority of voxels explainable with low relevant dimensionalities. The Motion Energy model also divides the voxels into two sets, one that can be explained with a low dimensionality and another that is either too complex or unexplainable with its features. In general, these two feature sets appear suitable for modelling the responses of specific voxels, which is what we would expect from an encoding model. In comparison, for K-means Layer 3 and the Semantic features the necessary dimensionalities are also low for a subset of the voxels. However, in the first few histogram bins we also see that a lot of voxels are only explainable with more complex dimensionalities. There is no divide between the voxel sets that is similar to that of K-means Layer 1 and the Motion Energy features. We know from the correlation results before that the Semantic features can generally explain voxels in intermediate and higher order visual areas better, while lower visual cortex voxels have lower or no correlations with the predictions of the linear model. Therefore we would expect that for the Semantic features the voxels with higher Relevant Dimensionalities reside in the lower visual areas. K-means Layer 3 however also explains many voxels in the lower visual cortex areas, while not being able to explain those areas that the Semantic features



(a) Motion Energy model.

(b) Semantic Features.

(c) K-means Layer 1.

(d) K-means Layer 3.

Figure 3.11: Histograms of Relevant Dimensionality for all recorded voxels and for different feature sets.

can explain. Fewer low relevant dimensionalities for Layer 3 are another sign for its lower predictive power.

Figure 3.12 shows the mean cumulative sum over all z_m-based squared noise levels in all visual ROI voxels. A constant noise floor is reached as soon as the graphs become linear with a constant slope. Before arriving at this constant noise level, the mean cumulative sums show differences in their gradients. The models with the largest mean coefficients z_m for the early d – here Motion Energy and K-means Layer 1 – are those that show the steepest gradient. For them either the correlation of the BOLD signal y with the principal components (expressed by the coefficients for small d) is larger than for other models, or the relevant dimensionality is small and they arrive at the constant noise level faster. In either case the high average levels of the cumulative sums for these two models indicate that large amounts of voxels in the visual system show the steep gradient. It also indicates that their average predictive power for visual system voxels is likely higher than that of the other models.

In comparison to K-means Layer 2 and Layer 3, the Semantic features also show a steep slope for small d, however they lead to lower mean coefficient levels than e.g. the Motion Energy features. This likely reflects how they can mainly explain a smaller amount of

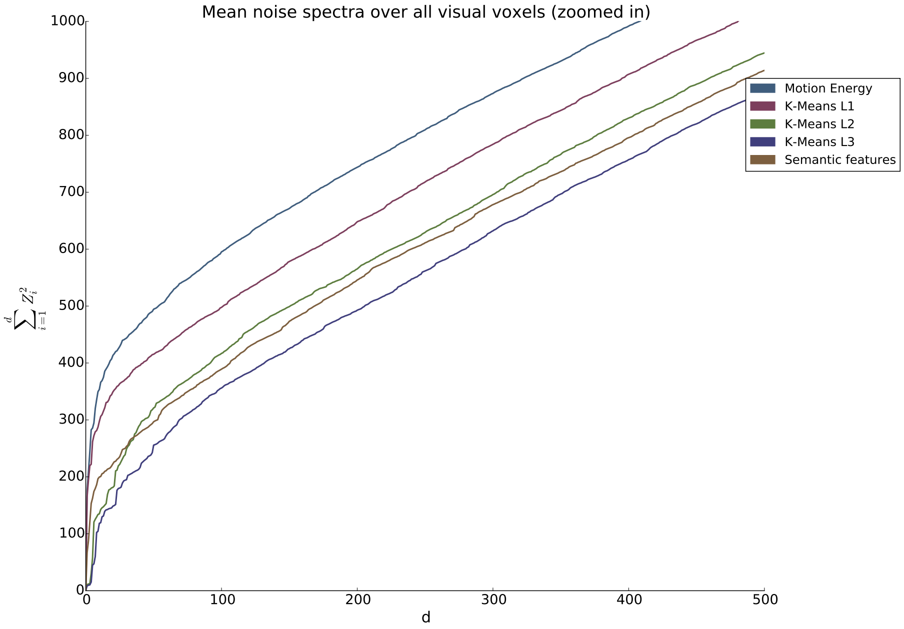


Figure 3.12: Cumulative noise spectra for all models over all visual ROIs. This shows the mean cumulative sum over the squared coefficients for a sub-selection of *d*. Note that this does not visualize the high variance for these values among the voxels in the mass-univariate RDE.

voxels well (i.e. the higher visual system areas). This might also be the case for Layer 2 and 3, but their smaller slope at the beginning indicates that their relevant dimension is high or that the principal components of their Kernel matrices don't correlate well (and early) with the voxel-wise BOLD signal.

3.4 RDMs for specific ROIs and models

Figure 3.13 shows the RDMs for the models K-means Layer 1, Motion Energy and Semantic features, as well as exemplary representations in specific ROIs. To get an impression of the contents of the stimulus data at a subselection of the analysed time points, see Figure 3.14. RDMs can only be provided for the resampled test set. Noise dominated the visualization when using the single recording sample of the full training set. The RDMs in this section are used for visualizing the representations and for their qualitative discussion. While it would be possible to provide quantitative results by correlating RDMs with each other, it is unclear whether this is a valid criterion. We therefore decided to exclude quantitative results from the discussion.

