Bar Ilan University
Department of Computer Science

Obtaining Scalable and Accurate Classification in Large Scale Spatiotemporal Domains

by

Igor Vainer

Submitted in partial fulfillment of the requirements for the Master's Degree
in the Department of Computer Science

Ramat Gan, Israel

· October 2009 ·

This work was carried out under the supervision of

Prof. Sarit Kraus
Prof. Gal A. Kaminka

Department of Computer Science, Bar-Ilan University

Abstract

We present an approach for learning models that obtain accurate classification of data objects, collected in large scale spatiotemporal domains. The model generation is structured in three phases: spatial dimension reduction, spatiotemporal features extraction, and feature selection. Novel techniques for the first two phases are presented, with two alternatives for the middle phase. We explore model generation based on the combinations of techniques from each phase. We apply the introduced methodology to datasets from the Voltage-Sensitive Dye Imaging (VSDI) domain, where the resulting classification models successfully decode neuronal population responses in the visual cortex of behaving animals. VSDI is currently the best technique enabling simultaneous high spatial ($10,000$ points) and temporal ($10\,ms$ or less) resolution imaging from neuronal population in the cortex. We demonstrate that not only our approach is scalable enough to handle computationally challenging data, but it also contributes to the neuroimaging field of study with its decoding abilities. The effectiveness of our methodology is further explored on a dataset from the hurricanes domain, and a promising direction, based on the preliminary results of hurricane severity classification, is revealed.

Acknowledgments

My first and foremost gratitude goes to my instructors, Sarit Kraus and Gal Kaminka, and to Hamutal Slovin, the domain expert in the VSDI field. This work would not have been possible without Sarit's ideas, helpful suggestions, skilled guidance, everlasting support and unique practicality. While one could only wish for having an instructor as great as she is, I was lucky enough to have two of them—giving an exceptional contribution together. Thanks to Gal, who has taught me many lessons in scientific thinking, I have started to understand what the research really is. He is an uncompromising educator with a distinct scientific integrity. My huge appreciation is of course given to Hamutal, the neuroimaging expert whom I had the pleasure to work with, for providing the exclusive VSDI data, and for making every possible effort to answer all of my inquiries and support my work. Thanks to her, I have managed to learn a thing or two in brain sciences. Truly, I cannot imagine a better team of professionals assisting and guiding me through this study.

Additionally, I wish to thank Elhanan Meirovithz for his help in VSDI data acquisition, and Ofer Garnett for his helpful advice. Great appreciation is due to Haim Avron for quick and professional work on the issue of hurricane datasets generation.

Finally, and the most importantly, I wish to thank my family for their enormous support—my parents, Lidia and Leonid, for help in difficult times; my beloved wife Ricki, the most wonderful partner I could ever hope for—without her enormous understanding, never-ending help and tremendous support during this long period of time, I could have never succeeded; and last, but not the least, my wonderful kids Ariel and Keren, for unwillingly giving me the time to finish this work, on the expense of our time together.

This research was supported in part by ISF grants #859/05 (to H. S.) and #1357/07.

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

There is much interest in applying machine learning in domains with large scale spatiotemporal characteristics. Examples range from learning patterns and discriminating cognitive brain states using functional Magnetic Resonance Imaging (fMRI) [1, 3, 12, 19, 21–23, 29, 42, 43], to developing techniques for classification of brain signals in Brain Computer Interfaces (BCI) [4,16,28,31,41,44], performing automated video classification [39], computer worm detection [33] and many more.

However, many existing techniques prove insufficient when the data is temporal (spanned over a time course) and spatially large (consists of a large number of locations in space). Classification often becomes computationally infeasible. Raw data collected along the time course in a high-resolutional space results in hundreds of thousands of features, for which classical, straightforward machine learning approaches become ineffective in practice. While there have been attempts at addressing these challenges [4, 43], they have proven insufficient.

In this work we present a methodology for both overcoming the scalability challenge and exploiting the spatiotemporal properties of the data for classification. Our methodology is based on common machine learning elements, and is comprised of three phases. First, we present a greedy *pixel selection* technique, i.e. choosing the most discriminative spatial characteristics within the full spatial range in a sample's space, based on the random subspace method [15]. Second, we provide two alternatives for *feature extraction*, applied on the spatially-reduced samples produced by the first phase: features as pixels in time and spatial averaging of pixel groups based on inter-pixel correlation. Finally, we employ a simple and yet effective *feature selection* based on information gain filtering.

Initially, we evaluate our methodology in the neuroimaging domain, and demonstrate how it helps to decode neuronal population responses in the visual cortex of monkeys, collected using Voltage-Sensitive Dye Imaging (VSDI) [30]. VSDI is capable of measuring neuronal population responses at high spatial ($10,000$ pixels of size 60×60 to $170 \times 170 \mu m^2$ each) and temporal ($10\,ms$ or less) resolutions. The produced data consists of tens of thousands of pixels (numeric values, correlated to locations in space), rapidly changing during the time course. Our methodology makes it possible to process this massive amount of data in a computationally feasible manner. It serves as a tool that aids to decode these responses, as we show how to carefully pick and process those specific properties of the data that carry the most discriminative nature. While first attempts to decode neuronal population responses collected using VSDI were performed in [2], no machine learning methods were used—a specially designed statistical approach of pooling rules was developed (relying on the amplitude of the response and other neuronal characteristics). To the best of our knowledge, this is the first time where machine learning techniques are applied in this field.

Further in our research, we explore the effectiveness of our presented methodology in its application to the hurricanes domain. While our introduced techniques were primarily developed with the help of thorough evaluation and exploration of the VSDI data, our approach is intended to be applicable in general spatiotemporal domains with analogous characteristics. In the hurricanes domain, we analyze historical data of the Atlantic region—satellite images and hurricane tracks—in attempt to classify the hurricane severity group by generating a dataset based on plain periodical satellite shots along the time course. The kind of application we explore here examines how our methodology handles the challenges proposed in a domain highly different than the VSDI.

The rest of this document is organized as follows: Chapter 2 contains a review of related work, briefly describing the machine learning tools used in our research; it provides a literature review from both spatiotemporal *domains* and spatiotemporal *techniques* perspectives. Chapter 3 is the core chapter of this work which describes our three phase methodology for spatiotemporal classification modeling. Chapter 4 presents the empirical evaluation techniques employed for the validation of our methodology in the VSDI domain, and contains a thorough analysis of the experimental results. In Chapter 5 we introduce the first results of applying our methods in the hurricanes domain, after presenting the challenges and discussing the differences between the two domains. Chapter 6 concludes our work by discussing its implications and contributions, along with the prospective directions for the future.

Chapter 2

Related work

In reviewing the related work, we separate the work on spatiotemporal *domains* from the work on spatiotemporal *techniques*. For the domains analysis, we explore the machine learning research related to VSDI—decoding in the neuroimaging field and classification of brain-emitted signals. In domains with the appropriate characteristics, such as fMRI and BCI, machine learning techniques are commonly employed. As for the survey of the techniques, we analyze how machine learning is applied to either spatial, temporal or spatiotemporal data—and explore the relevance of the analyzed methods to our work, since our main focus is on the techniques, rather than on the domains.

2.1 Related domains

The studies discussed next analyze the progress done in the last decade in the neuroimaging domains, and introduce the first work related to decoding in VSDI. This discussion is essential for understanding how the machine learning is currently employed in the neuroimaging field, and for learning from it for the purpose of our study.

2.1.1 fMRI from a spatial perspective

A classic goal in machine learning application to fMRI is the discrimination between different cognitive states—"decoding" of the states the brain is in. This is done by taking the fMRI brain images resulted from being in some state, and telling what the triggering state was. In the case of [17], individual classifier was built for each human subject to differentiate between states such as "observing a picture" or "reading a sentence", yielding an accuracy of either 80% or 96% (based on the experiment type). Same process was carried on when the discrimination was done by training a single classifier, and using it across different subjects [35]; or, by developing methods that can account for subject-specific variations [24]. Both of these studies have produced accuracies in the range of 60% to 80%, depending on the experiment types and methods. Spatial patterns of brain activity were also used for training classifiers applicable to lie detection [3], resulting in accuracies of above 88% on the test data.

In cases of sparse, high dimensional problems—where the number of features greatly exceeds the number of training examples per class—a hierarchical Bayesian framework was developed in [23], resulting in precision in the range of 53% to 85% (depending on the test subject and the experiment type).

A case study of visuo-motor sequence learning was presented in [29], yielding precisions in the range of 62.5% to 80%. In [14], motor tasks were predicted with a misclassification error ranging from 15% to 30%.

Multivariate pattern recognition algorithms, employing linear discriminant analysis using probabilistic methods, were applied for the decoding of mental states in [10]: these included predicting the orientation of invisible stimuli [8] (resulting in up to 60% and up to 80% accuracy rates in two different kinds of experiments) and predicting the stream of consciousness [9] (exploring the "binocular rivalry" phenomenon, reaching accuracy of $80-85\%$). Decoding of visual contents of the human brain—of seen and attended motion directions—was performed in [11, 12], gaining accuracy of $63-66\%$. This study consisted of classification of different visual orientations, including a "mind reading" experiment—revealing the focus of subject's attention on competing orientations.

2.1.2 Spatiotemporal fMRI

However, the studies presented until this point, did not model the fMRI as spatiotemporal data, but rather as spatial data only, usually from a single time interval. While being the most common non-invasive technique for brain study in humans, its deficiency is that it measures metabolic changes: the hemodynamic response occurring few seconds after the onset of the visual stimulus. Whereas the temporal resolution of neuronal activity is within tens of ms, the resolution of the fMRI signal is at least two orders slower. For this reason, fMRI studies don't usually take advantage of the temporal aspect.

In the work presented in [43], Zhang et al. claim that they took into account the temporal dimension of the fMRI for the first time. It was used to discriminate drug-addicts from non-drug-addicts, and an accuracy of up to 96% was reported. In [21], the temporal information of fMRI was used along with the spatial, "to infer where (in the brain) and when (in time) the discriminating information occurs", producing accuracy of 90%. Temporal information was also used in [22, 36], showing the advantage of using the temporal data. As well, a methodology was developed to offer the ability to locate spatial regions with temporal differences between groups, while simultaneously accounting for and identifying intergroup spatial and temporal variability [26], leading to accuracy of 85.71%. At last, a new feature selection technique for multivariate time series was proposed in [42] and successfully evaluated on fMRI datasets, for cases where the number of spatial features is much larger than the number of temporal features.

2.1.3 EEG and ECoG in BCI

Brain Computer Interface (BCI) technology is an area where brain-controlled computer systems are developed in order to operate the machine (e.g. prostheses, communication) by a brain activity (e.g. imagining a hand movement will cause a prosthetic computer-controlled arm to move). BCI is fundamentally based on techniques for classification of brain signals. The common method to read brain activity for BCI is Electroencephalogram (EEG), a multivariate time series data non-invasively collected from the scalp. Another method is Electrocorticography (ECoG), having a higher signal-to-noise ratio, as well as higher spectral and spatial resolutions, resulting from being invasive.

Machine learning based classification in EEG (e.g. hand movement imagery) was successfully employed in [28,31,41,44] with the temporal aspect of the data being utilized: in [44], for instance, a framework of feature extraction for classification of hand movement imagery EEG was proposed. Here, the best classification rate ($88 - 94\%$, varying on the subject) was achieved due to obtaining the optimal spatial and temporal features. Supervised feature selection via dependence estimation was used on BCI datasets in [31], resulting in highly varied error rates of $0.3 - 37.9\%$. A polygon feature selection method was developed in [28], for the classification of temporal EEG data from two or more sources, on the basis of quantifying structural changes with time—producing accuracy of $90 - 99\%$.

As for the high-dimensional multi-channel BCI data, a hybrid wavelet feature selection method, employing the mutual information filter and genetic algorithms, was proposed in [4]. This method was developed for the design of a multi-channel BCI system (as opposed to only one or two EEG channels), coping with the system's high dimensionality.

In ECoG, extremely small training data sets were handled in [25], where the problem of robust classification of ECoG signals for designing a closed-loop BCI control was examined. ECoG changes during movement of two different body parts were examined. The reported average error was 13.7% across nine subjects.

2.1.4 SPECT perfusion imaging

Another related domain, discussed in [32], is the spatial-only Single Photon Emission Computed Tomography (SPECT) domain. It studied the use of SPECT perfusion imaging for differentiation between images of healthy subjects and of Alzheimer's disease patients. The presented approach incorporates proximity information about the features and generates a classifier that selects the most relevant "areas" for classification, rather than the most relevant voxels (the analogous to pixels, but in three dimensional space). This approach resulted in success rates above 90%.

2.1.5 Decoding in VSDI

In addition to the temporal deficiency of fMRI, it samples the space in voxels of one to a few millimeters (only a few thousands of voxels, or well defined regions of interest). As opposed to it, VSDI is capable of measuring neuronal population responses at high spatial and temporal resolutions. Therefore, it provides a true insight as to the neuronal dynamics from both spatial and temporal aspects.

The first reference we have on the decoding of neuronal population responses to visual detection tasks using VSDI is [2]. A specially designed method based on six neuronal population responses pooling rules was used here, with no machine learning use whatsoever. This method relied on the amplitude of the response, and other neuronal characteristics—tailored strictly for VSDI. While this method has produced nearly perfect results, it incorporates a domain-specific knowledge—rendering it as not general enough for use in spatiotemporal domains other than VSDI.

2.2 Related techniques

The techniques described next are used for feature extraction and feature selection of problems that handle either spatial, temporal or spatiotemporal data. We explore the methods relevant for our work, and survey the classification algorithms being used.

2.2.1 Spatial machine learning

While the focus of our work is classification in large scale spatiotemporal domains, some techniques for spatial domains alone—which either don't take advantage of the temporal aspect, or don't inherently have it—are worth mentioning.

For the reasons presented above, in 2.1.2, fMRI studies don't usually take advantage of the temporal aspect. Such studies included picking the top n most active voxels based on t-test [17] or on average between the voxels [35]; picking the top n most discriminating voxels, based on training a classifier per each voxel [19]; or, picking the n most active voxels per Region Of Interest (ROI) [19]. The classifiers in this case were mostly Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), and k-Nearest Neghbor (kNN). While these studies managed to produce moderate to high accuracy results, they relied on relatively small resolutions of data (where training a classifier per voxel was admissible), or on expert knowledge (defining an ROI). The methods we present in our work require no prior knowledge, are aimed at very high resolutional data, and exploit both temporal and spatial dimensions.

When a higher resolutional data was faced in fMRI, a use of uniform sampling of the data was employed in [3], utilizing SVM with Gaussian kernel. While sampling is indeed a reasonable tool when handling high resolutional data, a uniform sampling is lacking in a sense that it does not choose the more discriminative data-points, in favor of the less discriminative ones. Hence, while our pixel selection technique is also based on sampling, it adopts elements from the random subspace method presented in [15] rather than basing its selection on uniform sampling.

Last, we present a domain in which a classification of very high resolution panchromatic images from urban areas was done. In [5], an area filter was used to extract information about the inter-pixel dependency. Using a linear composition of kernels, a kernel was defined using both the spectral (i.e. the original gray level of each pixel) and the spatial information, reaching partial success in some types of areas. While this domain resembles the domains in our focus only in its spatial nature, we liked the idea of exploring the inter-pixel dependency, and further developed it for our feature extraction technique.

2.2.2 Temporal analysis and reduction techniques

Classification in large scale spatiotemporal domains often requires both spatial and temporal dimensionality reduction. When the temporal reduction is due, the methods described below provide a good base to rely on. While the methodology we present in Chapter 3 in our work uses much simpler temporal reduction techniques than the methods surveyed below, we introduce these methods for the purpose of the future work discussion in Chapter 6.

Discrete Wavelet Transform (DWT) was used in [34] for dimensionality reduction of time series. The objective here was to find a representation at a lower dimensionality that preserves the original information, describing the original shape of the time series data as closely as possible. An improved version of k-means clustering algorithm introduced here was shown to have results superior to k-means.

DWT and Discrete Fourrier Transform (DFT) were also successfully employed in [20] for time series data mining, where each time series was compressed with wavelet or Fourier decomposition, but instead of using only the first coefficients, a method of choosing the best coefficients for a set of time series was presented.

A different approach was presented in [33], a study for improving computer worm detection using Artificial Neural Networks (ANN). The temporal analysis techniques described here involved a simple sliding window, a simple exponential compression and a Poisson exponential compression. Even when the temporal dimension was reduced, implying the apparent loss of data (some of which is essential for successful classification) this study has still managed to achieve 85% as its best accuracy result.

2.2.3 Spatiotemporal machine learning

In fMRI exploiting the temporal dimension, the following heuristic was employed in [22]: features were defined as voxel-timepoint pairs, ranked by how well they individually classify a training set, and the top 100 features for the final classifier were chosen. While individual training of classifiers for all time-space combinations is computationally unfeasible in large scale domains, we chose to adopt the time-space combination approach in our work. Additional work that has inspired us is [43], in which one of the introduced techniques for feature selection was defining voxel-specific time-series analysis, by ranking features by mutual information with respect to the class variable. From the ranked features, the n highest ranked were selected, and closeness of each pair of voxels' time series was measured. Despite the high reported success rates, the techniques in [43] are computationally expensive in large scale domains such as ours.

The method presented in [41], utilized in the BCI domain, maintains the correlation information between spatial time-series items by utilizing the correlation coefficient matrix of each such item as features to be employed for classification. Then, the Recursive Feature Elimination (RFE) is being used for feature subset selection of time-series datasets. RFE was first proposed in gene selection problem in [6], where the SVM's weight factor was used as ranking criterion for features, and the features with the smallest ranking criterion were recursively eliminated. However, applying RFE or an RFE-like procedure, in a similar manner on our type of data is computationally expensive[1]—nevertheless, we do adopt the approach of correlation between spatial elements in one of our feature extraction techniques

[1] As shown in our experiments in Chapter 4.

The approach in [41] has also inspired the work carried out in [29], which selected voxel-pairs based on their ability to discriminate the target classes. In turn, the correlation value between these pairs was used as feature vector to train the GNB, SVM and kNN classifiers.

A last example from spatiotemporal domains is automated video genre classification [39]. In this case, the problem was investigated by first computing a spatiotemporal combined audio-visual “super” feature vector (of very high dimensionality). Then, the feature vector was further processed using Principal Component Analysis (PCA) to reduce the spatiotemporal redundancy while exploiting the correlations between feature elements. This method yielded a 86.5% accuracy on average. However, the PCA-based techniques in multivariate time-series datasets are known to be problematic in regard to scalability (Yang and Shahabi [40] attempted to address this problem), which is more than evident in our domain.

Chapter 3

Spatiotemporal classification modeling

In this chapter, we present a *three phase methodology* for building scalable models for spatiotemporal data classification. To describe our methodology, we first formalize the problem in Section 3.1. In Section 3.2, we provide a brief overview of the main phases of our methodology. In the sections following it, we describe each of the phases in detail.

3.1 Problem formalization

A spatiotemporal domain contains n pixels that constitute the global pixel set $P = \{p_1, p_2, \ldots, p_n\}$. Every pixel p_i, $i \in \{1, \ldots, n\}$ represents a concrete location in space, in which a series of m contiguous values in time is measured. The intervals between each two consequent values in time are equal. In turn, p_i^t, $i \in \{1, \ldots, n\}$, $t \in \{1, \ldots, m\}$ indicates the specific timeframe t along the time course, at which the value of p_i is measured. In fact, p_i^t represents the pixel-in-time combination of pixel p_i and time t.

A finite training samples set of size k in the spatiotemporal domain is defined as: $S = \{s_1, s_2, \ldots, s_k\}$, where a single sample s_l, $l \in \{1, \ldots, k\}$ is a set of vectors: $s_l = \{\overline{p_1}, \ldots, \overline{p_n}\}$, where a vector $\overline{p_i} = \left\langle v_1^i, \ldots, v_m^i \right\rangle$, $v_t^i \in \mathbb{R}$, $t \in \{1, \ldots, m\}$ denotes the actual m values along the time course, measured for the pixel p_i in the sample s_l. Each training sample $s_l \in S$ is labeled with a class label $c \in C$. For an infinitely large universal set U of all possible unlabeled samples $u = \{\overline{p_1}, \ldots, \overline{p_n}\}$, $u \in U$, the classification problem is to build a model that approximates classification functions of the form $f : U \longrightarrow C$, which map unclassified samples from U to the set of class labels C.

3.2 Methodology overview

In the next sections, we describe each of the phases of our methodology in detail. Section 3.3 presents $GIRSS$, a technique for selecting the pixels that have the most discriminative characteristics within the full spatial range of a sample's space. Next, in Section 3.4, we introduce two alternative techniques for extracting the features from the pixels selected in the first phase—the PIT, a simple pixel-in-time approach, and the $IPCOSA$, an inter-pixel correlation based spatial averaging method. The third phase described in Section 3.5 presents an effective application of feature selection on the product of the second phase, to further improve the abilities of the remaining features that constitute the generated models. Figure 3.1 sketches the outline for our methodology.

3.3 Pixel selection via greedy improvement of random spatial subspace

The technique described here uses common machine learning tools in order to reveal the most informative pixels, which will define the features to be used with our model. The discriminative nature of the selected pixels stems from analyzing their measured values along the time course. Due to the high spatial and temporal resolutions of the domains in question, our data is comprised of hundreds of thousands of basic data-points. Hence, using the most granular, basic values of the sample's space as features will lead to an extremely high dimensional feature space, rendering classification, or even feature dimensionality reduction techniques, unfeasible. We present here a greedy approach based on the random subspace method [15] for selecting by iterative refinement, the set of pixel subsets from which we can eventually derive the sought-after pixel set.

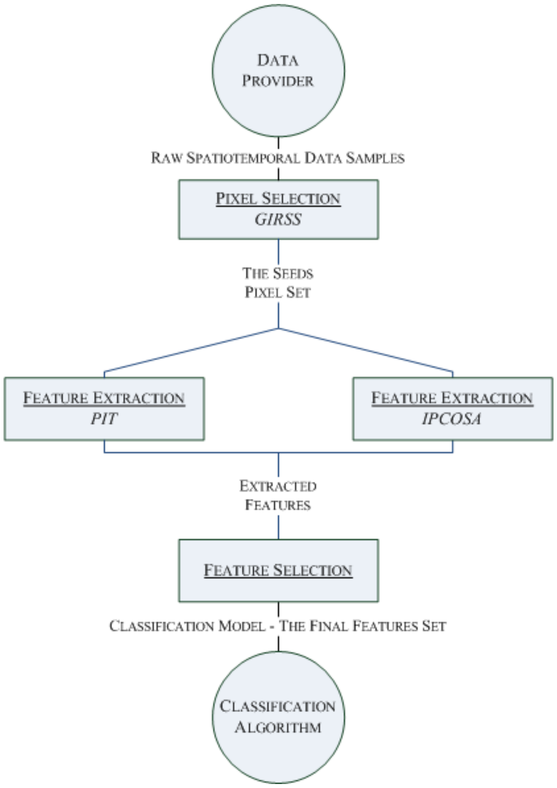


Figure 3.1: The outline of the classification modeling methodology.

3.3.1 The *GIRSS* algorithm

In Algorithm 3.1, we randomly generate r pixel subsets of a requested size u (number of pixels in a subset). Handling small pixel subsets yields an easier handling of a reduced spatial dimension. However, in order to cover a large portion of pixels (inherently, features) in the data and to establish credibility for the selected pixels, we need to rely on a wide-enough selection of such subsets[1].

The classification capabilities of each of the generated pixel subsets are then roughly evaluated using our pixel set evaluation method (Algorithm 3.2). This method is a heuristic for giving an evaluation score to a pixel subset, which in fact builds a small classification model based on it. Here, pixel values in time (all pixel-time pairs) are defined as features

[1] From our experience, having $u \cdot r \approx 1.5n$, u and r being of about the same order of magnitude (see Table 4.1), is usually more than enough—as it provides a broad coverage of the pixels space, and at the same time a fast-enough handling of individual subsets (of course, sensitivity analysis for these two parameters is due when refining our technique).

(step 2), as was done in [22]. Then an Information Gain (InfoGain) based feature selection [38] is applied to select only the features with positive InfoGain scores (step 3). Our usage of InfoGain for ranking features by mutual information with respect to the class is inspired by [43], an fMRI study exploiting the temporal dimension. The resulting feature set is cross-validated using linear-kernel SVM (WEKA's implementation of the SMO algorithm, [38]) to obtain an evaluation score (cross-validation accuracy of the evaluated set). The produced scores are then ordered in a descending order, and the greedy phase begins.

Algorithm 3.1 Greedy Improvement of Random Spatial Subspace—$GIRSS\left(S, C, u, r\right)$

```
Input: Sample set $S$, label set $C$, size of random spatial subspace $u$, number of random spatial subspaces $r$

1. Initialize pixel subsets evaluation scores vector: $Z\left[1:r\right] \longleftarrow 0$
2. for $i = 1$ to $r$ do:
    (a) Generate the random permutation vector: $n^i = permute\left(\{1, 2, \ldots, n\}\right)$
    (b) Generate the index vector: $d^i = \left\{n_1^i, n_2^i, \ldots, n_u^i\right\}$
    (c) Select pixel subset (random spatial subspace) indicated by $d^i$: $\tilde{P}^{d^i} \subset P$
    (d) Save the pixel subset's evaluation score:
        $Z\left[i\right] \longleftarrow evaluatePixelSet\left(S, Y, \tilde{P}^{d^i}, u\right)$
3. Produce sorted indices vector $I_Z\left[1:r\right] \longleftarrow indices\left(sort\left(Z\left[1:r\right]\right)\right)$ to contain indices
   of $Z\left[1:r\right]$ in the order matching the sorted scores of $Z\left[1:r\right]$ (highest scores leading).
4. Initialize the set of pixel subsets $\Gamma$ with the highest-ranked pixel subset:
   $\Gamma \longleftarrow \left\{\tilde{P}^{d^{I_Z[1]}}\right\}$
5. Initialize $z$ with the score of the highest-ranked pixel subset: $z \longleftarrow Z\left[I_Z\left[1\right]\right]$
6. for $j = 2$ to $r$ do:
    (a) $\Gamma^{'} \longleftarrow \Gamma \cup \left\{\tilde{P}^{d^{I_Z[j]}}\right\}$
    (b) $P^{'} \longleftarrow extractHighestRankedPixels\left(S, C, \Gamma^{'}, |\Gamma^{'}|, Z\left[1:r\right]\right)$
    (c) $z^{'} \longleftarrow evaluatePixelSet\left(S, C, P^{'}, u\right)$
    (d) if $z^{'} > z$, update the $\Gamma$ and its score: $z \longleftarrow z^{'}$, $\Gamma \longleftarrow \Gamma^{'}$.
7. $P^* \longleftarrow extractHighestRankedPixels\left(S, C, \Gamma, u, Z\left[1:r\right]\right)$

Output: Pixel set $P^* = \{p_1^*, p_2^*, ..., p_u^*\}$ (top $u$ spatial subspace representatives).
```

During the greedy phase, we maintain a set Γ of pixel subsets, of which the desirable pixel set can be derived at any time. Initially, Γ is initialized with the highest-ranked pixel subset (along with its evaluation score). In each iteration over the ranked pixel subsets list, the next subset in the list joins Γ. A set of pixels of size u is then extracted from Γ (refer to Algorithm 3.3), and evaluated (again, using Algorithm 3.2). The greedy step: if the resulting evaluation score is higher than the existing evaluation score of Γ, the current pixel subset remains in Γ. Otherwise, it is discarded. This way Γ maintains only those pixel subsets along the way which are capable to produce a highly evaluated pixel subset (whose size is equal to the size of any of the pixel subsets in Γ), in any requested time. Finally, when the iteration over the pixel subsets is over, the desirable set of pixels is extracted from Γ to serve as the pixel selection.

```
**Algorithm 3.2** Pixel Set Evaluation—$evaluatePixelSet\left(S, C, P^{'}, u\right)$

**Input**: Sample set $S$, label set $C$, sorted pixel set $P'$, size of the random spatial subspace $u$.

  1. $P^{''} \longleftarrow p_i \in P^{'} \mid i \in \left\{1, \ldots, min\left(u, |P^{'}|\right)\right\}$.
  2. Extract feature-set: $F = \left\{p_j^t \mid t \in \{1, \ldots, m\}, \forall p_j \in P^{''}\right\}$ over the sample set $S$.
  3. Perform feature-selecton in $F$ to obtain reduced feature set $F^{'}$, using $InfoGain\left(S, F, C\right)$, producing scores: $IG\left(p_j^t\right), \forall p_j^t \in F$ . Select only features having $IG\left(p_j^t\right) > 0$.
  4. $z \longleftarrow$ Accuracy score of a 10-fold cross-validation of $F^{'}$ applied on $S$ using $SVM\left(S, F^{'}, C\right)$.

**Output**: Evaluation score $z$.
```

The extraction of the highest ranked pixels set from Γ (Algorithm 3.3), at any stage of *GIRSS*, is done as follows: each individual pixel subset in Γ is turned into a feature set, where pixel values in time are defined as features (step 2a). An InfoGain based feature selection is applied on this feature set, and the InfoGain scores for each feature are taken (step 2b). The score for each individual pixel is calculated by averaging (along the number of pixel instances) the weighted averages of InfoGain scores (along the pixel's time course in each of the feature sets) (step 2c). The evaluation score of each pixel subset in Γ is used as the weight for computing the grand-average, effectively giving higher weight to pixels and features stemmed from highly evaluated pixel subsets.

Algorithm 3.3 Highest Ranked Pixels Extraction—
$extractHighestRankedPixels\left(S, C, \Gamma, u, Z\left[1:r\right]\right)$

```
Input: Sample set $S$, label set $C$, set of pixel subsets $\Gamma = \{P_1, P_2, \ldots\}$, size of the random spatial subspace $u$, pixel subsets score vector $Z\left[1:r\right]$.
  1. Initialize pixels score vector: $\rho\left[1:n\right] \longleftarrow 0$ and pixels instances vector: $\iota\left[1:n\right] \longleftarrow 0$.
  2. for $\forall P_i \in \Gamma$ do:
      (a) Extract feature-set: $F = \left\{p_j^t \mid t \in \{1, \ldots, m\}, \forall p_j \in P_i\right\}$ over the sample set $S$.
      (b) Rank features in $F$ using $InfoGain\left(S, F, C\right)$ producing scores: $IG\left(p_j^t\right), \forall p_j^t \in F$.
      (c) for $\forall p_j \in P_i$ do: $\rho\left[j\right] = \frac{\rho[j]\cdot\iota[j]+Z[i]\cdot\frac{\sum_{t=1}^{m} IG\left(p_j^t\right)}{m}}{\iota[j]+1}$, $\iota\left[j\right] = \iota\left[j\right] + 1$.
  3. Produce sorted pixel indices vector $I_\rho\left[1:n\right] \longleftarrow indices\left(sort\left(\rho\left[1:n\right]\right)\right)$ to contain indices of $\rho\left[1:n\right]$ in the order matching the sorted scores in $\rho\left[1:n\right]$ (highest scores leading).
Output: Top $u$ ranked pixels $p_{I_\rho[l]} \in P$, $l \in \{1, \ldots, u\}$.
```

3.3.2 Complexity analysis

The time and space complexity of Algorithm 3.2, the *evaluatePixelSubset*, is linear in the number of all combinations of pixel-in-time pairs per sample—i.e., in the number of basically defined features. Therefore, the first two steps in the algorithm have a cost of $O\left(kmu\right)$, and so is the InfoGain feature selection step (which is linear in the number of features). An efficient, state of the art SVM implementation, is linear in the number of samples (inherently, features), so our cross-validation using the SVM also has a bound of $O\left(kmu\right)$. Thus, the overall time and space complexity of Algorithm 3.2 is $O\left(kmu\right)$.

In *extractHighestRankedPixels*, Algorithm 3.3, the size of Γ is bounded by r, so the algorithm's single loop is performed at most r times. The cost of each iteration of the loop is $O\left(kmu\right)$, for the reasons stated earlier in the analysis of Algorithm 3.2 (feature set extraction, InfoGain ranking and values averaging). Besides the loop, we initialize and sort a vector of length n. This results in the overall time complexity of the algorithm of $O\left(n \log n + rkmu\right)$. The space complexity is different: we only need $O\left(n\right)$ storage for the scores vector, and an additional $O\left(kmu\right)$ space for a single loop iteration. Therefore, the space complexity of the algorithm is $O\left(n + kmu\right)$.