Overall the method has lead to heterogeneous representations within the different ROIs and feature sets, which can also be seen from Figure 3.13. The strong dissimilarity among the Semantic features is due to the manual labelling of the diverse objects appearing in the scenes. Figure 3.15 shows RDMs for Layer 1 and Layer 3 of the K-means model. The

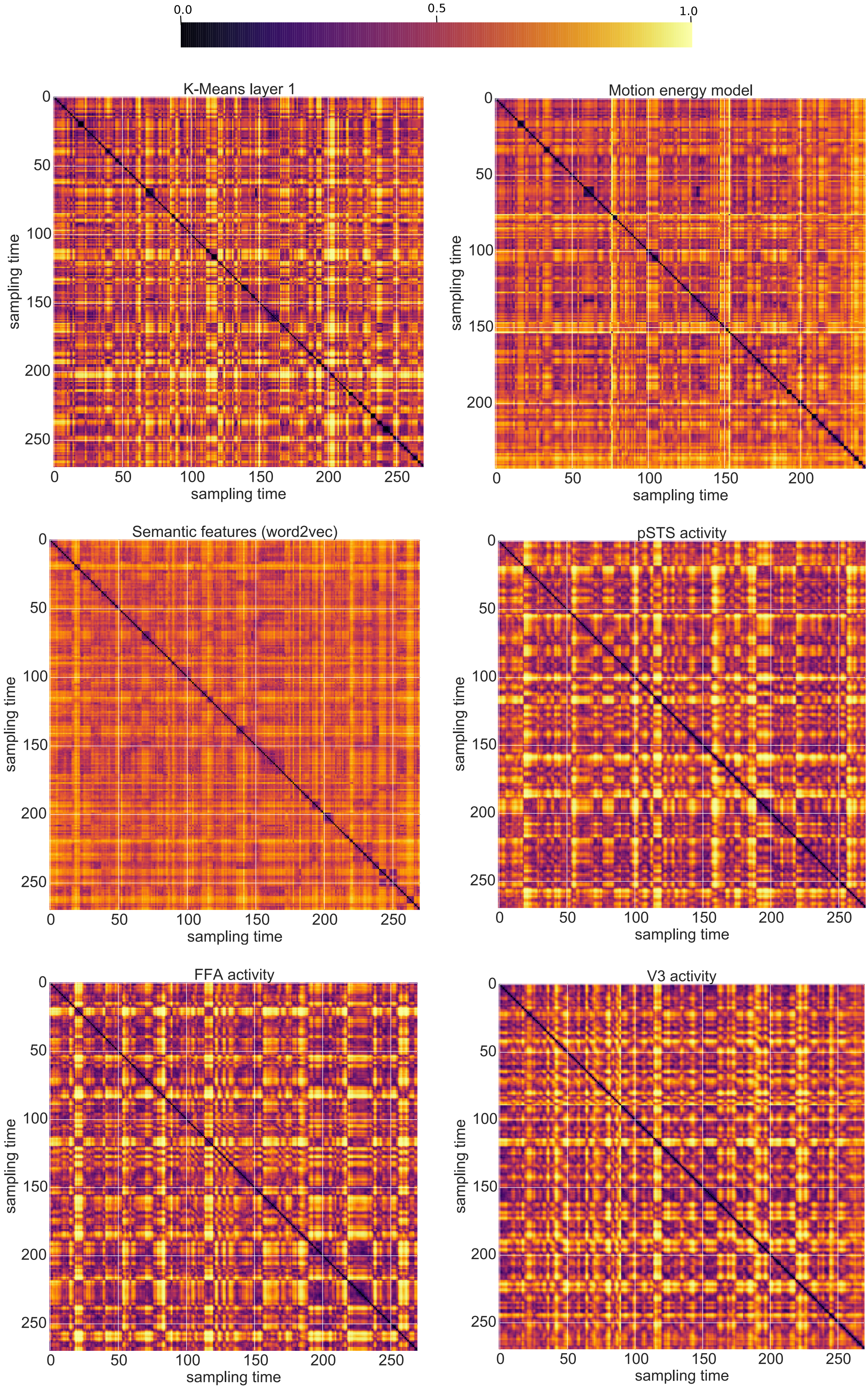


Figure 3.13: Representational Dissimilarity in the validation sets for two feature sets and four ROIs. 0.0 is similar and 1.0 dissimilar.

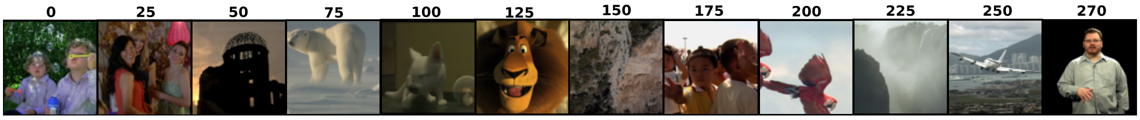


Figure 3.14: Exemplary content of the validation set presented during a subselection of sampling points.

higher level training either leads to an overall loss of information or to an intensification of some detected properties (or both). It is possible that what is detected in the higher levels is a differentiation between animate and inanimate objects, or moving and static scenes respectively.