The time complexity of the main Algorithm 3.1, the $GIRSS$, is analyzed as follows. Each iteration of the first loop, repeated r times, has a cost of $O\left(n\right)$ added to the cost of Algorithm 3.2, the $evaluatePixelSubset$, resulting in an iteration cost of $O\left(n + kmu\right)$. Therefore, the loop's total cost is $O\left(r\left(n + kmu\right)\right)$. Each iteration of the second loop, which is also repeated r times, has a cost of Algorithm 3.3, the $extractHighestRankedPixels$, added to the cost of $evaluatePixelSubset$—resulting in the total cost of $O\left(r\left(n\log n + rkmu\right)\right)$. Lastly, we sort a vector of length r once, a step that costs $O\left(r\log r\right)$. Overall, the time complexity of $GIRSS$ is $O\left(r\left(\log r + n\log n + rkmu\right)\right)$.

As for the space complexity of $GIRSS$, it is $O\left(r + n\right)$ for the evaluation scores vector and random permutation vectors generation, $O\left(n + kmu\right)$ for the calls to Algorithm 3.3, and $O\left(kmur\right)$ for the maintenance of the data structures during the executions of the loops inside the $GIRSS$. Altogether, this results in the space complexity of $O\left(n + rkmu\right)$.

We believe that there is space for further complexity reduction. However, we were satisfied with the performance of the presented version during the experimental evaluation, as detailed in Chapter 4, so the current implementation was retained.

3.4 Feature extraction

Methods described here are applied on the pixel selection results of the first phase (Section 3.3). We present two alternative feature extraction approaches in order to cope with variability evident in different spatiotemporal datasets. Even when the datasets originate from the same domain, they can bear different spatial characteristics, expressed in the noise level and the resolution of the signal collected during the dataset construction. The alternatives provided here are each aimed at a different datasets sector.

3.4.1 Features as pixels in time—the PIT

The straightforward approach for extracting a feature set F from a given pixel set $P^* = \{p_1^*, p_2^*, \ldots, p_u^*\}$ over the sample set S, is to define it as all pixel-in-time combinations

$F = \left\{p_j^t \mid t \in \{1, \ldots, m\}, \forall p_j \in P^*\right\}$, yielding $u \cdot m$ features. We used this approach in Section 3.3 for ranking pixel subsets and feature sets. While for simpler classification tasks this is satisfactory—fast, simple and effective (Chapter 4), a method described next is suggested for more complex tasks.

3.4.2 Spatial averaging of pixel groups based on inter-pixel correlation

The motivation for this method is to overcome the negative effects of a possibly noisy data by performing a spatial-level averaging of pixels that share a common nature. This requires that the trends of their change along the time course will have similar characteristics. Two questions raised here are:

- How to measure similarity between the pixels?
- How to choose "similar" pixels in space, designated for averaging?

The way we measure similarity is by employing Pearson's product moment coefficient [7] between pairs of pixels. This method is simple, suitable with the type of data we have, and was successfully used for calculation of correlation scores in multivariate time series (where correlation is employed for discrimination of target classes [29, 41]).

As for the second question, we perform pixel averaging within groups of "similar" *neighboring* pixels. The reason for this lies in the nature of our data—a non-trivial negative correlation exists between all pixel-pairs correlations and all pixel-pairs distances[2], showing that higher distances between pixels lead to lower correlations between them. Therefore, choosing neighboring groups of pixels as a whole, having a high inter-group similarity, has the potential to reveal stronger discriminative characteristics—rather than picking individual pixels from the same group.

3.4.2.1 The *IPCOSA* algorithm

In Algorithm 3.4 we show how the neighborhood formation for pixel groups generation is done. This formation is based on a given pixel set, a product from the previous phase

[2] During the experimental evaluation of all VSDI datasets (Chapter 4), the coefficient between all pixel-pairs correlations and all pixel-pairs distances was within the range of $\approx -0.45 \pm 0.5$.

introduced in Section 3.3—we refer to this set as “the seeds”. First, we calculate a correlation coefficient matrix C and a distances matrix D between all pixel pairs (step 3); these matrices are symmetric (only one triangle above or below the diagonal is essential). Then we define the set of pixel subsets Δ, which will eventually hold the groups of neighboring pixels that share a similar nature. Next, we employ a graded group formation phase (step 5), where the correlation strength dictates the group formation order: groups having the strongest inter-similarity are generated first, ensuring that the eventually formed groups exploit the similarity property to its full extent (only positive correlation coefficient thresholds are used)[3].

The group formation is subject to the following guidelines: a group of pixels must contain at least one seed within it to base the group on. Once chosen, the seed's proximate neighbors' correlation scores are examined. Neighbors with scores that fit the graded correlation threshold join the seed's group. Recursively, the correlation scores of the neighbors of each of the newly-joined group members are tested, and additional pixels conforming to the correlation and the proximity requirements join the group. Eventually, a group stops expanding once none of the group members' neighbors fits the requirements. At this step, a formed group joins Δ, and its members are no longer available for formation of new groups. A group may consist of a sole seed (step 6). At the end of the group formation phase, Δ contains groups of neighboring pixels, each based on one or more seeds. Some groups have stronger inter-similarity than the others, but due to our graded group formation phase, even the weaker groups are generally based on non-negligible positive correlation scores[4].

At the final phase of our algorithm, the feature extraction is based on Δ's pixel groups: pixel values at each of the points in time are averaged along their spatial dimension—across all pixels within each of the groups of Δ[5]. The resulting features represent the average-in-time of similar pixels, as opposed to the pixel-in-time approach presented in Subsection 3.4.1. For seeds pixel set of size u, there will be at most $u \cdot m$ features (number of formed

[3] Our choice of τ was 0.05 in all our experiments.

[4] As our empirical evaluation of VSDI data shows (Chapter 4), in most cases the weakest formed groups are based on a correlation coefficient of at least 0.4.

[5] Various seeds-based spatial averaging methods were tested during our empirical evaluation of VSDI data, in order to choose the most appropriate method. Please refer to Section 4.5 for additional details.

groups will not exceed the number of seeds, as each group must contain at least one seed).

3.4.2.2 Complexity analysis

For the analysis of time complexity, the $IPCOSA$ has three major parts: the initialization and the calculation of the correlation and the distance matrices, the groups formation and the maintenance of Δ, and the spatial average calculations. The first part has a cost of $O\left((knm)^2\right)$. The second part has a complexity of $O(n)$, thanks to the fact that the groups formation in grids is done by exploring the finite and bounded set of only the closest neighbors of each of the pixels, where each pixel relation evaluation is done only once—resulting in the number of such evaluations being linear in the number of pixels. Since Δ contains at most u groups (due to the property of at least one seed per group), the cost of the third part is simply $O(um)$, $u << n$. Overall, the time complexity of $IPCOSA$ is $O\left((knm)^2\right)$.

The space complexity of $IPCOSA$ is, therefore, $O\left(n^2\right)$ for building the matrices in the first part, $O(n)$ for storing the group formation information in Δ during the execution of the second part, and $O(um)$ for the feature generation step in the third part. Altogether, the space complexity is $O\left(n^2 + um\right)$.

However, it is easy to notice that the version of the algorithm presented here is suboptimal, mainly for the purpose of clarity—there is no need for the initial calculation, nor the storage, of the correlation and the distance matrices. Their values can be computed on demand, while the calculations spread pattern in the grid-formatted space guarantees that only $O(n)$ calculations will be performed (during the second part of the algorithm). Thus, the actual optimal implementation of $IPCOSA$ has complexities of $O(knm)$ for time and $O(n + um)$ for space.

Algorithm 3.4 Inter-Pixel COrrelation based Spatial Averaging—$IPCOSA\left(S, C, P^*, \tau\right)$

Input: Sample set S, label set C, seeds pixel set P^* of size u, correlation threshold step $\tau \in [0, 1]$.

```
1. Set neighboring distance threshold $\mu$ (e.g. for spatially grid-formatted domains: $\mu = \sqrt{2}$). $p_1$ and $p_2$ are neighbors iff $distance\left(coords\left(p_1\right), coords\left(p_2\right)\right) \leq \mu$.

2. Initialize correlation coefficient matrix: $C = 0_{n \times n}$ and distance matrix: $D = 0_{n \times n}$ (symmetric).

3. for $\forall p_i \in P$ do:

   (a) Vectorize all $\overline{p_i}$ values of $p_i$ over the sample set $S = \{s_1, s_2, \ldots, s_k\}$ to produce super-vector of length $m \cdot k$ with all of concatenated $\overline{p_i}$ values:
       $q_i = \left\langle \left\langle v_1^i, \ldots, v_m^i \right\rangle_{s_1} \cdots \left\langle v_1^i, \ldots, v_m^i \right\rangle_{s_k} \right\rangle$
   (b) for $\forall p_j \in P$, $p_i \neq p_j$ do (for every pair $p_i, p_j$):
       i. Vectorize all $\overline{p_j}$ values of $p_j$ over the sample set $S = \{s_1, s_2, \ldots, s_k\}$ to produce super-vector of length $m \cdot k$ with all of concatenated $\overline{p_j}$ values:
          $q_j = \left\langle \left\langle v_1^j, \ldots, v_m^j \right\rangle_{s_1} \cdots \left\langle v_1^j, \ldots, v_m^j \right\rangle_{s_k} \right\rangle$
       ii. Compute correlation coefficient: $C_{(i,j)} = correlation\left(q_i, q_j\right)$.
       iii. Compute distance: $D_{(i,j)} = distance\left(coords\left(p_i\right), coords\left(p_j\right)\right)$.

4. Initialize $\Delta$, the set of pixel subsets: $\Delta \longleftarrow \emptyset$, and $R$, the retaining pixel set: $R \longleftarrow P$.

5. for $r \in \{1, 1-\tau, 1-2\tau, \ldots, \tau\}$ do:

   (a) while $\exists p \in R$ s.t. $p \in P^*$ ($p$ is a seed) and $\exists \hat{p} \in R$ s.t. $C_{(\hat{p},p)} \geq r - \tau$ and $D_{(\hat{p},p)} \leq \mu$:
       i. Initialize $G$, pixel subset group, $G \longleftarrow \{p\}$.
       ii. $R \longleftarrow R \setminus \{p\}$
       iii. while $\exists p^{'} \in R$ and $\exists \tilde{p} \in G$ s.t. $C_{\left(\tilde{p},p^{'}\right)} \geq r - \tau$ and $D_{\left(\tilde{p},p^{'}\right)} \leq \mu$:
            A. $G \longleftarrow G \cup \left\{p^{'}\right\}$
            B. $R \longleftarrow R \setminus \left\{p^{'}\right\}$
       iv. $\Delta \longleftarrow \Delta \cup \{G\}$

6. for $\forall p \in R$ s.t. $p \in P^*$ ($p$ is a remaining seed in $R$) do:

   (a) $R \longleftarrow R \setminus \{p\}$, $G \longleftarrow \{p\}$, $\Delta \longleftarrow \Delta \cup \{G\}$

7. Initialize feature-set $F^*$ over the sample set $S$, $F^* \longleftarrow \emptyset$.

8. for $t = 1$ to $m$ do:

   (a) Define $f^t$—the average of values of all pixels in $G$ at time $t$:
       $f^t = \frac{\sum_{i=1}^{|G|} v_t^i}{|G|}$, $s.t. \overline{p_i} = \left\langle v_1^i, \ldots, v_m^i \right\rangle$, $v_t^i \in \mathbb{R}$, $t \in \{1, \ldots, m\}$, $\forall p_i \in G$
   (b) $F^* \longleftarrow F^* \cup \{f^t\}$
```

Output: Feature set F^* over the sample set S.

3.5 Feature selection

To further improve model quality and reduce the feature-space dimensionality, feature selection is applied on the extracted features. InfoGain-based feature selection [38] is applied on the given feature set F of the samples set S, producing scores: $IG\left(S, f\right), \forall f \in F$. Then, only the features with positive InfoGain scores: $IG\left(S, f\right) > 0$ are selected.

The motivation: the features produced in Section 3.4 are based on pixel selection from Section 3.3, where the whole time-spectrum of pixels or pixel groups is preserved. However, points along the time course exist, during which the spatial discriminative nature is not realized (e.g. long before the onset of the signal in VSDI). Not only that these points in time are ineffective for the emphasis of the spatial characteristics, but they sometimes obscure their discriminating potential. InfoGain filtering drops those unwanted features with negligible scores, whose contribution is neutral or negative.

Chapter 4

Empirical evaluation of VSDI data

The primary goal in our work is to suggest a combination of effective techniques for obtaining scalable and accurate classification in large scale spatiotemporal domains. To reach this goal, we demonstrate how our techniques are evaluated in the VSDI domain and applied to VSDI datasets. The accuracy of the classification is validated by the evaluation of our classification performance. The scalability of our methods is shown by exploring their feasibility from the run-time perspective. This is done by emphasizing the lessons learned from the experience we had with applying approaches similar in nature to the ones reviewed in Chapter 2. Many of these approaches use the most granular values of the sample's space for feature selection and classification, which eventually leads to an extremely high dimensional feature space. Our failure in employing these approaches is compared to the success of showing the feasibility of our methodology. We additionally compare our results to those achieved by a domain expert faced with the same tasks.

4.1 Datasets

Each evaluated dataset is based on a single imaging experiment performed in the visual cortex of one animal and composed from multiple trials. In each experiment, the monkey was shown a set of different visual stimuli, one specific stimulus per trial. Each stimulus presentation was repeated 20-30 times. Neuronal population responses in the visual cortex evoked by the stimulus, were recorded using VSDI. The imaged area was divided into a grid of pixels, and population response (summed membrane potentials of all neuronal elements) of each pixel was recorded during the time window of the trial [30]. Each trial in an experiment is a sample in our sample space. A sample consists of all pixels of the recorded area, where a pixel is a time-series of values collected along the time course of the trial[1]. These values represent the rawest possible data-points—with no averaging across trials, whether in time or space, therefore directly reflecting unprocessed measurement points. Hence, the VSDI decoding we did was performed at a single trial level. Each sample is labeled with a class that represents the appropriate stimulus. The datasets differ in the number and the type of the presented stimuli, both affecting the complexity of the decoding. Being able to perform successful classification of these datasets, is being able to "read" what the monkey has seen without seeing it ourselves.

4.1.1 Dataset 1: Oriented Gratings (simple)

The monkey was presented with two different drifted square gratings at horizontal and vertical orientations, and a blank control image with no stimulus (Figure 4.1). Each of the 293 samples in the dataset had 2162 pixels (a 46 × 47 matrix) along 51 time points. The three classes had almost uniform distribution where the mode class constitutes 34.13% of the population (setting the baseline accuracy, i.e. ZeroR [38]).

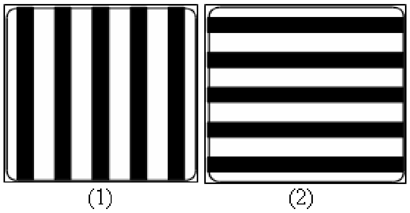


Figure 4.1: Stimuli for the Oriented Gratings dataset: (1) Drifted square gratings at vertical orientations; (2) Drifted square gratings at horizontal orientations; (3) Blank control image (not presented).

[1] Please refer to Section 4.2 for visualization examples of the raw VSDI data.

4.1.2 Dataset 2: Gabors (complex)

The monkey was presented with five different Gabor based orientations in space and a blank control image (Figure 4.2). Each of the 153 samples had 10, 000 pixels (a 100 × 100 matrix) along 51 time points. The six classes had almost uniform distribution where the mode class constitutes 18.95% of the population (ZeroR baseline accuracy).

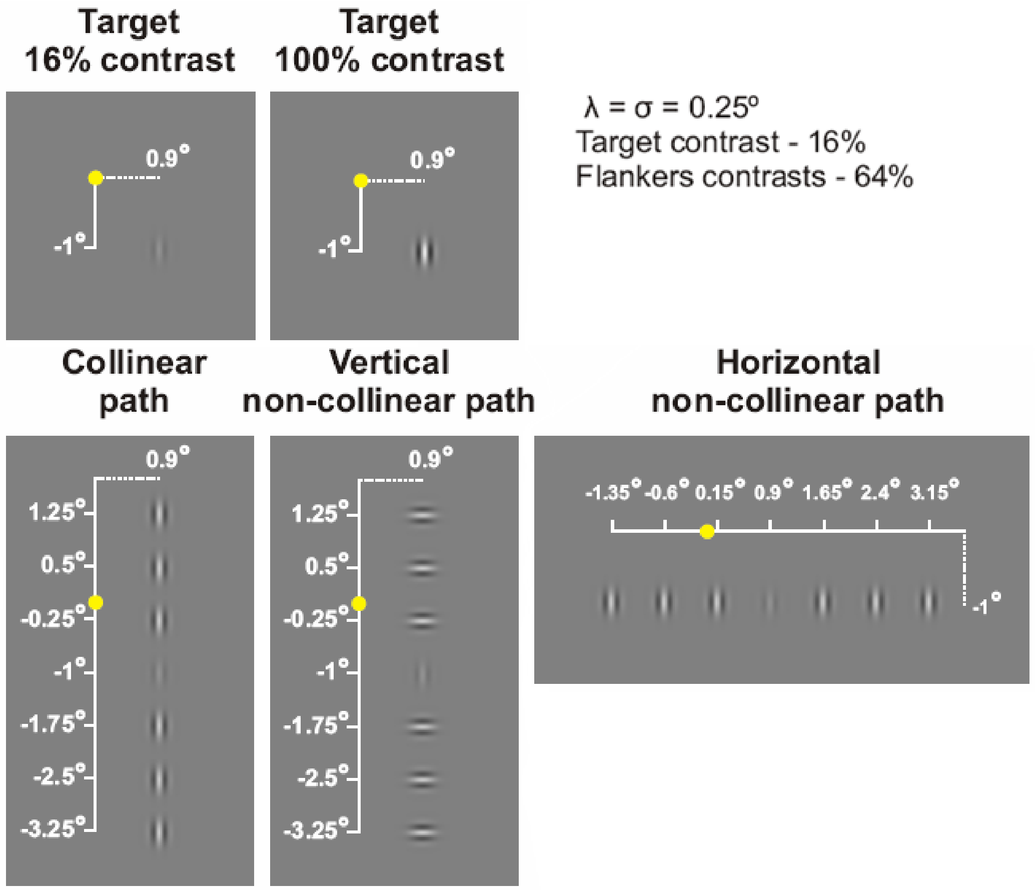
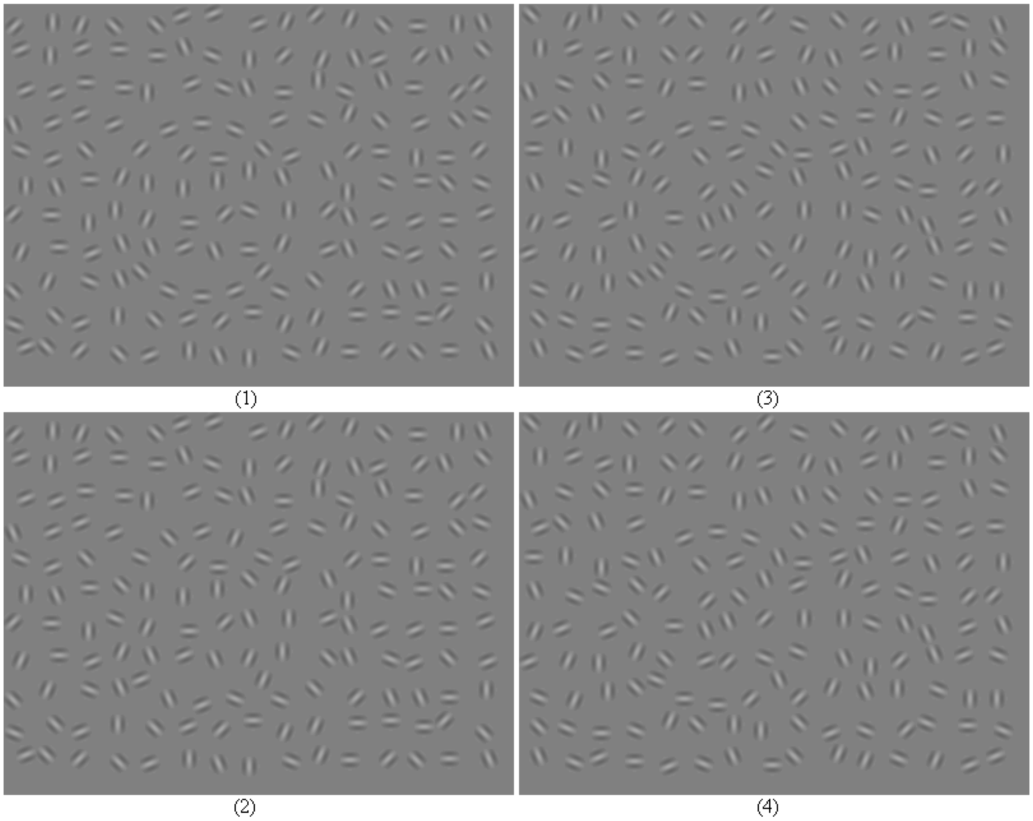


Figure 4.2: Stimuli for the Gabor dataset (the numbers and the degrees on the white axes are not part of the stimuli; blank control image not presented). The yellow point represents the fixation point location.

4.1.3 Dataset 3: Contours (hard)

The monkey was presented with four different Gabor-based Contours in space and a blank control image (Figure 4.3). The four Gabor-based Contours divide into two pairs, where the differences between the classes in each pair are very subtle and hardly noticeable. Each of the 124 samples had $10,000$ pixels (a 100×100 matrix) along 61 time points. The five classes had almost uniform distribution where the mode class constitutes 23.39% of the population (ZeroR baseline accuracy).



(1)

(3)

(2)

(4)

Figure 4.3: Example of stimuli for the Contour dataset: (1) Circle 1; (2) Masked circle 1; (3) Circle 2; (4) Masked circle 2; (5) Blank control image (not presented).

4.2 Raw VSDI data visualization

Visualizing data composed of thousands of pixels is not an easy task. Thus, we chose to present only a small number of pixels, taken from a single sample for each of the stimuli types, of the dataset which is the easiest for classification. In Figures 4.4, 4.5 and 4.6 below we display the values of $\sim 12\%$ from 2162 pixels close to the ROI_1 area of the Oriented Gratings dataset. Each Figure is based on a single sample out of 293 samples, each of

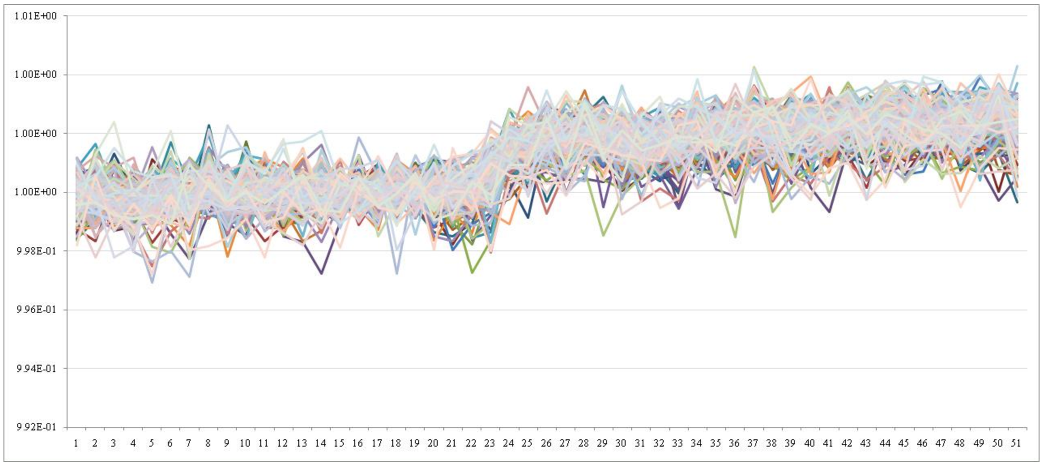


Figure 4.4: Visualization of a sample labeled with Class 1 from the Oriented Gratings dataset. Only the values of $\sim 12\%$ of the pixels close to the ROI_1 area are displayed along the full time course.

which represents one specific stimulus out of the three possible types. Visual comparison of the different stimuli samples of the Oriented Gratings by a naked eye, hints that the classification of this dataset will not be a hard task—which, in practice, turned out to be the case—unlike the two other datasets (whose data we chose not to visualize here for the same very reason).

4.3 Experiments methodology

4.3.1 Defining the *Oracle*

As a part of the evaluation methodology for the pixel selection technique presented in Section 3.3, we define the *Oracle*: a pixel selection method, a best-effort attempt by a human expert to provide a pixel set which, in her professional opinion, has the most potential to successfully discriminate between the different classes of the training samples set. The *Oracle* is asked to manually pick a set of pixels of some size: $\Omega = \{p_1, p_2, \ldots\}$, $\Omega \subseteq P$, also known as the ROI (Region Of Interest). This set is referred to as the "gold standard", where the aim is to build an accurate classification model using the most discriminating pixels. The success rates achieved by using Ω for pixel selection are compared to the

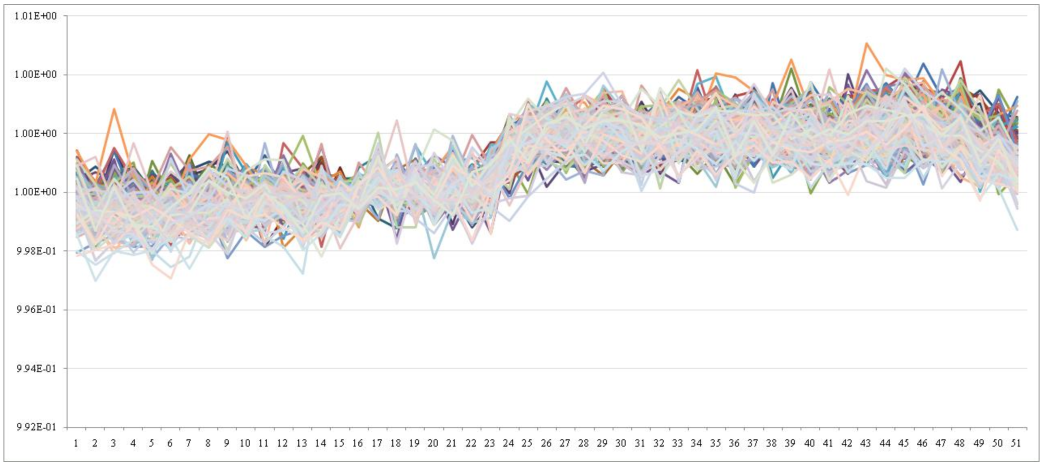


Figure 4.5: Visualization of a sample labeled with Class 2 from the Oriented Gratings dataset. Only the values of $\sim$ 12% of the pixels close to the ROI_1 area are displayed along the full time course.

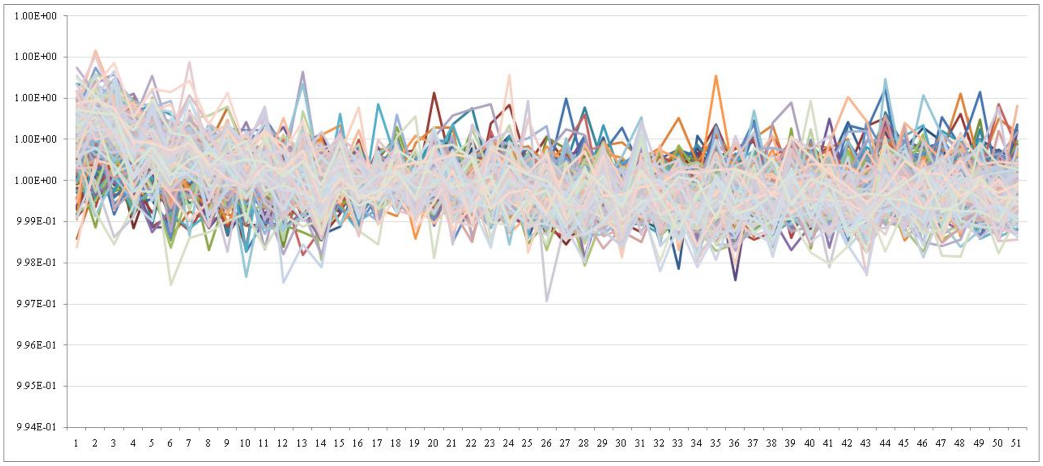


Figure 4.6: Visualization of a sample labeled with Class 3 (blank control image) from the Oriented Gratings dataset. Only the values of $\sim$ 12% of the pixels close to the ROI_1 area are displayed along the full time course.

success rates of using the pixel set selected by $GIRSS$, our pixel selection technique.

4.3.2 Experimental setup

In the experimental setup, the domain expert was requested to provide an ROI(s) of pixels for each dataset:

- For the Oriented Gratings dataset, we were given a single ROI along the time course of our experiments, the ROI_1.
- In the case of the Gabors, after the first line of experiments with ROI_1—the original ROI— we were given ROI_2, an improved version based on the results of using ROI_1[2].
- With the Contours case, three different ROIs of pixels were given in advance, each for individual evaluation by our techniques.

We constructed classification models using both pixel selection techniques in the first phase, in combination with the two feature extraction techniques in the second phase: $\{Oracle, GIRSS\} \times \{PIT, IPCOSA\}$, and with the application of the feature selection (Section 3.5) in the third phase. Figure 4.7 sketches the outline of our experimental methodology model construction. The resulting models were evaluated using a 10-fold cross-validation of the multi-class SMO implementation of SVM with linear kernel [38]. Each model's evaluation was performed a number of times (each trial yielding a different random 10-fold division), as specified in the results Table 4.1.

4.4 Results

The results are presented in Table 4.1. The table is divided into three sections where each section presents the experimental results for each of the VSDI datasets. Inside each section, the results are divided by the method used for the pixel selection phase—the $GIRSS$ technique and the $Oracle$ approach. For each of the compared pixel selection methods, displayed are the accuracy results of combining the method with each of the feature extraction techniques—the PIT and the $IPCOSA$ (refer to the table caption

[2] Please see results of ROI_2 pixel set of Gabors dataset in Table 4.1, compared to the results of ROI_1.

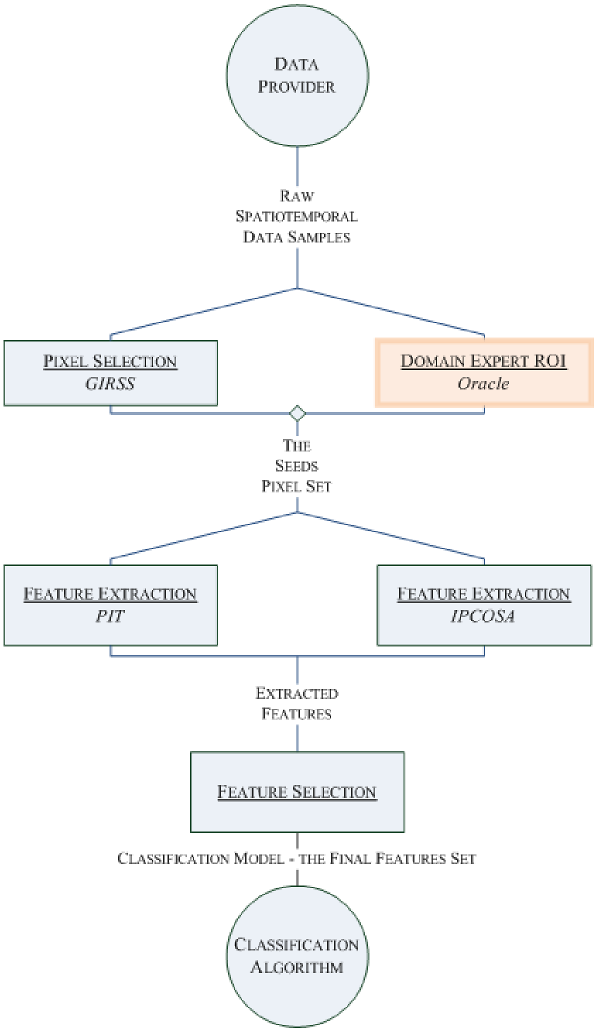


Figure 4.7: The outline of the experimental methodology model construction. An additional *Oracle* node is added to the classification modeling methodology outlined in Figure 3.1. With the help of the *Oracle*, we evaluate the performance of our pixel extraction technique—the *GIRSS*.

for the legend of the table entries). These results are discussed in depth in the next subsections.