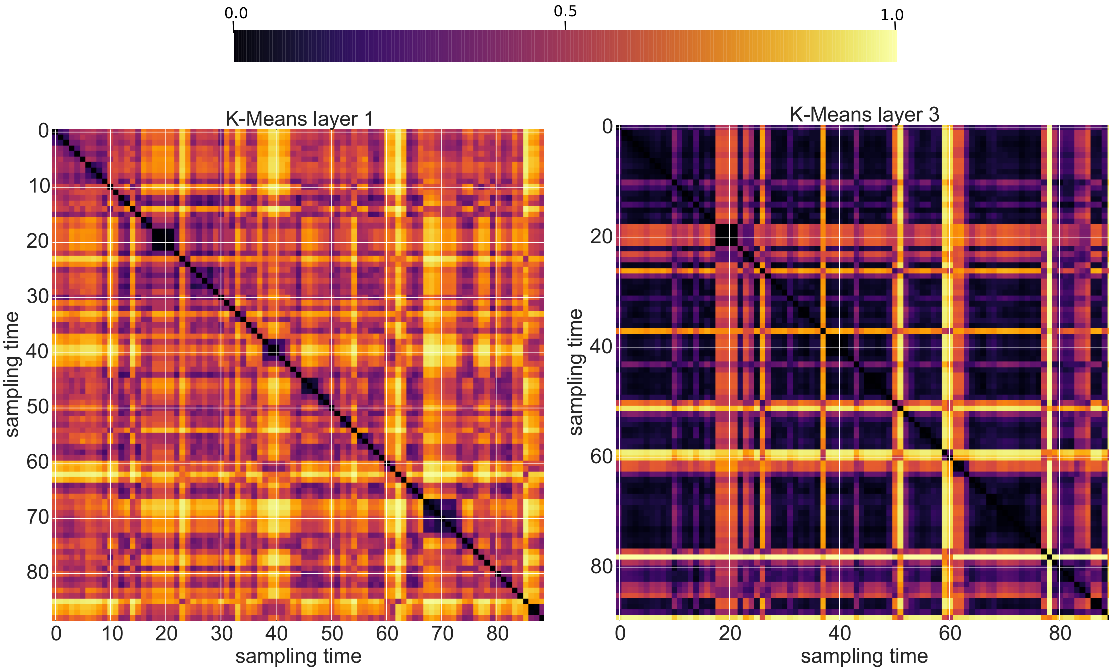


Figure 3.15: Representational dissimilarity for L1 and L3 for the first recording session of the test data.

We explore this more deeply by exemplary investigation of the content of scenes in the first recording session from Figure 3.15. The most prominent feature in the K-means RDMs are highly dissimilar sampling points, visible as coherently bright stripes. These also occur across the Motion Energy model and in many ROIs, such as in LO and FFA. Looking at the stimulus content in, for instance, sampling point 20, we can see that it consists of an urban beach scene, where there is movement only in the lower half of the frames. Around 60, a scene of a free climber falling into an abyss was presented, with large red or black patches. These scenes have similar representations in both Layer 1 and Layer 3. At 50, Layer 3 detects a scene highly dissimilar to all other stimuli. In the stimulus data this is a short static sequence of the Hiroshima Peace Memorial silhouette at sunset. Within the test set, these dissimilar areas are also unique in their structural organization, since most scenes show more feature-rich content across the subpatches than those large coherently coloured frames presented here. Layer 3 appears to mainly differentiate between scenes with overall rich features and those with coherent patches, and scenes with widely spread rich movement and those that only contain movement in subpatches.

For investigating similar patches, we would like to focus on sampling points where a coherently dissimilar feature set is interrupted by sections of similarity. We first notice that these are mainly picked up by Layer 1. Around 40, there is a long scene of a convoy moving towards the static camera, where the motion mainly consists of flags slowly moving in the wind. A feature set with close distance to this can be found at 28, which shows a group of women talking, also recorded with a static camera. Another feature set similar to the one at 40 is a recording of a polar landscape occurring around 73, over which the camera is panning slowly. The scene at 28 continues with frequent cuts to other scenes of persons. Similar to its representation is 58, which shows a climbing scene with frequent cuts. The actual image content of all these scenes is highly dissimilar to each other, which is possibly reflected in the distances all being above 0.3 in the similarity spectra. Layer 1 appears to detect motion differences, differences in the actual image content, and frequent cuts.

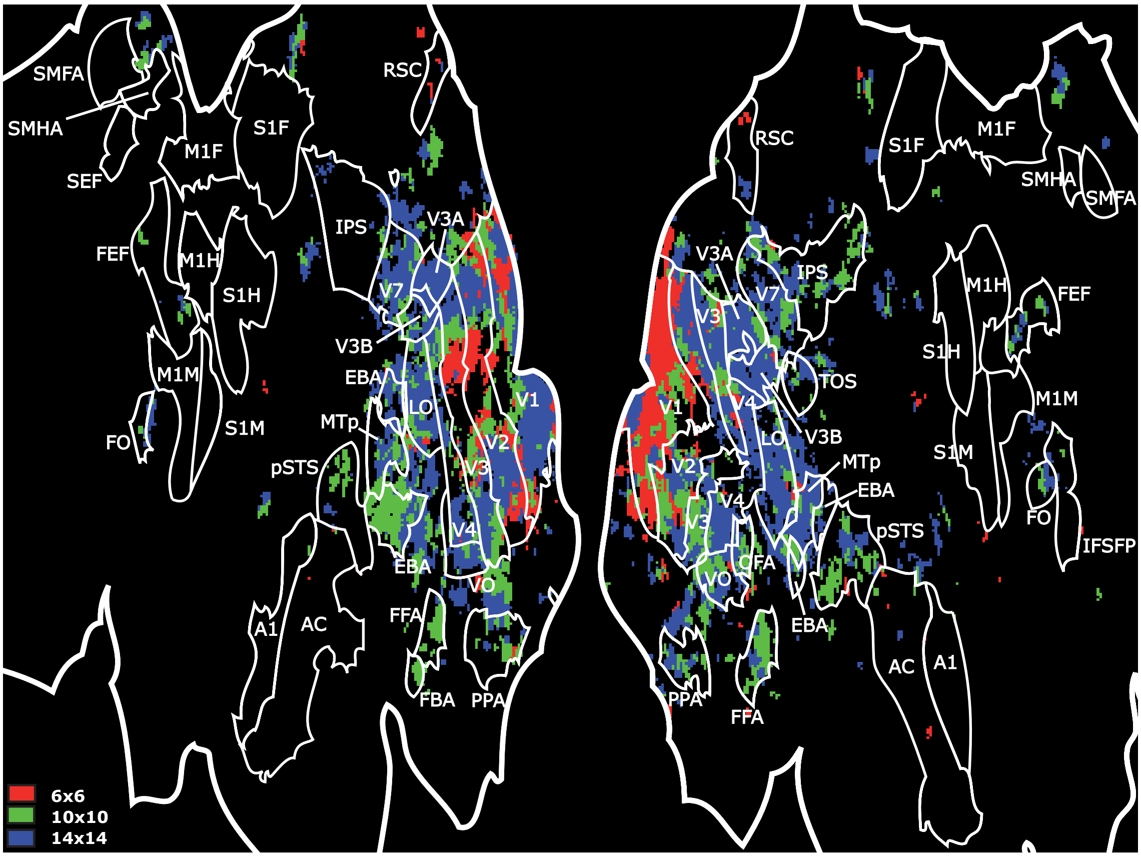


Figure 3.16: Distribution of scale-wise feature sets that account for maximum predictability per voxel. Only voxels with correlations > 0.4 are shown.