Table 4.1: The results of applying each of the combinations: $\{Oracle, GIRSS\} \times \{PIT, IPCOSA\}$ on each dataset. $ROI_x\,(y)$ is an ROI pixel set x of size y. The numbers in brackets for u (number of pixels in a random pixel subset) and r (number of random pixel subsets) are their respective values. The results of the form $\mu \pm \sigma\%\,(n)$ have μ representing the average accuracy between the trial runs, σ representing the standard deviation and n representing the number of trial runs. The entries in bold represent the best accuracies obtained per each pixel selection method of each of the datasets.

ORIENTED GRATINGS

BASELINE: 34.13%

	Oracle	*GIRSS*
	$ROI_1\,(154)$	$u\,(154)\,,r\,(20)$
PIT	$\mathbf{95.4 \pm 0.4\%\,(10)}$	$\mathbf{94.9 \pm 1.3\%\,(10)}$
IPCOSA	$79.3 \pm 3.2\%\,(10)$	$88.5 \pm 4.0\%\,(10)$

GABORS

BASELINE: 18.95%

		Oracle	
	$ROI_1\,(104)$	$ROI_2\,(218)$	
PIT	$55.0 \pm 1.5\%\,(10)$	$68.8 \pm 1.1\%\,(10)$	
IPCOSA	$57.2 \pm 3.3\%\,(10)$	$\mathbf{71.0 \pm 2.4\%\,(10)}$	
		GIRSS	
	$u\,(100)\,,r\,(150)$	$u\,(100)\,,r\,(125)$	$u\,(100)\,,r\,(100)$
PIT	$79.1 \pm 1.7\%\,(10)$	$78.4 \pm 2.1\%\,(10)$	$79.0 \pm 1.8\%\,(10)$
IPCOSA	$\mathbf{81.8 \pm 1.4\%\,(10)}$	$81.8 \pm 2.3\%\,(10)$	$80.7 \pm 1.7\%\,(10)$

CONTOURS

BASELINE: 23.39%

		Oracle	
	$ROI_1\,(151)$	$ROI_2\,(227)$	$ROI_3\,(155)$
PIT	$44.9 \pm 2.3\%\,(10)$	$50.6 \pm 2.4\%\,(10)$	$\mathbf{73.3 \pm 1.6\%\,(10)}$
IPCOSA	$40.2 \pm 3.1\%\,(10)$	$47.6 \pm 3.0\%\,(10)$	$65.7 \pm 1.8\%\,(10)$
		GIRSS	
	$u\,(151)\,,r\,(100)$	$u\,(500)\,,r\,(100)$	
PIT	$71.9 \pm 2.8\%\,(10)$	$69.6 \pm 2.8\%\,(10)$	
IPCOSA	$72.4 \pm 2.7\%\,(10)$	$\mathbf{73.1 \pm 2.1\%\,(10)}$	

4.4.1 Performance evaluation

In regard to the classification performance, besides aspiring to achieve the most accurate results, it also was as much as important for us to show that the results we acquire are not inferior to the ones achieved by exploiting the domain expert's guidelines. Indeed, we have shown that for two types of data (Oriented Gratings, Contours), our pixel selection technique is capable of producing pixel sets having as good discriminative abilities as the best of provided ROI sets. Moreover, for the Gabors data type, our results were superior not only to the initially provided ROI_1, but also to the revised ROI_2. The high accuracy of the Oriented Gratings dataset is somehow expected due to the apparent differences between its visual stimuli, but it is not for granted considering the baseline of 34.13%. The accuracies in running $GIRSS$ on the other two datasets provide results considerably above the baselines.

4.4.2 $GIRSS$ vs. the ROI

With the Gabors case, we see major differences when our selected pixel sets are compared to the ROI pixel sets. These differences are expressed by comparing the Figure 4.8 to Figures 4.9 and 4.10. While both of the ROI sets (and specifically the improved ROI_2) were defined within the V1 area (primary visual cortex), our sets (of the same size) show a wide spread of pixels among numerous sites, including V2 (secondary visual cortex). One can claim that the comparison is not adequate, since the ROI was limited to V1. Nevertheless, we claim the opposite—our results reveal that while the initial working hypothesis of a neuroimaging expert can be restricted to a specific cortical site (e.g. V1 activity is sufficient for decoding the Gabors' visual stimuli), in practice, a collaboration between the representative populations from numerous sites shows much higher contribution to classification.

Major differences with resembling characteristics can be also observed in the Contours case. Here, the pixels selected by $GIRSS$ cover a wide area spread, again, along multiple sites in the visual cortex. The comparison of ROI_3, the best performing Contours ROI, to the results of our technique, is demonstrated in Figure 4.11 vs. the Figures 4.12 and 4.13.

With the rather simplistic case of the Oriented Gratings, where the types of the visual stimuli are noticeably different, and the decoding task at hand is far from being complex, the pixel sets produced by $GIRSS$ show a wide spread across the cortical area (which is almost five times smaller than the tested area in the other datasets). This spread of

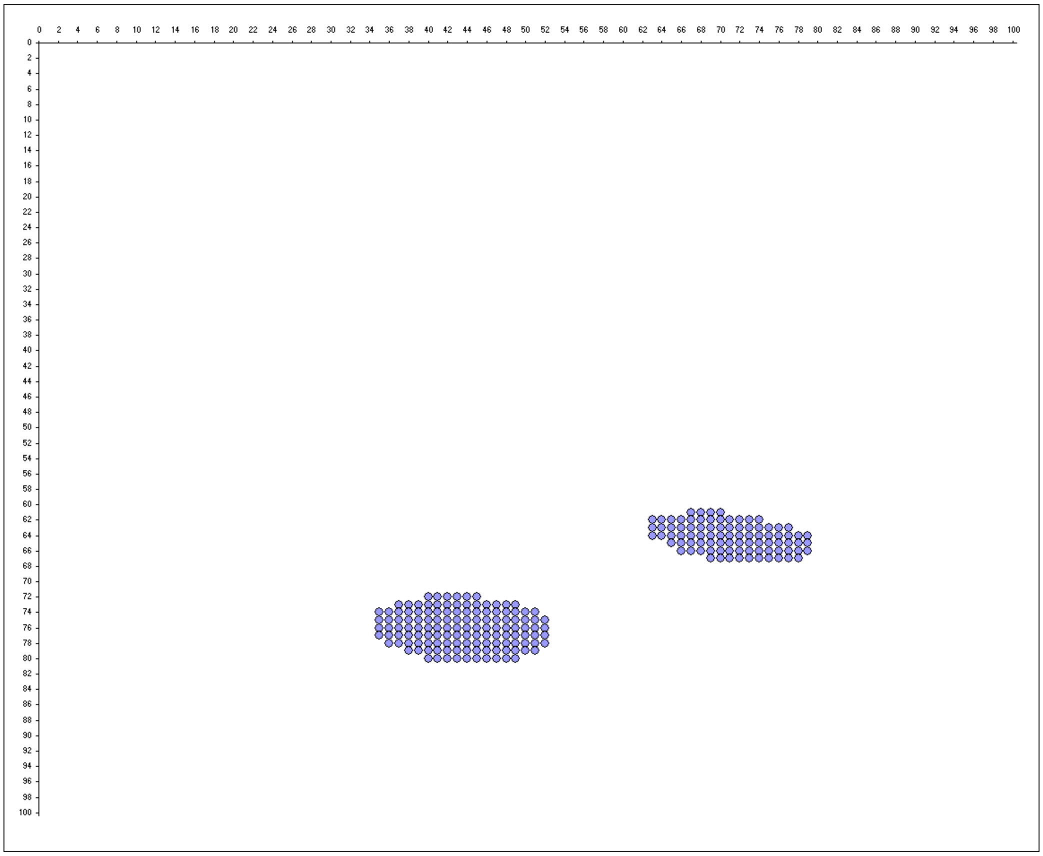


Figure 4.8: Gabors dataset, $ROI_2(218)$—the best performing *Oracle*'s ROI pixel set. The imaged area of pixels is depicted on the grid (all pixels are in V1).

pixels, being 7.1% of the complete pixel range, seems to almost uniformly cover the whole area. Comparing the accuracies of the pixels selected by *GIRSS* to those provided in the ROI (concentrating in one specific spot), we reveal approximately the same high accuracy results. These findings only strengthen the claim that the Oriented Gratings dataset is a comparatively easy task. The differences are shown in Figures 4.14 and 4.15.

4.4.3 Spatial averaging contribution

Due to the high resolution of the signal in the Oriented Gratings, we see that the spatial averaging only worsens the results instead of improving them. This is an expected result—the signal in this case arises from small orientation columns, while averaging over space smears them out, causing the loss of signal—hence, the loss of the data's essential

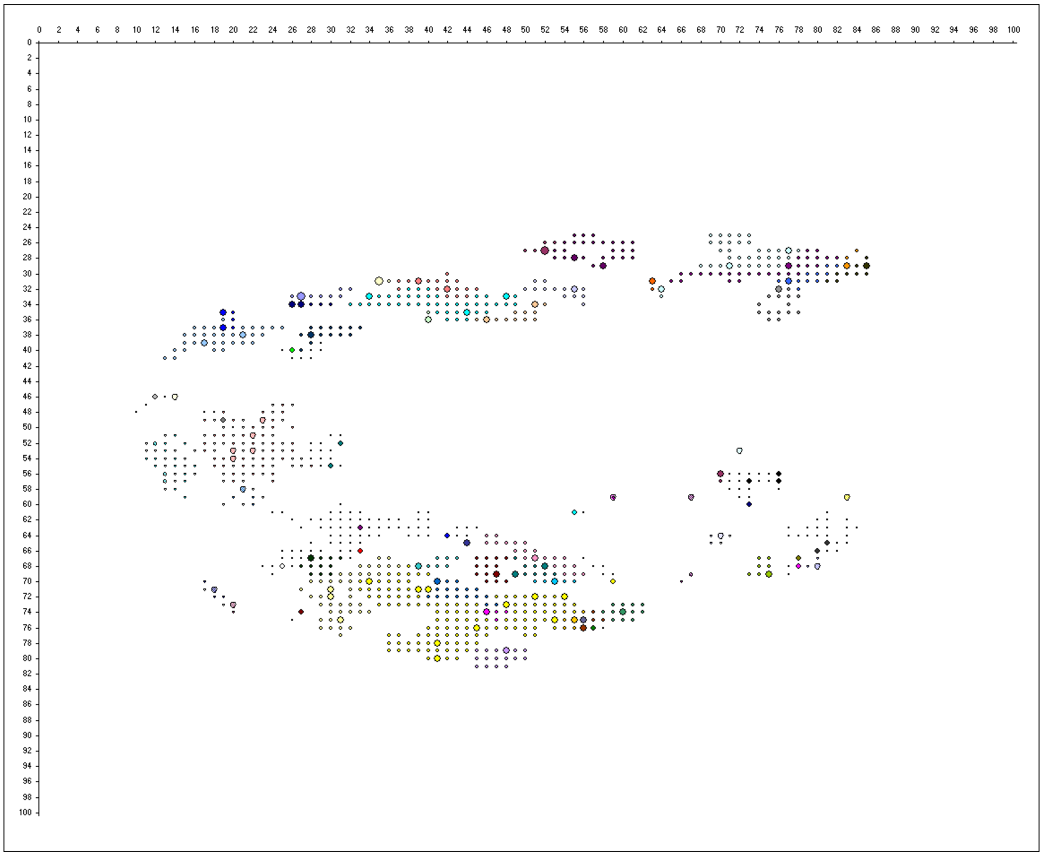


Figure 4.9: Gabors dataset, sample fold result—the imaged area of pixels depicted on the grid. Applied *GIRSS* with $u(100)$ and $r(125)$ to produce "seeds" pixel set (large circles). Applied *IPCOSA* to improve the accuracy of *PIT* from 81.25% to 93.75%. Neighborhoods of pixels for averaging are formed around the seeds (small circles, having the seeds' colors). The different sizes of pixels between the neighborhoods express the strength of the inter-correlation within each neighborhood, compared to the other ones.

properties. However, with the Gabors and the Contours, we see quite the opposite—spatial averaging provides additional enhancement to the classification abilities. Being much harder to distinguish than with the first dataset case, the types of the visual stimuli of these two datasets lead to collection of data in which the activation has, at least partially, low spatial frequency characteristics, as opposed to the Oriented Gratings (some of the information in this case has to do with the retinotopic activation). In conclusion, the spatial averaging role depends on the size of the neuronal spatial modules that encode it, leaving space for improvement by the advanced feature extraction technique in datasets characterized by low spatial frequency.

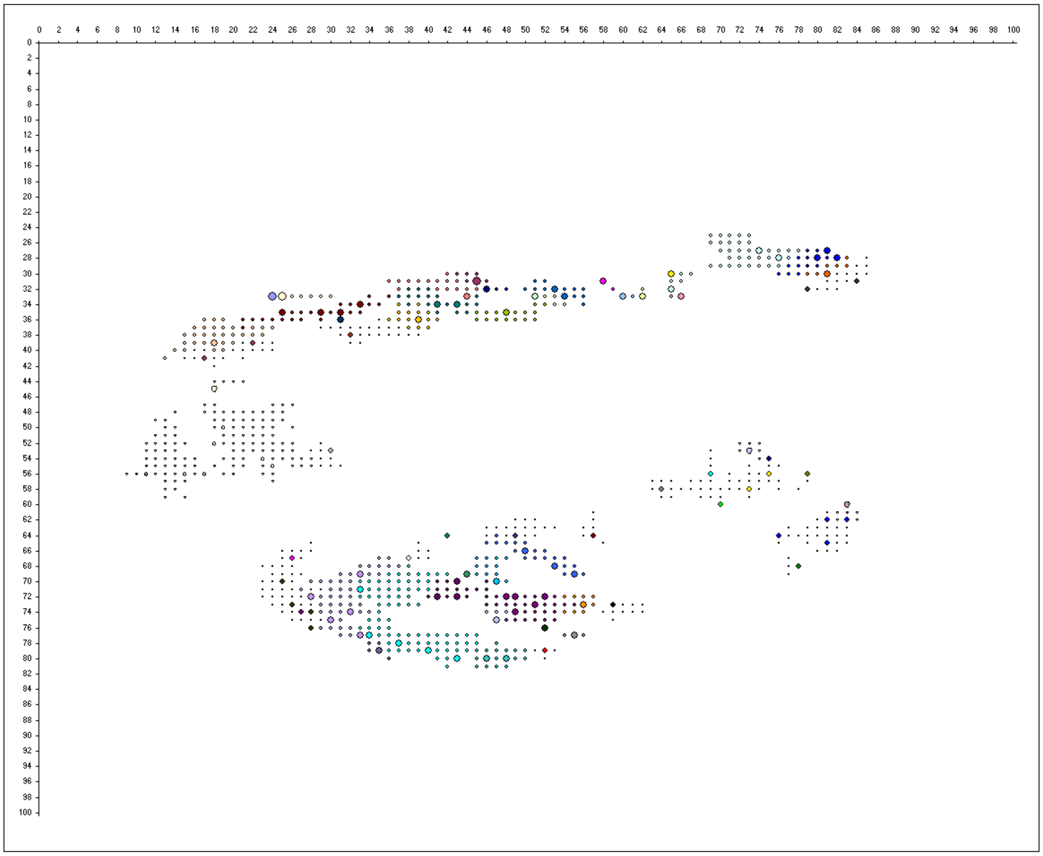


Figure 4.10: Gabors dataset, sample fold result—the imaged area of pixels depicted on the grid. Applied $GIRSS$ with $u(100)$ and $r(125)$ to produce "seeds" pixel set (large circles). Applied $IPCOSA$ to improve the accuracy of PIT from 87.67% to 100%. Neighborhoods of pixels for averaging are formed around the seeds (small circles, having the seeds' colors). The different sizes of pixels between the neighborhoods express the strength of the inter-correlation within each neighborhood, compared to the other ones.

4.4.4 Validation of the results

To further establish the credibility of our results and disproof the likelihood of "free of charge" high accuracy rates or of possible overfitting, we proceeded with additional validation of the results produced by using our three phase methodology. In VSDI data in particular, the significance of each of the stimuli conditions is realized only after the visual stimulus onset, that is to say—the discrimination between the different stimuli (i.e. the classification of the different classes), is only possible after the stimuli were shown to the monkey, and the appropriate neuronal population responses were provoked. Had we

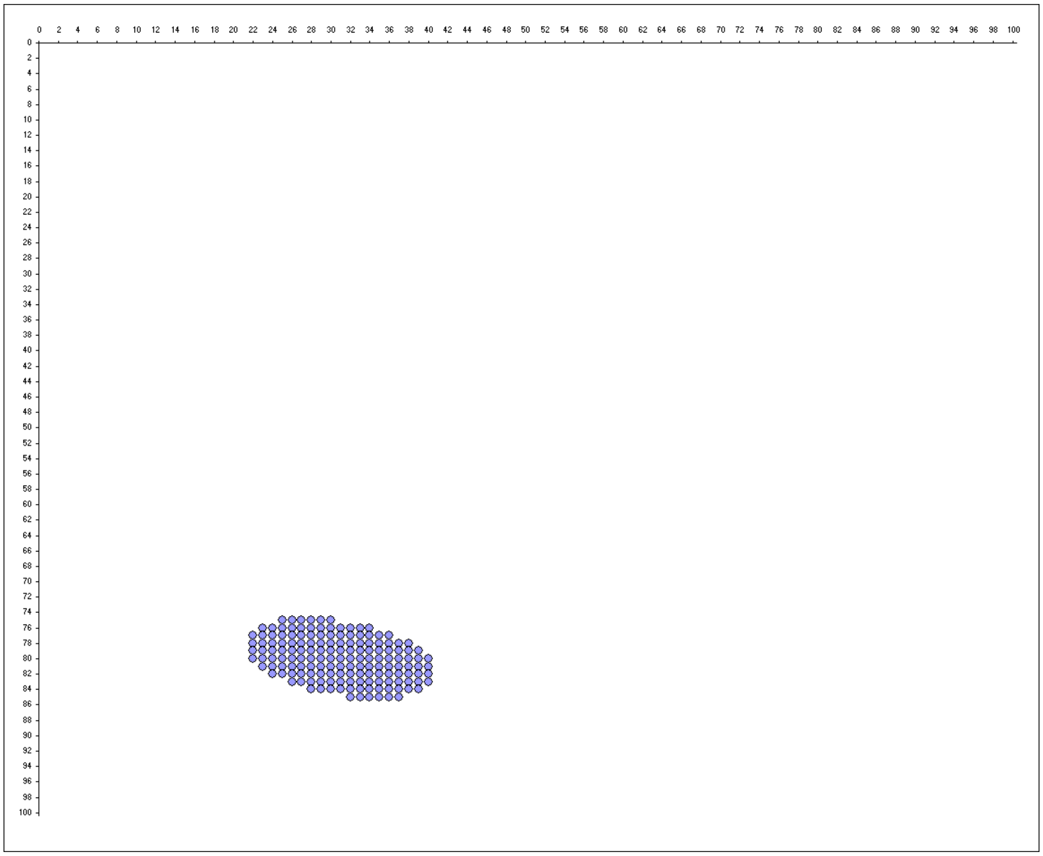


Figure 4.11: Contours dataset, $ROI_3(155)$—the best performing *Oracle*'s ROI pixel set. The imaged area of pixels is depicted on the grid.

observed the responses of the same neuronal populations, solely before the onset of the stimuli, we would not expect to have the ability to discriminate between them—simply because of the fact that the behavior of these responses is expected to be similar to the ones provoked by the blank control image, where no stimulus is presented (which is exactly the case).

The logic discussed above lays the foundations of our validation procedure. We carried the same experiments as detailed in Section 4.3, with two differences. First, in all our datasets, the time course was reduced to only the first consecutive points in time where we know for sure that the onset of the stimuli was not present. Second, pixel selection via the *Oracle* was not included in this procedure—knowing that *GIRSS* has at least as good classification capabilities as the *Oracle*, such type of comparison at this stage is

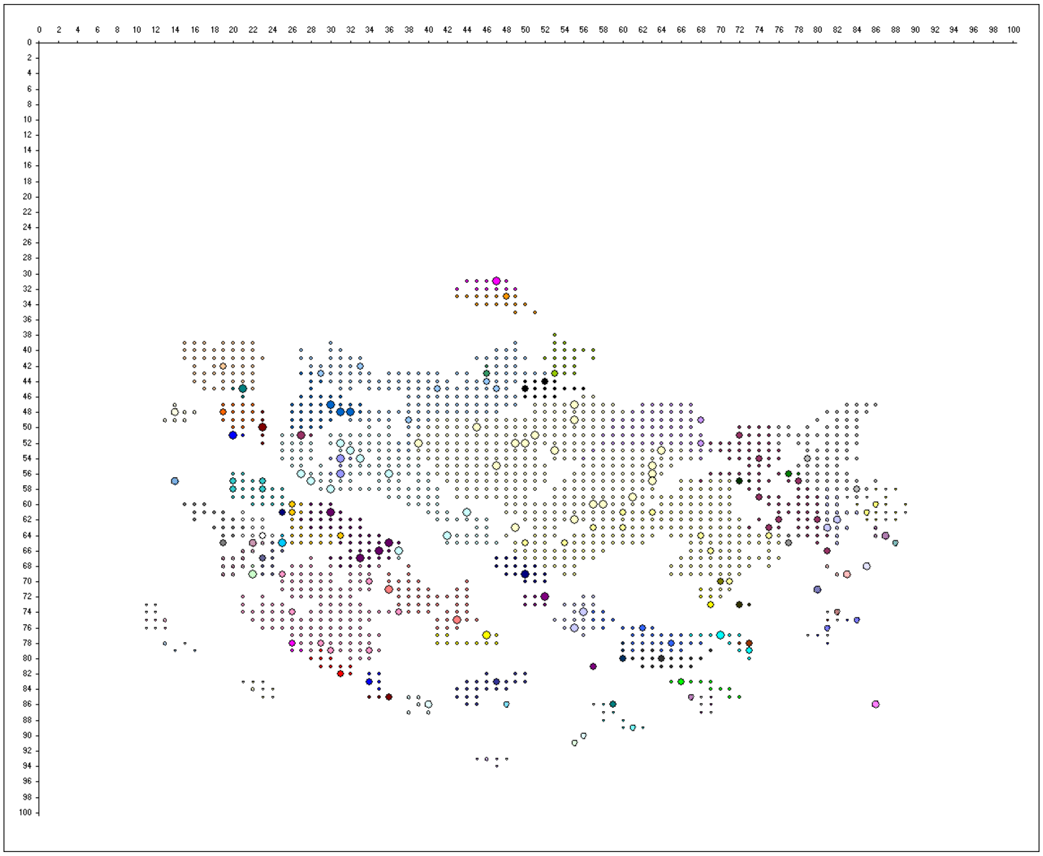


Figure 4.12: Contours dataset, sample fold result—the imaged area of pixels depicted on the grid. Applied *GIRSS* with $u(151)$ and $r(100)$ to produce "seeds" pixel set (large circles). Applied *IPCOSA* to improve the accuracy of *PIT* from 69.23% to 84.62%. Neighborhoods of pixels for averaging are formed around the seeds (small circles, having the seeds' colors). The different sizes of pixels between the neighborhoods express the strength of the inter-correlation within each neighborhood, compared to the other ones.

redundant.

That being the case, we would expect the classification results to be close to baseline accuracies of each of the datasets. These results are presented in Table 4.2. The table is divided into three sections where each section presents the experimental results for each of the VSDI datasets. Inside each section, displayed are the accuracy results of combining the *GIRSS* pixel selection method with each of the feature extraction techniques—the *PIT* and the *IPCOSA* (refer to the table caption for the legend of the table entries). Indeed, we can safely say that the results of this stage are as expected—roughly the same as the chance level.

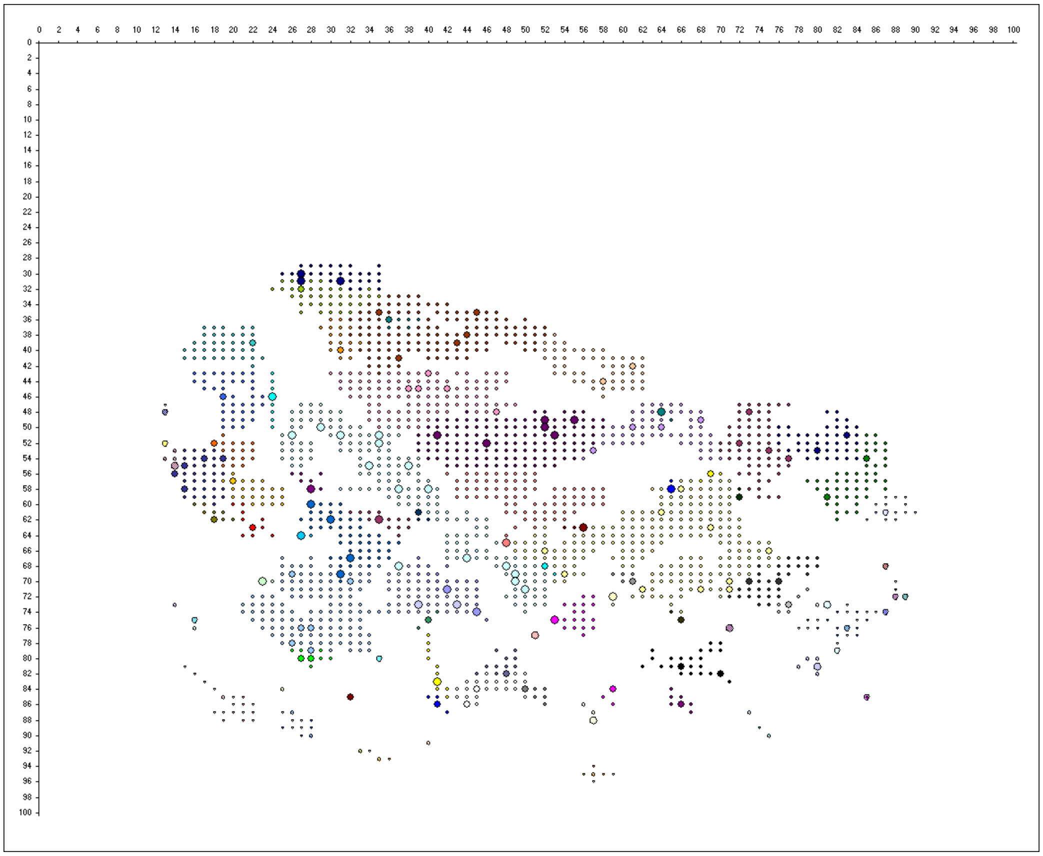


Figure 4.13: Contours dataset, sample fold result—the imaged area of pixels depicted on the grid. Applied $GIRSS$ with $u(151)$ and $r(100)$ to produce "seeds" pixel set (large circles). Applied $IPCOSA$ to improve the accuracy of PIT from 84.62% to 92.31%. Neighborhoods of pixels for averaging are formed around the seeds (small circles, having the seeds' colors). The different sizes of pixels between the neighborhoods express the strength of the inter-correlation within each neighborhood, compared to the other ones.

4.4.5 Scalability and feasibility

All early attempts to handle the data before basing our pixel selection on random subspace [15], such as employing techniques that base their feature extraction, selection and classification on the full spatiotemporal range (resembling methods proposed in [17, 19, 22, 35]), ended with impractical running times, I/O and memory requirements.

Examples for the said above include, among the rest, the following scenario: The ROI_1 set of the Gabors dataset, which includes 104 pixels, was turned into a feature-set using the PIT approach (Subsection 3.4.1), resulting in 5304 features. The features were ranked

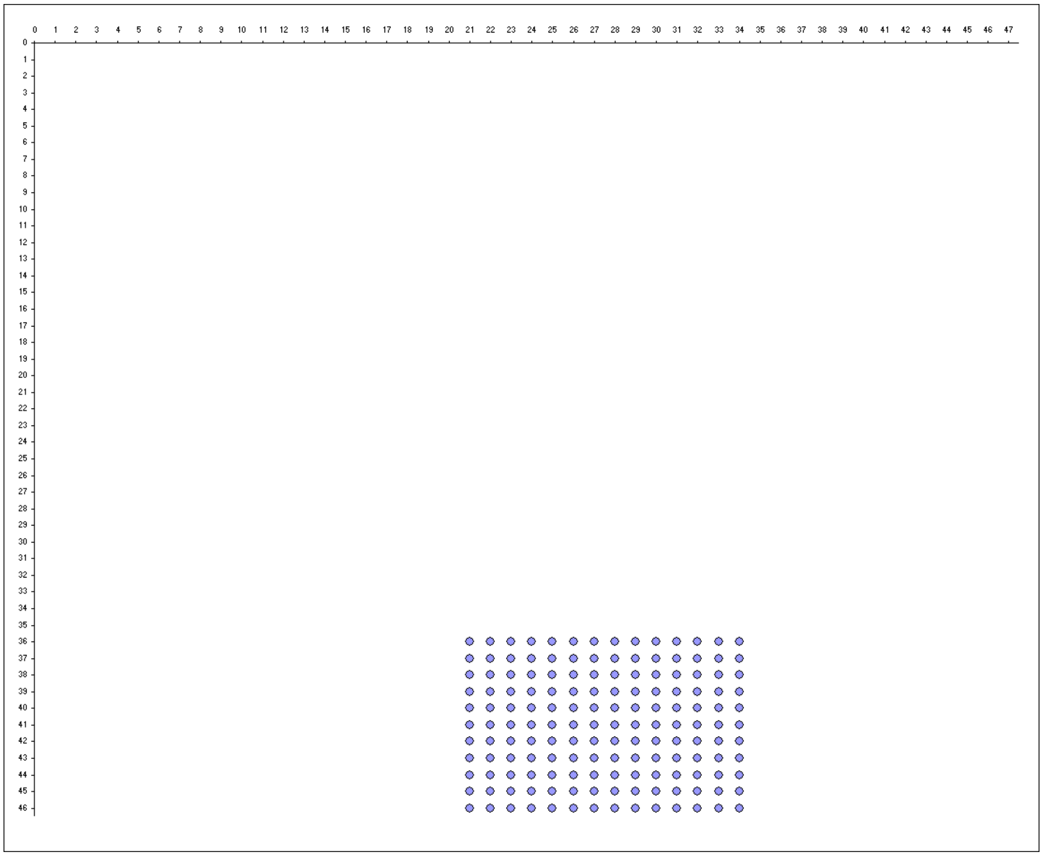


Figure 4.14: Oriented Gratings dataset, $ROI_1(154)$—the single *Oracle*'s ROI pixel set. The imaged area of pixels is depicted on the grid.

with InfoGain scores (as in Algorithm 3.3, step 2b), and sorted from top down—with the lowest ranked feature at the bottom of the list. Iteration over the feature set was performed in an RFE-like [6] manner (but with a different feature weighting mechanism): first, the whole feature set was preserved, the dataset was trained using the SMO classifier, either InfoGain filtered as in Section 3.5 or not filtered at all, and 10-fold cross-validated; the evaluation score of this step was written down. In the second iteration, the lowest ranked feature was removed from the feature set, training and validation was repeated, and a new evaluation score was written down. In this recursive manner, the process was repeated for all the features, until the last and only, top ranked feature, remained in the feature set. The number of top-ranked features to choose for the model construction was the one with the highest evaluation score collected during the execution of this procedure.