3.5 Notes on other training outcomes

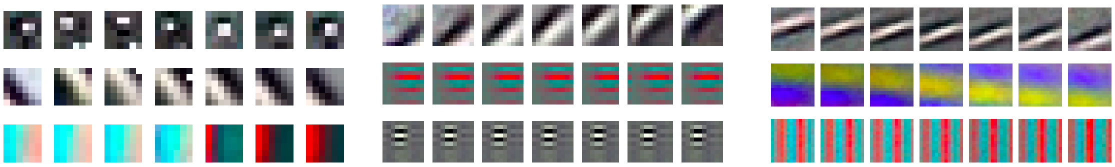
This section shortly illustrates the outcome of the unsupervised learning process with example prototypes and receptive fields. It also shows how the three K-means Layer 1 centroid sizes are distributed over the early visual cortex.

3.5.1 Scale-wise predictions in layer 1

Here we shortly analyse whether the differently sized prototypes from Layer 1 also lead to different predictability distributions among the voxels. Figure 3.16 shows how the three scales $S6 \times S6 \times T7$, $S10 \times S10 \times T7$ and $S14 \times S14 \times T7$ are distributed over the well-predictable cortical areas: A mass-univariate ridge regression model has been trained with the feature sets of each scale. The figure shows for every voxel which of the feature sets leads to the best prediction for our single subject.

Note that while these distributions appear to exhibit structure, the differences between the distributions are small nevertheless. For our single subject, the differences between the distributions of voxel wise-correlations $S6 \times S6 \times T7$ / $S10 \times S10 \times T7$ and $S6 \times S6 \times T7$ / $S14 \times S14 \times T7$ are significant with $p \ll 0.05$. The difference between distributions $S10 \times S10 \times T7$ / $S14 \times S14 \times T7$ is not significant ($p = 0.0733$). Choosing a larger patch size than $S14 \times S14 \times T7$ might have been reasonable here, but was avoided due to the increased computational load during feature extraction.

3.5.2 Prototypes



(a) $S6 \times S6 \times T7$ (b) $S10 \times S10 \times T7$ (c) $S14 \times S14 \times T7$

Figure 3.17: Example filters for all prototype sizes in K-means layer 1. The actual patch sizes do not match these print sizes.

Figures 3.17a, 3.17b, 3.17c show example centroids learned for the three spatial scales on the temporal RGB patches. The centroids were selected as to show specific properties often occurring when learning in this unsupervised fashion, but do not reflect the overall distribution of the filters. Multiple runs of the K-means based prototype learning resulted in different centroid arrangements and some difference in centroids. Over multiple training rounds no major change in model performance could be seen, but many similar or equal centroids could be observed.

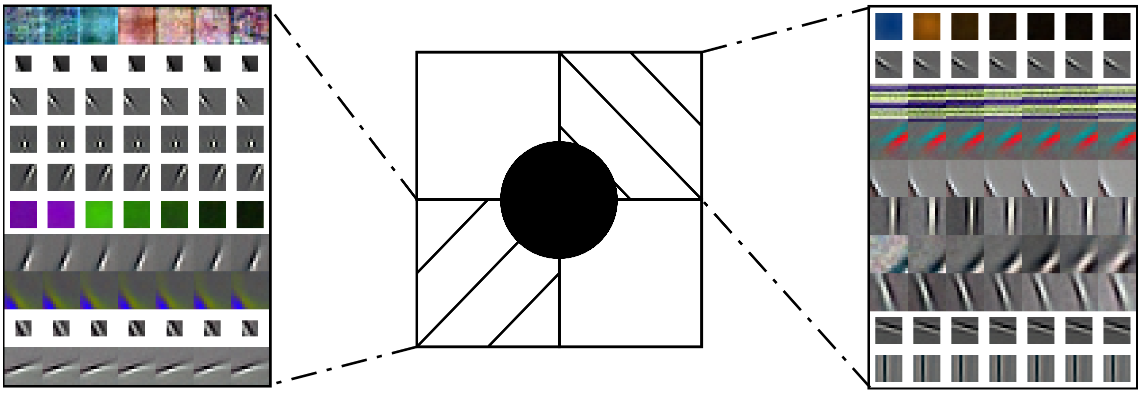


Figure 3.18: Two examples for learned receptive fields in Layer 2. Shown are the top 10 correlating centroids of the receptive fields. Note that receptive fields are learned over all scales.

3.5.3 Receptive fields

Figure 3.18 shows the top correlated features in two example receptive fields of the RN = 4 × 20 learned receptive fields in Layer 2. Note that in Layer 2 we combined each outer pooling region with the fovea pooling region for learning the receptive fields. The learning process was described in section 2.2.6.

4 Further Discussion

In this section we will discuss biological interpretations, possible limitations of the K-means model and also would like to illustrate future potential of the deep unsupervised encoding approach.

4.1 Biological interpretation

When developing an encoding model for the visual system, it is desirable that the various steps are backed up with knowledge about biological functions. The K-means model features a couple of biologically properties that we would like to briefly describe in this section.

Competitive Hebbian Learning Prototype learning with K-means can be seen and expressed as competitive Hebbian learning. Using the simplified description for Hebbian learning, we can see the cluster assignment as *fire together* and the cluster relocation as *wire together*. The winner-takes-all aspect in prototype learning is the competitive property.

Whitening As mentioned before, the whitening transformation of the input patches is crucial for the classification performance of this soft K-means variant (shown in [Coates et al., 2011], proof in [Vinnikov and Shalev-Shwartz, 2014]). In research about the visual system, there is belief and experimental evidence that the center-surround cell structure in the retina is adapted to the statistical properties of natural scenes and carries out a whitening transformation before forwarding the visual information (now less redundant) into the LGN and the occipital lobe [Daniel J. Graham, 2006].