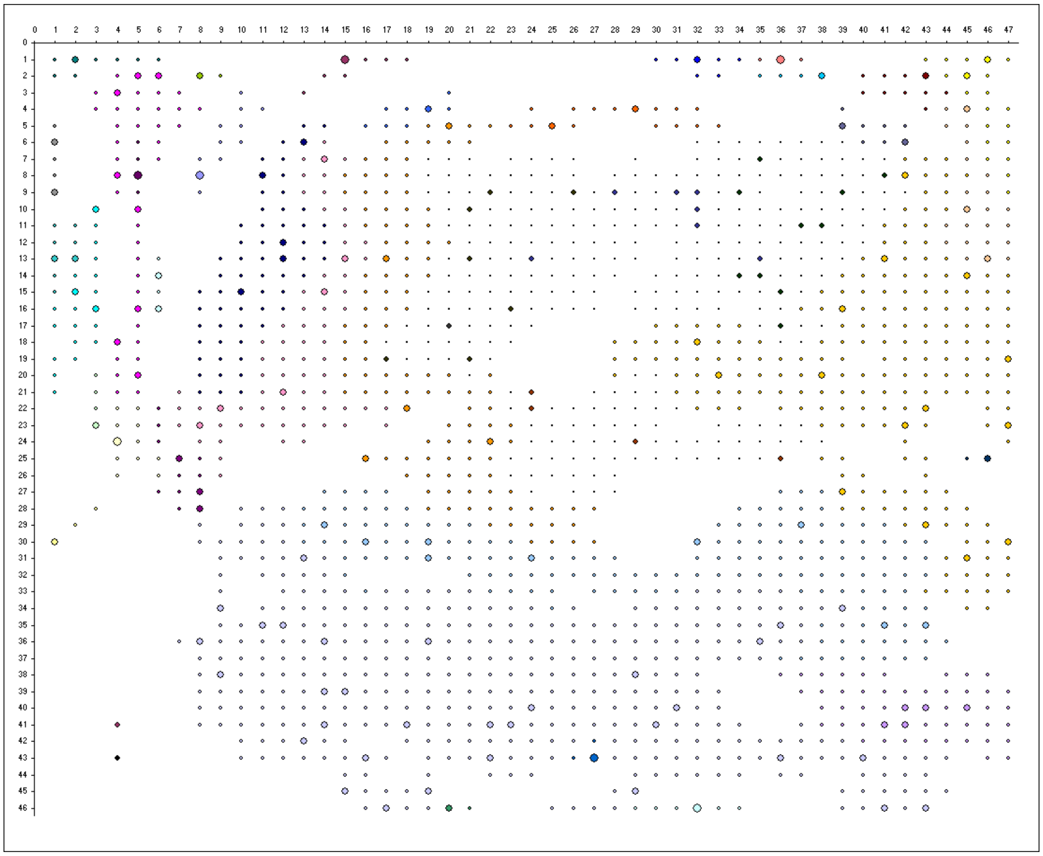


Figure 4.15: Oriented Gratings dataset, sample fold result—the imaged area of pixels depicted on the grid. Applied $GIRSS$ with $u(154)$ and $r(20)$ to produce "seeds" pixel set (large circles). Applied $IPCOSA$ to improve the accuracy of PIT from 96.55% to 100%. Neighborhoods of pixels for averaging are formed around the seeds (small circles, having the seeds' colors). The different sizes of pixels between the neighborhoods express the strength of the inter-correlation within each neighborhood, compared to the other ones. This is a rare case where $IPCOSA$ improves accuracy for Oriented Gratings.

Without judging the quality or the motivation for the above scenario, the magnitude of its running times is roughly the same as of the methods having a resembling nature, such as [19,22]—where a classifier is trained for each pixel-time combination during the process. The running times of this scenario applied to the Gabors ROI_1 is between 25 to 50 hours, depending on whether the InfoGain filtering was applied. Had we managed to run this scenario on the full Gabors pixel set rather than on ROI_1, this would at first seem like it would have taken between 10 to 20 days; however, this relation is not linear—although these estimates are for classifier training and evaluation based on 10000 pixels, as opposed

Table 4.2: The *validation procedure* results of applying each of the combinations: $\{GIRSS\} \times \{PIT, IPCOSA\}$ on each dataset, with the time course reduced to the first 10 points in time before the visual stimulus onset for the Oriented Gratings and the Gabors, and the first 3 points in time for the Contours. The numbers in brackets for u (number of pixels in a random pixel subset) and r (number of random pixel subsets) are their respective values. The results of the form $\mu \pm \sigma\%\,(n)$ have μ representing the average accuracy between the trial runs, σ representing the standard deviation and n representing the number of trial runs. The entries in bold represent the best accuracies obtained per each dataset.

ORIENTED GRATINGS

BASELINE: 34.13%, CHANCE: 33.33%

	$GIRSS$
	$u\,(154)\,,r\,(20)$
PIT	$31.6 \pm 1.8\%\,(5)$
$IPCOSA$	$\mathbf{34.0 \pm 1.3}\%\,(\mathbf{5})$

GABORS

BASELINE: 18.95%, CHANCE: 16.67%

	$GIRSS$
	$u\,(100)\,,r\,(150)$
PIT	$\mathbf{17.5 \pm 1.6}\%\,(\mathbf{5})$
$IPCOSA$	$15.4 \pm 2.1\%\,(5)$

CONTOURS

BASELINE: 23.39%, CHANCE: 20.00%

	$GIRSS$	
	$u\,(151)\,,r\,(100)$	$u\,(500)\,,r\,(100)$
PIT	$21.0 \pm 2.3\%\,(5)$	$19.0 \pm 2.9\%\,(5)$
$IPCOSA$	$\mathbf{22.7 \pm 0.6}\%\,(\mathbf{5})$	$22.4 \pm 0.6\%\,(5)$

to only 104, they are based on a recursion that starts from 104 pixels only. A more correct estimate would be based on running times of $\sim$ 30 minutes per pixel on average, resulting in an initial estimate of, and easily surpassing, 200 days.

Moreover, we were not even able to run this scenario, neither a few other ones having a resembling pixel-time pairs based iterative nature, due to the impractical I/O and memory requirements. The basic instances initialization during the initial loading of the Gabor dataset would take tens of minutes due to an intensive I/O, only to crash later on insufficient memory (albeit using a 32 bit architecture OS); this would happen before completing the initialization—not to speak of moving to the next step of basic low profile operations such as InfoGain based filtering. While the Contours dataset has about the same impractical magnitudes, with Oriented Gratings the experience is slightly different. Here, the loading of a dataset based on only 2 out of 3 available classes, having a 5 times smaller pixels number, but having twice as bigger number of samples, would succeed after less than a few minutes. After additional few minutes of cross-validation though, approximately at the 5th fold, the process would crash—yet again—on insufficient memory.

Finally, the memory obstacle remains relevant even if we remove the run-time challenges of training classifiers for each pixel-time combination. It is enough to see that we cannot load the initial data based on all the available raw values, even before moving to any feature selection or classifier training steps.

However, with $GIRSS$ and $IPCOSA$ we were able to build models using a single-threaded Java application on a Core 2 Duo machine with 2GB of RAM, in less than 2 hours for the Oriented Gratings, roughly 8 hours for the Gabors, and between 8 to 13 hours for the Contours datasets. Using the PIT instead of the $IPCOSA$ lowers these times by up to an order of magnitude. Truly, our proposed models are not only feasible, but practical.

4.5 Seed based spatial averaging comparison

When designing our feature extraction technique, the $IPCOSA$, presented in Subsubsection 3.4.2.1, we examined a number of spatial averaging tools for calculating the average value. The question of how to calculate the average of a pixel group having one or more seeds within it, and formed in a phase that has a particular graded correlation threshold (the τ), was examined in six different ways, and compared.

Each pixel group is a pixel set $P^{'} = \left\{p_1^{'}, p_2^{'}, \ldots\right\}$. The seeds group is a pixel subset $P^* = \{p_1^*, p_2^*, \ldots\}$ of $P^{'} : P^* \subseteq P^{'}$. The group $P^- = \left\{p_1^-, p_2^-, \ldots\right\}$ is defined as: $P^- = P^{'} \setminus P^*$. Each pixel p in any of the defined sets has only one value in time: $\overline{p} = \langle v_T \rangle$, $v_T \in \mathbb{R}$, $T \in \{1, \ldots, m\}$ (since the averaging is done in a specific fixed point in time T), so for simplicity we will refer to the pixel's single value in time using the pixel's notation (e.g. p will denote v_T). The averaging methods are:

1. Plain average: $\frac{\sum_{p^{'} \in P^{'}} p^{'}}{|P^{'}|}$

2. Weighted average: $\frac{\sum_{p^* \in P^*} p^* + \tau \sum_{p^- \in P^-} p^-}{|P^*| + \tau |P^-|}$

3. Square weighted average: $\frac{\sum_{p^* \in P^*} p^* + \tau^2 \sum_{p^- \in P^-} p^-}{|P^*| + \tau^2 |P^-|}$
4. Group and seeds splitted weighted average: $\frac{\frac{|P^-|}{|P^*|} \sum_{p^* \in P^*} p^* + \sum_{p^- \in P^-} p^-}{2|P^-|}$
5. Thresholded group and seeds splitted weighted average: $\frac{\frac{|P^-|}{|P^*|} \sum_{p^* \in P^*} p^* + \tau \sum_{p^- \in P^-} p^-}{(1+\tau)|P^-|}$
6. Seeds only average: $\frac{\sum_{p^* \in P^*} p^*}{|P^*|}$

The results of the comparison are presented in Figure 4.16, depicting the average accuracies of methods applied on each of the 7 experimental runs executed on the Gabors dataset (the results slightly differ from the presented in Table 4.1 due to variations in the r parameter). The selected method is the plain average method.

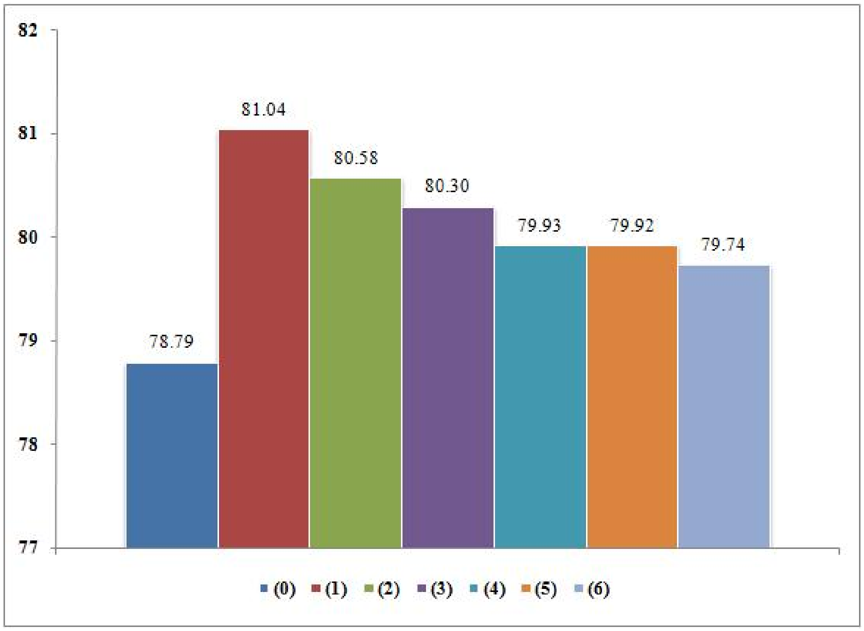


Figure 4.16: Comparison of seed based pixel groups averaging methods: (0) Accuracy of applying the PIT; (1) Accuracy of applying the $IPCOSA$ with plain average; (2) Accuracy of applying the $IPCOSA$ with weighted average; (3) Accuracy of applying the $IPCOSA$ with square weighted average; (4) Accuracy of applying the $IPCOSA$ with group and seeds splitted weighted average; (5) Accuracy of applying the $IPCOSA$ with thresholded group and seeds splitted weighted average; (6) Accuracy of applying the $IPCOSA$ with seeds only average.

4.6 Sliding windows: averaging along the time course

During our work, we performed various experiments with a time-course reduction using a simple Sliding Window (SW) technique. The motivation for this kind of work comes from two reasons. First, applying SW technique is appropriate with datasets in which the time dimension poses a dimensionality threat. In such cases, effective reduction of the time dimension makes the classification modeling more feasible. In addition, this technique can reduce the influence of a noisy data along the time course. However, the first reason was irrelevant, having the number of time points in our data significantly lower than the number of pixels. In regard to the second reason, one of the experimental techniques is detailed below.

When using a SW, two parameters are defined: the time window size w and the overlap factor o. The w specifies the number of consecutive points in time along which the averaging is performed. The o specifies the extent of the overlap between each two consecutive time windows. For instance, having $w = 3$ and $o = 1$, indicates averaging in the following manner:

$$average\left(timepoint_1, timepoint_2, timepoint_3\right)$$
$$average\left(timepoint_3, timepoint_4, timepoint_5\right)$$
$$\vdots$$
$$average\left(timepoint_{m-4}, timepoint_{m-3}, timepoint_{m-2}\right)$$
$$average\left(timepoint_{m-2}, timepoint_{m-1}, timepoint_m\right)$$

Note that the following will always hold: $w \geq 2$, $o \leq w - 1$; we have also defined that overlap must exist: $o \geq 1$.

Given the 51 time points in the Gabor dataset, we have tried every possible combination conforming to the following definition: $2 \leq w \leq 10$, $1 \leq o \leq w - 1$ (resulting in 45 different combinations). We then randomly generated 70 pixel subsets, each having a 100 pixels in a subset. For each of these subsets, and for each of its SW variations (resulting in

$(1 + 45) \cdot 70 = 3220$ of both regular and time-averaged pixel subsets), we have applied a slight variation of our pixel set evaluation method (Algorithm 3.2), one that handles the values averaged over the time course. The evaluation scores produced by this method were then compared and analyzed, and a comparison between the "regular" (baseline) pixel subsets and the time-averaged pixel subsets was made.

The following conclusion was reached: while the sliding window can significantly improve the evaluation score of an arbitrary pixel subset by up to 11.1% (the improvement gain), this improvement will be significant only for the weaker pixel subsets—the ones that in the first place, prior to averaging, were producing low evaluation scores. In fact, the higher the evaluation score of a baseline pixel subset was, the lower was the gain in accuracy (the delta) of applying any of the 45 sliding window variations. The highest ranked baseline pixel subsets do not benefit from the application of the sliding window. To support this claim, we checked the correlation between the evaluation scores of the baseline pixel subsets, compared to the maximal delta among all 45 possible deltas of the same subset, and revealed a negative correlation coefficient of -0.62. Figure 4.17 illustrates these findings. Nevertheless, it is important to mention that the SW had never decreased the evaluation score of the time-averaged pixel subsets—there were no negative deltas.

In a different type of experiment, a wide variety of sliding windows was applied on the Gabors ROI_1 subset (as opposed to applying them on randomly generated pixel subsets). While the evaluation of this subset, along with the evaluation of its SW variations, was different than the techniques introduced in this work (e.g. different usage of InfoGain, application of a feature discretization technique, etc.), the comparison still showed that the maximal gained evaluation score of ROI_1 of $\sim 57\%$ could not be surpassed by any of the tested SW variations. When a computationally intensive, RFE-like procedure (resembling the one demonstrated in Subsection 4.4.5) was applied on every SW variation, after much effort and weeks of waiting, a combination of features was found that had produced an evaluation score of 60.8%. While this may show that an apparent benefit can be gained from applying SW, finding this appropriate feature combination using these methods is impractical, for the reasons detailed in Subsection 4.4.5. Moreover, even if there exists a

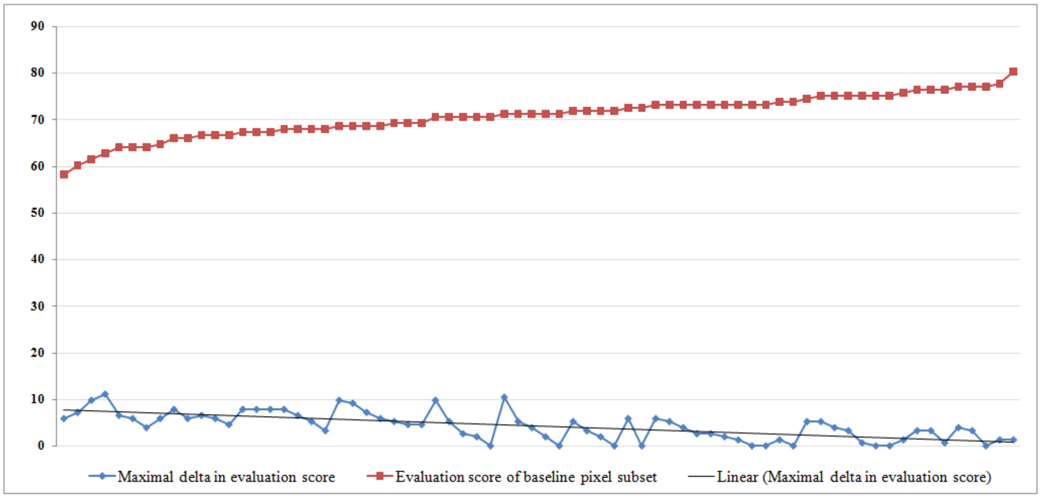


Figure 4.17: Comparison of the evaluation scores of baseline pixel subsets to the maximal improvement gained (maximal delta) in the evaluation scores of these subsets, after applying the 45 sliding window variations. The chart is ordered by ascending evalution scores of the baseline pixel subsets. A linear trendline is displayed for the maximal delta series.

practical way of extracting the appropriate feature combination, the conclusion from the previous type of experiment discussed above shows that the gain in the evaluation score is apparent only for the more inferior pixel subsets, such as ROI_1.

Chapter 5

Application to the hurricanes domain

In Chapter 4, we have demonstrated how our techniques were evaluated in the VSDI domain by applying them to VSDI datasets. We have shown the effectiveness of our methods by validating the classification accuracies of the generated classification models, and argued about their scalability. However, while our methodology was developed based on the VSDI data, it is intended to be effective in general spatiotemporal domains with resembling characteristics. In this chapter we present the first results of applying our methods in a completely different area—the hurricanes. By analyzing periodical satellite images of the Atlantic region, along the time course, we classify the hurricane severity.

5.1 Hurricanes data

In this section, we describe the nature of the data originated from the hurricanes domain. First, we provide a brief background about hurricanes, relevant to the type of the presented classification. Later, we discuss the source data used for dataset generation. Finally, we list the challenges arising in using this data—compared to the VSDI domain.

Following the data description and based on its main aspects, we detail the design of the suggested dataset generation method.

5.1.1 Atlantic hurricanes

Tropical cyclones [37] are storm systems that originate in tropical areas near the equator, over large bodies of warm water. During their life time, tropical cyclones change intensity (strength) and location. *Hurricane* is a term used in most of the Western Hemisphere, specifically in the Atlantic and in the Northeast Pacific regions, for a tropical cyclone that has exceeded certain intensity. Their path of motion, caused by the streams in the Earth's atmosphere and by the Earth's rotation, is called a *track*. They produce extremely powerful winds, heavy rain and flood, and when near coastal regions, they are able to cause severe damage. Tropical cyclones in the Atlantic and in the Northeast Pacific are classified into three main groups of ascending severity: tropical depressions, tropical storms and hurricanes. Within the hurricanes group, the Saffir-Simpson Hurricane Scale [27] divides the hurricanes into five categories by the intensities of their sustained winds. Hurricanes are formed during the *hurricane season*. This season mostly occurs from June to November in the Northern Atlantic Ocean.

In conclusion, a hurricane in the Atlantic region is formed over the Atlantic ocean during the hurricane season. It begins as a low severity tropical cyclone, then moves along its track for an unexpected number of days, while changing its strength (respectively, its severity) during its life course, and finally dissipates. The severities of the hurricanes are thus, from low to high: tropical depression (TD), tropical storm (TS), and categories 1 to 5 according to the Saffir–Simpson Hurricane Scale.

5.1.2 Source images data

The source images data we use in our study are taken from the Global ISCCP B1 Browse System (GIBBS) repository[1] [13], a comprehensive weather satellites data resource. The relevant Atlantic area satellite imagery is the "full disk" images (i.e. the actual shots of the full earth) taken by the consecutively launched Geostationary Operational Environmental Satellite (GOES) satellites, the GOES-8 and GOES-12, operating in two sequential periods of time. The infrared images produced by these two satellites are taken from the same angle and centered over the Atlantic region of the globe. All images are aligned with the

[1] The GIBBS repository is available at http://www.ncdc.noaa.gov/gibbs/.

same longitude, latitude and political lines. GIBBS images from GOES-8 and GOES-12 are available from the beginning of the hurricane season of 1995 to the current day, and taken every 3 hours; the size of each image is 1200 × 1200 pixels. An example of a full sized image is presented in Figure 5.1. Images from various points in time may appear missing, some may occur corrupted, and some of them are incomplete. An example of a partial, incomplete image appears in Figure 5.2.

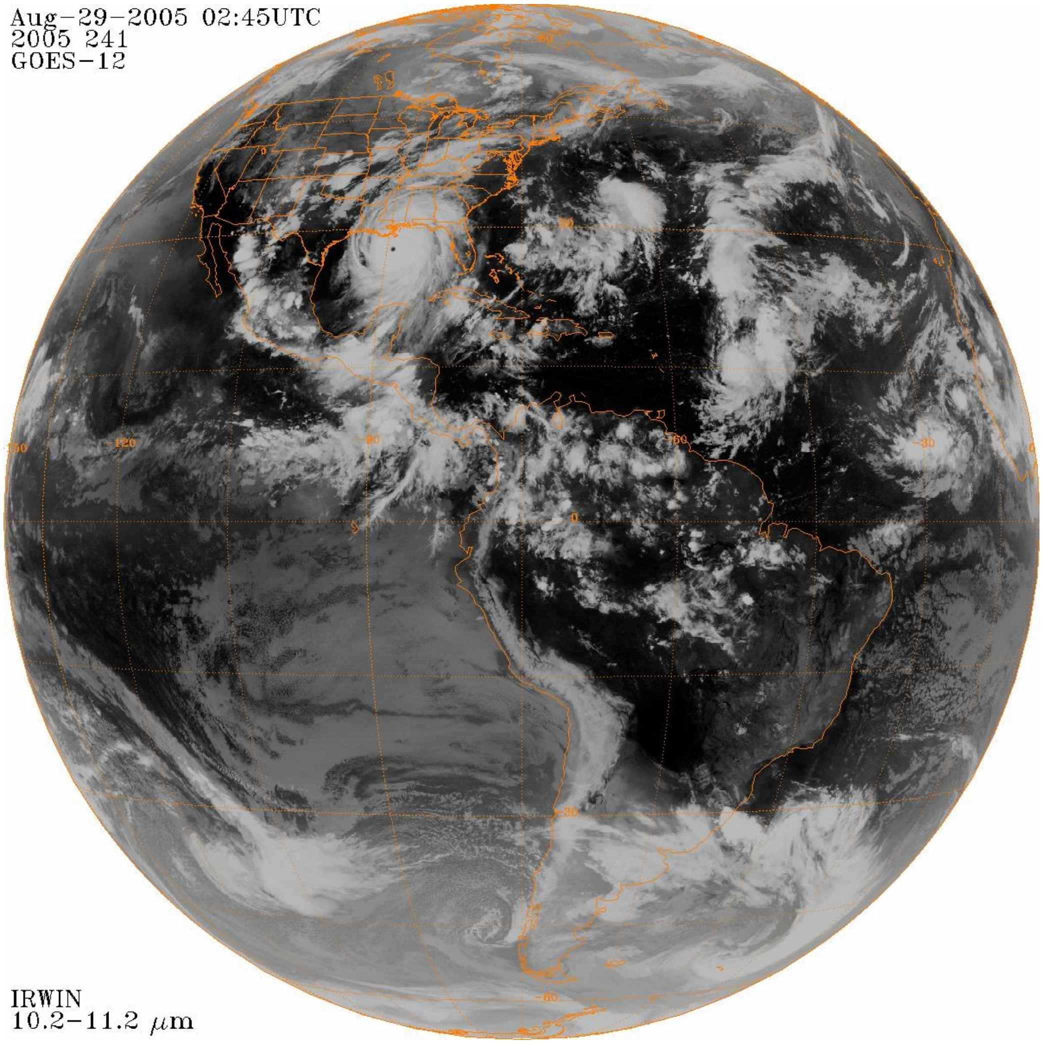


Figure 5.1: GOES-12 infrared (IR) image of the Atlantic region. Famous hurricane Katrina from the 2005 season can be seen in the South area of the United States, affecting Louisiana, Mississippi, South Florida, etc.

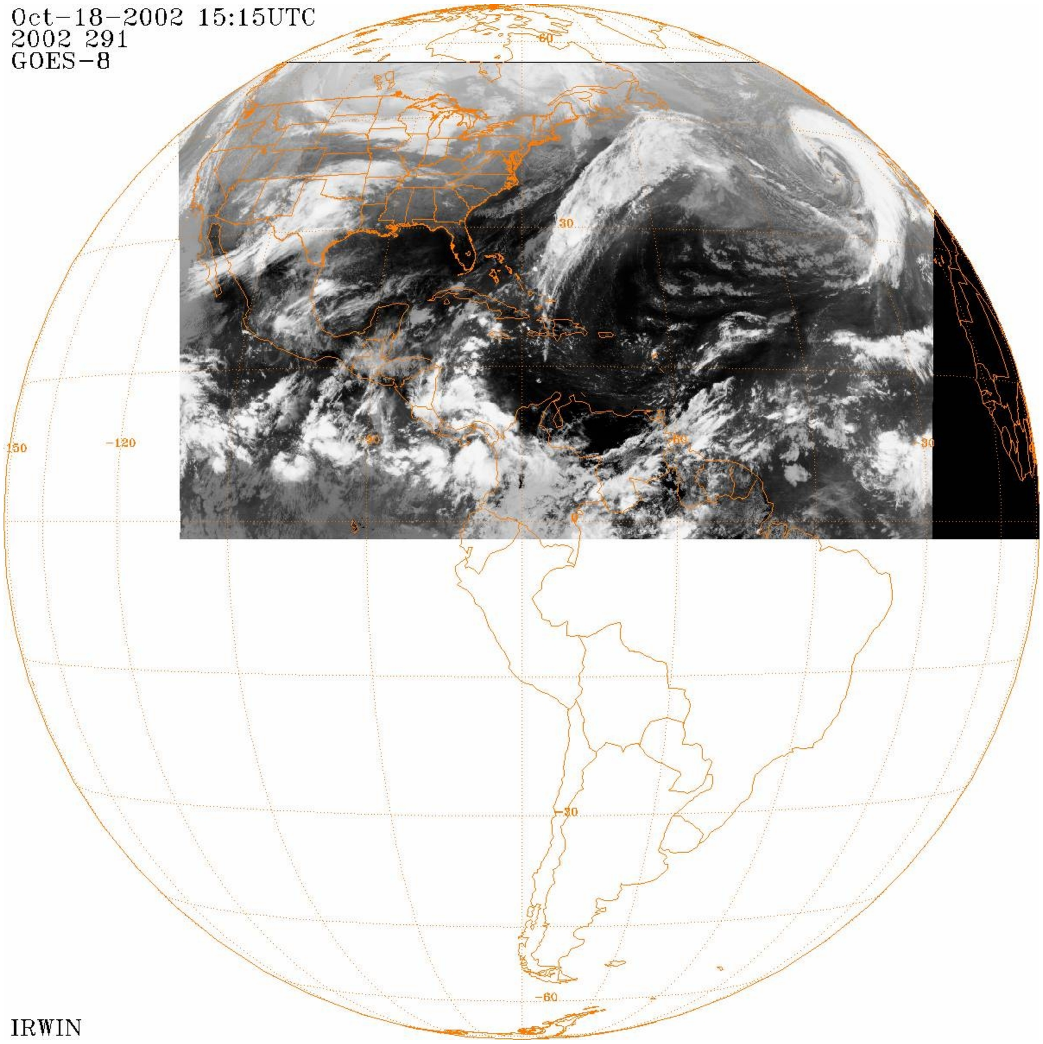


Figure 5.2: Incomplete GOES-8 infrared (IR) image of the Atlantic region.

5.1.3 Challenges and comparison to VSDI

The nature of the data taken from the hurricanes domain is substantially different from the VSDI data in a few major issues, yielding quite a few challenges on the classification task. The main differences and the resulting challenges are detailed below.

1. Technical issues affecting model generation:

   (a) *Size and scale*: The satellite images are of much higher spatial resolution than the VSDI, and if the complete time span of the hurricane is taken into account, they also have a higher temporal resolution. The produced datasets are between 40 to 80 times larger than VSDI[2]. Nevertheless, our modeling methodology was designed to answer this challenge.

   (b) *Missing, incomplete and corrupted images*: Our chosen source of data is very comprehensive. However, it has been manually compiled during a long period of time, based on various sources. Therefore, there are many cases of missing data and corrupted or incomplete images, the reason for which can be simply human mistakes. With VSDI, all the data was complete and precise.

   (c) *Lack of knowledge and expertise in dataset generation*: While with the VSDI data we had the privilege of having a domain expert, who has provided us the professionally prepared datasets, in the hurricanes case the dataset generation method was defined on our own, on a trial and error basis. It is a research task in itself to define what a properly generated dataset is.

2. Hurricane properties, compared to VSDI:

   (a) *Spatial spread pattern*: As opposed to the VSDI data, where the most discriminating pixels selected for the model generation are revealed in specific static areas, the hurricane tracks begin in a wide variety of geographic locations, and move unpredictably all over the area—implying that a selection of one location in the pixel selection phase is not much better than a selection of the other.

   (b) *Nonuniform time span*: While all samples of the VSDI data have a concrete beginning and ending times (as the neuronal population responses appear to take approximately the same amount of time as a reaction to stimuli, at least for the same stimuli type), each hurricane takes a different time—the variability is large. Naturally, hurricanes of lower severities last shorter, and hurricanes of higher severities last longer—but even in the same severity group, the differences between the life spans of individual hurricanes are large.

[2] Please refer to Subsection 5.1.4 for further details.

   (c) *Overlapping*: Each sample of VSDI data reflects the neuronal population responses to a single, isolated type of stimuli. However, different hurricanes of different severities tend to coexist, fully or partially, at the same time, and sometimes even in the same location—they overlap.

   (d) *Severity change during the time course*: The life of the hurricane begins with one of the lowest severities, but during its life span, the severities change (usually rise to their peak and then drop). The time of change and the new severity following the change are unpredicted, and vary highly between every two hurricanes.

3. Class and sample definition task—as opposed to the naturally well defined problem in the VSDI domain (classifying the different visual stimuli in a single imaging experiment), the question of how to build a learning sample and how to define its class in the hurricanes domain is much harder when taking into account the hurricane properties:

   (a) *Sample and its class*: The natural thing to do is to define each hurricane as a single sample, and to specify the sample's class as the hurricane's severity (i.e. its maximal severity during its life time). However, the *overlapping* and the *severity change during the time course* properties change this tendency.

   (b) *Sample start time*: While we strive to define alignment along the time course, as with the case of VSDI (where the neuronal population responses react approximately at the same time after being exposed to the visual stimuli), the *spatial spread pattern* and the *nonuniform time span* properties make the alignment task harder.

   (c) *Sample end time*: Mainly because of the *nonuniform time span* and the *overlapping* properties, it is not trivial to define the end time of a sample. Provided that a sample is defined based on a specific hurricane, it cannot end just where the hurricane ends, as the durations of the hurricanes highly vary—and all samples used by our technique must have a uniform length.

5.1.4 Dataset generation method

Taking into account the points listed in Subsection 5.1.3, we have designed the hurricane dataset generation method in the following way.

The sample and the time course: Each sample was defined to start at the precise start time of an individual hurricane, and to end after exactly 15 days—based on a series of 120 consecutive images (the number of points in time). The grayscale values (0-255) of the pixels in the image were taken as the basic values in a sample. The hurricanes that define each sample were manually picked to have minimal overlapping with other hurricanes within the time range of 15 days since their start time—however, overlapping was still allowed. The motivation behind allowing the overlapping is simply because the overlapping cannot be discarded; on one hand, it frequently exists in practice, and on the other hand, time alignment and normalization is crucial.

The geography of the sample: From each "full disk" image in the series of the sample's images, an area of interest of size 540×300 was identified as the sample's spatial dimension. This area within the image was defined according to the geographic location which most of the hurricane tracks pass through. This area is depicted in Figure 5.3. The remaining part of the image is mostly irrelevant, as the biggest part of it does not contain hurricane tracks most of the time.

The class of the sample: The sample's class was defined as follows—if the severity of the hurricane is between tropical depression and category 2, and the other overlapping hurricanes in its time range do not surpass category 2, the sample was labeled LOW; if the severity of the hurricane is category 3 and above, disregarding the severity of the other overlapping hurricanes, the sample was labeled $HIGH$. The reason for binning each of the individual seven severities into only two classes, the LOW and the $HIGH$, is mainly because of the *overlapping* property, and because of a highly non-uniform distribution of hurricanes from different severities. For instance, as detailed in Figure 5.4, there were 93 tropical storm severity cyclones in the years of 1995 to 2008 (inclusive), but only 8 category 5 hurricanes and only 15 category 2 hurricanes in the same period.

Handling the missing data: Coping with the missing data begins in the pixel level, and goes all the way through all phases of our modeling methodology. Once an area of pixels