Neural network implementation It is possible to implement K-means in a neural network. For the specific variant of soft K-means used in this thesis, see [Pehlevan and Chklovskii, 2015].

Simplicity If evolution has found and implemented algorithms in biological systems, then it is likely that these algorithms rely on large-scale effects emerging from the interplay of small components. This interplay should be robust against external perturbations and not highly vulnerable to the chaotic properties of dynamical systems. K-means is a very basic algorithm, essentially relying on two simple steps. We propose that the possibility of the biological implementation of variants of it should be considered and further studied.

Although these properties exist it should be noted that the K-means model can at most represent an oversimplified version of realistic neural processes. Nevertheless, discussions like this may contribute to the search for a general learning algorithm in natural learning systems like the brain.

4.2 Limitations of K-means-based deep encoding

We briefly discuss the problems and limitations we see in using the presented K-means model as an encoding model.

Applicability for videos The original model by [Coates et al., 2011] was studied on datasets of static images, but not on video datasets. The extension of visual feature learning to the temporal dimension is not straight-forward, and most efficient video feature learning networks have been published only recently (e.g. at NIPS 2014 with [Donahue et al., 2014]). Better temporal modelling can be achieved with e.g. LSTM blocks, which we had not considered when developing the model.

Pooling strategy The summation pooling over the visual field in the first K-means layer uses only 5 regions. Higher numbers or a block-grid design as in [Nishimoto et al., 2011] would quickly lead to extreme feature dimensions since every new region increases feature dimension by K. In [Coates et al., 2011] 4 pooling regions had been used. Due to the spatial representation of visual information in the cortex, more fine-grained pooling fields are advantageous for visual encoding models. We consider the lack of them a huge downside of our model.

Initial clusters The prototypes in the K-means routine were initialized with a random data point in [Coates et al., 2011]. While in the beginning we shortly tested other routines, we continued with their original initialization in the course of the project since we could not observe major changes in the predictive power. However, since one of the well-known downsides of K-means is the stability of the prototype solutions, if this model should be studied further one should explore whether different initialization routines influence the predictive power in detail.

Category coverage Our model does not explain most of the higher-level voxels that are explained by the Semantic features. It is questionable whether the description of categories (or ROIs covering object categories such as FFA) can be achieved with a purely unsupervised model. We had decided to keep the complexity low and try the purely unsupervised approach of [Coates et al., 2011] and [Coates and Ng, 2012]. Nevertheless, possible extensions include adding a supervised stage or a layer where unsupervised category learning occurs.

Encoding model selection The parameters KN and R showed influence on the predictive performance of the model. Therefore a grid search for these parameters had been performed both with using the same KN and R for each level and with their layer-wise selection based on their predictive performance. We discontinued the layer-wise parameter selection when it became apparent that an informed choice could not be made valid from a machine learning point of view. As an example, during the layer-wise parameter selection one could have selected the parameters so that the number of newly explained voxels is maximized. This choice, however, would have relied on a performance measurement acquired from the test set, which would have lead into overfitting and would have violated the idea behind using such a test set. Nonetheless layer-wise selection of the parameters would be preferable, and should be used, if a valid selection routine can be found.

4.3 General Limitations

Small number of subjects Single-subject studies don't intend to be representative for a general public. A fully developed individual brain should be analysed, avoiding new problems introduced in the preprocessing routines of multi-subjects studies in fMRI. This of course has the consequence that the value of a scientific statement may become arguable, and that only a limited number of subjects can be studied comparatively in a single publication.

Dataset reuse The dataset and stimuli from [Nishimoto et al., 2011] recorded by the Gallant lab have already been used for several years by various groups. Certainly, when a large number of research groups working on encoding models rely on the same single-subject datasets, one should view their results with caution until they can be transferred to new datasets. Currently one candidate for a novel large-scale multimodal stimulus dataset is being published at `http://studyforrest.org/`, but so far (August 2015) only the pure audio stimulus recordings are completely available.

4.4 Potential of encoding based on representation learning

An in-depth and general introduction to the advantages and potential of encoding models can be found in [Gallant et al., 2011]. In brief: The most important difference to conventional predictive fMRI studies is that encoding models use a random set of stimuli showing natural statistics, step back from stimulus labels, and extract features with a more generalizable and bottom-up method such as Gabor wavelet-based edge extraction. Based on these features they predict voxel activity in a specific cortical region of interest such as the visual system. Decoding can be done directly by inverting the encoding model. In this section we discuss the potential of *deep encoding models* more specifically.

Spatial organization of hierarchical representation Layer-wise prediction with different feature sets makes it possible to study how complex representations in the brain gradually emerge, e.g. along the ventral and dorsal pathways. The goal would be an association between the features in various layers and voxels over a large cortical map. These associations between visual cortex regions and feature sets could then be further compared to existing knowledge about the visual system. Such visualizations could provide remarkable insight into what certain cortical regions or voxels respond to. To further motivate such efforts it should be mentioned that the *Deconvolutional Network* introduced in [Zeiler et al., 2010] exhibited third-layer representations strongly resembling the *primal sketch tokens* that David Marr hypothesized in his theory about the visual system [Marr, 1982] (see Figure 4.1).

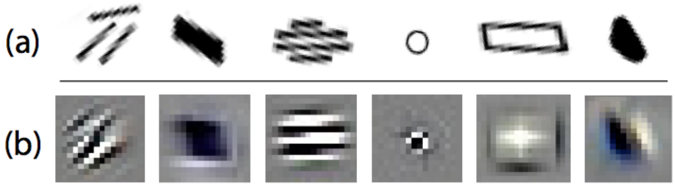


Figure 4.1: Representations in the third layer of a deconvolutional network in comparison to Marr's Place tokens. a) *Place tokens* from Fig. 2-4 of *Vision* by David Marr. b) 3rd-level representations from [Zeiler et al., 2010] (also the illustration source).

First level features are usually straight-forward to visualize. For higher layers, advanced deep neural network visualization techniques are required. Simple weighted correlation-based visualization does not sufficiently explain what is happening in a deep neural network. Such advanced methods are currently being investigated by the machine learning community, with one notable recent example being [Zeiler and Fergus, 2013] who visualized the layers of [Krizhevsky et al., 2012] with what is essentially a gradient ascent [Simonyan et al., 2013]. Also, for the same purpose there has been research into inverting deep neural networks [Mahendran and Vedaldi, 2014], [Dosovitskiy and Brox, 2015]. It is also possible to analyse in detail the relevancy of structures in the input space for the predictions of a specific voxel down to single pixels [Bach et al., 2015].

Perception reconstruction One useful property of encoding models is that a decoding model is basically its inverted form. This is a property making them more powerful than pure decoding models. It is not possible to do the reverse, i.e. it is not possible to derive an encoding model from a decoding model. Since deep encoding models could potentially provide visual descriptions of mid- and higher level processing stages, research in reconstruction may strongly benefit from research in deep neural network visualization. This is currently under investigation in several groups working with deep encoding models. If more features than those represented in the striate cortex can be explained, decoding of imagination and dreams might as well become possible. Perception reconstruction is a promising area of research in itself: One can easily imagine further scientific or therapeutic applications of brain-reading methods. Such reconstruction attempts also simply kindle scientific curiosity as an end in itself. Note that the most advanced dream decoding algorithm so far algorithm could decode dreams up to the level of object categories [Horikawa et al., 2013], and that so far there where no successful attempts to actually *reconstruct* visual contents of dreams. Aiming at reconstructing visually perceived stimuli *perfectly* using the BOLD signal can be viewn as a data-driven engineering approach with a clear reference about whether the model is correct. This also has the potential to lead to more insight into the visual cortex and into properties of the BOLD signal itself.

Deep encoding and deep neural networks The recently investigated deep neural networks usually contain a label-based supervised stage at the end, which functions as their prediction benchmark. Without such class prediction stages, due to the number of parameters basically any representations could be learned during the unsupervised stages. A different approach for improving the prediction performance of a deep neural network while avoiding the complexity of the unsupervised learning outcome could be to carefully fit the unsupervised learning stages to biological processing stages. Such approaches would require a database of large-scale multi-subject data under various conditions. We assume that investigating the capabilities of such a biologically fitted unsupervised hierarchy is promising.

5 Conclusion

During this project, unsupervised features extracted from videos within a K-means-based feature learning hierarchy were used to predict the BOLD signal in human visual areas elicited in response to spatio-temporal stimuli. We could show that with this completely unsupervised learning procedure a functioning encoding model for the lower visual cortex areas can be created. We have seen that, in general, the unsupervised feature learning approach is both capable of providing features that can predict brain activity, and can also serve as a basis for the analysis of features preferred by specific cortical areas. We have verified the ridge regression results with the help of Relevant Dimensionality Estimation and investigated what the higher K-means levels possibly represent with Representational Dissimilarity Matrices. While we presented the visual and quantitative results of a single subject in this thesis, the single-subject data of the other recorded subjects has been analysed within the project's scope and has lead to similar comparative results for 4 of the 5 other subjects.

However, only few new voxels in higher and intermediate ROIs – mainly in pSTS, IPS, MTp and EBA – can be explained by higher layers of the K-means hierarchy. While higher levels identify abstract features, those features are not object categories, but broad and unspecific image features such as strong differences in movement and large homogeneously coloured areas. Also, the hand-crafted Motion Energy model and the manually labelled Semantic features outperform the completely unsupervised feature hierarchy, and are highly superior in the ability to distinguish between lower and higher visual cortex areas. With our K-means model we could not find higher level representations that were similar to the continuous semantic spaces that had been found before.

We expect that the main flaws of the higher K-means hierarchies reside in the pooling strategy that is not rich in detail, in no suitable temporal learning in the higher layers and in a lack of specific approaches to unsupervised category learning.

Generally, further studying deep encoding model approaches with unsupervised components is promising. In principle, in this project we could show that also for spatio-temporal data unsupervised approaches could help understanding the brain's internal language. Further insights into this language have the promising potential to lead to brain reading tools in the future.

A Reference models

A.1 Motion energy model

Motion energy features are one of the standard methods for describing low-level and localized Fourier-like motion perception in the mammalian visual system. They originally were proposed in [Adelson and Bergen, 1985] as a solution for the phase dependency problem that occurs when detecting features with spatio-temporal filters. The model was originally chosen for [Nishimoto et al., 2011] because it has proven to be consistent with a large body of experimental data of the visual system.

In general, motion energy is the *sum of the squared output* of spatiotemporal filters which are 90° out of phase in space. These filters are said to be *in spatial quadrature.* This summation leads to a phase-independent response – see Figure A.1 for a simple illustration. Furthermore, usually a non-linearity is applied on this sum of squares to compress the feature data.

In the non-directional motion energy model from [Nishimoto et al., 2011] used here, the gabor wavelet basis set consists of 6555 3D filters. The filters are created by completely combining sets of six spatial frequencies [0, 2, 4, 8, 16, 32] (measured in cycles per image), three temporal frequencies [0Hz, 2Hz, 4Hz] and eight directions [0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°]. Using all these filters (except of a subset considered unnecessary) would result in the directional motion energy model also discussed in this publications. To get

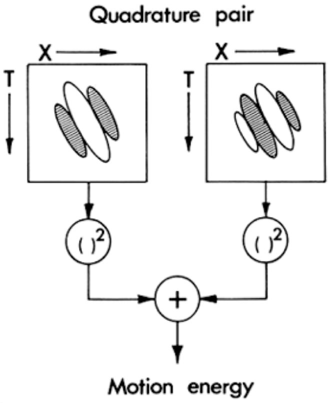


Figure A.1: The sum of quadrature pairs of spatio-temporal filters, resulting in a motion energy scalar. Illustration source: [Adelson and Bergen, 1985].

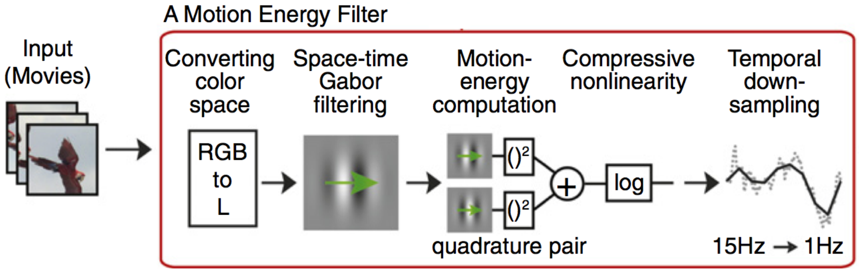


Figure A.2: Complete motion energy feature extraction from [Nishimoto et al., 2011] (also the illustration source).

the non-directional version, the outputs of anti-directional filters (e.g. 180° vs. 0°) are summed at each spatial position, orientation and temporal frequency. Please refer to the supplementary material of [Nishimoto et al., 2011] for full details on the structure of the Gabor Filter Pyramid.

The video stimuli are first passed through a color space conversion into the *CIELAB* colorspace. The model then proceeds to work on the pixel data in the luminance channel. Patches from the video data are taken from a block-grid structure and convolved with all quadrature filter pairs from the filter bank, extracting their motion energy. The block-grid structure is adaptive to the spatial frequency of the filters. Refer to Figure A.2 for an illustration.

Note that as in any model discussed in this project the motion energy features are finally temporally down-sampled from the frequency used in the experiment (15Hz) to the sampling rate of the BOLD signal acquisition (1Hz or 2Hz) by taking the mean over the cycle. Finally each motion energy feature is rescaled to zero mean and unit variance for the machine learning algorithms to work properly.

Complete details for all intermediate steps can be found in [Nishimoto et al., 2011] and especially in its supplementary material. In practical instructions, the motion energy feature extraction can be described as follows:

1. Create the described set of spatio-temporal 3D Gabor filter pairs. Pairs should be in spatial quadrature, i.e. such that one filter is zero when the other one is at maximum.
2. Extract the luminance channel from the color-converted original video data.
3. Convolve all filters with the video stimuli's luminance channels. This is done by taking patches from a grid placed over the stimuli with all filters.
4. Square the convolved responses of the spatial quadrature filters. Take the complete sum over the two resulting vectors.
5. Sum the complete sums of the spatial quadrature filters.
6. Pass the final sum through the log-transformation (the static non-linearity) to obtain the final motion energy feature.

7. Rescale each motion energy feature with a z-transformation to let the feature reside within zero mean and unit variance.

During this project we had the opportunity to access and experiment with the original motion energy model code from [Nishimoto et al., 2011].

A.2 Semantic features

The Semantic features we use are not identical to the WordNet labels from [Huth et al., 2012]. We instead used a set of 1000 labels that was created semi-automatically based on a `word2vec` vector space trained on the Japanese Wikipedia corpus. For every one-second movie clip, 5 Japanese students were asked to annotate them in their native language. These labels were then projected into the `word2vec` vector space. The final `word2vec` representation for each 1-second clip was created by averaging over the 5 representation vectors. In essence this is similar to the `WordNet` labels from [Huth et al., 2012], however it uses a continuous vector space for representation. Note that this feature set is in a preliminary state and was acquired through personal communication.

List of Figures

List of Tables

Bibliography

[Adelson and Bergen, 1985] Adelson, E. H. and Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299.

[Agrawal et al., 2014] Agrawal, P., Stansbury, D., Malik, J., and Gallant, J. L. (2014). Pixels to Voxels: Modeling Visual Representation in the Human Brain. *arXiv preprint arXiv:1407.5104*.

[Bach et al., 2015] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7).

[Bengio et al., 2013] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828.

[Bishop et al., 2006] Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 1. Springer.

[Blakemore and Cooper, 1970] Blakemore, C. and Cooper, G. F. (1970). Development of the brain depends on the visual environment. *Nature*, 228:477 – 478.

[Braun et al., 2008] Braun, M., Buhmann, J., and Müller, K.-R. (2008). On Relevant Dimensions in Kernel Feature Spaces. *Journal of Machine Learning Research*, 9:1875–1908.

[Bressler et al., 2013] Bressler, D. W., Fortenbaugh, F. C., Robertson, L. C., and Silver, M. A. (2013). Visual spatial attention enhances the amplitude of positive and negative fMRI responses to visual stimulation in an eccentricity-dependent manner. *Vision Research*, 85:104–112.

[Cadieu et al., 2014] Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. a., Majaj, N. J., and DiCarlo, J. J. (2014). Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. *arXiv preprint arxiv:1406.3284*.

[Carandini et al., 2005] Carandini, M., Demb, J. B., Mante, V., Tolhurst, D. J., Dan, Y., Olshausen, B. A., Gallant, J. L., and Rust, N. C. (2005). Do we know what the early visual system does? *The Journal of Neuroscience*, 25(46):10577–10597.

[Coates and Ng, 2011] Coates, A. and Ng, A. Y. (2011). Selecting receptive fields in deep networks. In *Advances in Neural Information Processing Systems*, pages 2528–2536.

[Coates and Ng, 2012] Coates, A. and Ng, A. Y. (2012). Learning feature representations with K-means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer.

[Coates et al., 2011] Coates, A., Ng, A. Y., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223.

[Daniel J. Graham, 2006] Daniel J. Graham, Damon M. Chandler, D. J. F. (2006). Can the theory of whitening explain the center-surround properties of retinal ganglion cell receptive fields? *Vision research*, 46(18):2901–2913.

[Donahue et al., 2014] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2014). Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *arXiv preprint arXiv:1411.4389*.

[Dosovitskiy and Brox, 2015] Dosovitskiy, A. and Brox, T. (2015). Inverting Convolutional Networks with Convolutional Networks. *arXiv preprint arXiv:1506.02753*.

[Felleman and Van Essen, 1991] Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1(1):1–47.

[Gallant et al., 2011] Gallant, J. L., Nishimoto, S., Naselaris, T., and Wu, M. C. (2011). System identification, encoding models and decoding models: A powerful new approach to fMRI research. In Kriegeskorte, N., editor, *Visual Population Codes*, chapter 6, pages 163–188. The MIT Press.

[Gerven, 2014] Gerven, M. A. J. V. (2014). Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Brain's Ventral Visual Pathway. *arXiv preprint arXiv:1411.6422v1*.

[Gerven and Lange, 2010] Gerven, M. A. J. V. and Lange, F. P. D. (2010). Neural Decoding with Hierarchical Generative Models. *Neural Computation*, 3142:3127–3142.

[Güçlü and van Gerven, 2014] Güçlü, U. and van Gerven, M. a. J. (2014). Unsupervised Feature Learning Improves Prediction of Human Brain Activity in Response to Natural Images. *PLoS Computational Biology*, 10(8):e1003724.

[Häusler et al., 2013] Häusler, C., Susemihl, A., and Nawrot, M. P. (2013). Natural image sequences constrain dynamic receptive fields and imply a sparse code. *Brain research*, 1536:53–67.

[Horikawa et al., 2013] Horikawa, T., Tamaki, M., Miyawaki, Y., and Kamitani, Y. (2013). Neural Decoding of Visual Imagery During Sleep. *Science*, 340(6132):639–642.

[Horton, 2006] Horton, J. C. (2006). Ocular integration in the human visual cortex. *Canadian Journal of Ophthalmology*, 41(5):584–593.

[Hu et al., 2014] Hu, X., Zhang, J., Qi, P., and Zhang, B. (2014). Modeling response properties of V2 neurons using a hierarchical K-means model. *Neurocomputing*, 134(0):198 – 205.

[Hubel and Wiesel, 1959] Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574.

[Hubel and Wiesel, 1970] Hubel, D. H. and Wiesel, T. N. (1970). The period of susceptibility to the physiological effects of unilateral eye closure in kittens. *The Journal of Physiology*, 206(2):419–436.

[Huth et al., 2012] Huth, A. G., Nishimoto, S., Vu, A. T., and Gallant, J. L. (2012). A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, 76(6):1210–1224.

[Hyvärinen et al., 2009] Hyvärinen, A., Hurri, J., and Hoyer, P. O. (2009). *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, volume 39. Springer.

[Ito and Komatsu, 2004] Ito, M. and Komatsu, H. (2004). Representation of angles embedded within contour stimuli in area V2 of macaque monkeys. *The Journal of Neuroscience*, 24(13):3313–3324.

[Kay et al., 2008] Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, 452(7185):352–355.

[Khaligh-Razavi et al., 2014] Khaligh-Razavi, S.-M., Henriksson, L., Kay, K., and Kriegeskorte, N. (2014). Explaining the hierarchy of visual representational geometries by remixing of features from many computational vision models. *bioRxiv preprint bioRxiv:009936*.

[Konda et al., 2013] Konda, K. R., Memisevic, R., and Michalski, V. (2013). The role of spatio-temporal synchrony in the encoding of motion. *arXiv preprint arXiv:1306.3162*.

[Kriegeskorte et al., 2008] Kriegeskorte, N., Mur, M., and Bandettini, P. (2008). Representational similarity analysis–connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:(4).

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Le et al., 2011] Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[Mahendran and Vedaldi, 2014] Mahendran, A. and Vedaldi, A. (2014). Understanding deep image representations by inverting them. *arXiv preprint arXiv:1412.0035*.

[Marr, 1982] Marr, D. (1982). *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. University Press Group Limited.

[Mika et al., 1998] Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M., and Rätsch, G. (1998). Kernel PCA and De-Noising in Feature Spaces. *NIPS*, 4(5):7.

[Montavon, 2013] Montavon, G. (2013). *On layer-wise representations in deep neural networks*. PhD thesis, Technische Universität Berlin.

[Naselaris et al., 2011] Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fMRI. *Neuroimage*, 56(2):400–410.

[Naselaris et al., 2009] Naselaris, T., Prenger, R. J., Kay, K. N., Oliver, M., and Gallant, J. L. (2009). Bayesian reconstruction of natural images from human brain activity. *Neuron*, 63(6):902–915.

[Nishimoto et al., 2011] Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, J. L. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, 21(19).

[Olshausen and Field, 2005] Olshausen, B. A. and Field, D. J. (2005). How close are we to understanding V1? *Neural computation*, 17(8):1665–1699.

[Pehlevan and Chklovskii, 2015] Pehlevan, C. and Chklovskii, D. B. (2015). A hebbian and anti-hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features. *arXiv preprint arXiv:1503.00680*.

[Ramakrishnan et al., 2015] Ramakrishnan, K., Scholte, H. S., Groen, I. I. a., Smeulders, A. W. M., and Ghebreab, S. (2015). Visual dictionaries as intermediate features in the human brain. *Frontiers in Computational Neuroscience*, 8(January):1–10.

[Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

[Simonyan et al., 2013] Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034*, pages 1–8.

[Vangeneugden et al., 2014] Vangeneugden, J., Peelen, M. V., Tadin, D., and Battelli, L. (2014). Distinct neural mechanisms for body form and body motion discriminations. *The Journal of Neuroscience*, 34(2):574–585.

[Vinnikov and Shalev-Shwartz, 2014] Vinnikov, A. and Shalev-Shwartz, S. (2014). K-means recovers ICA filters when independent components are sparse. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 712–720.

[Zeiler and Fergus, 2013] Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*.

[Zeiler et al., 2010] Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R. (2010). Deconvolutional networks. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE.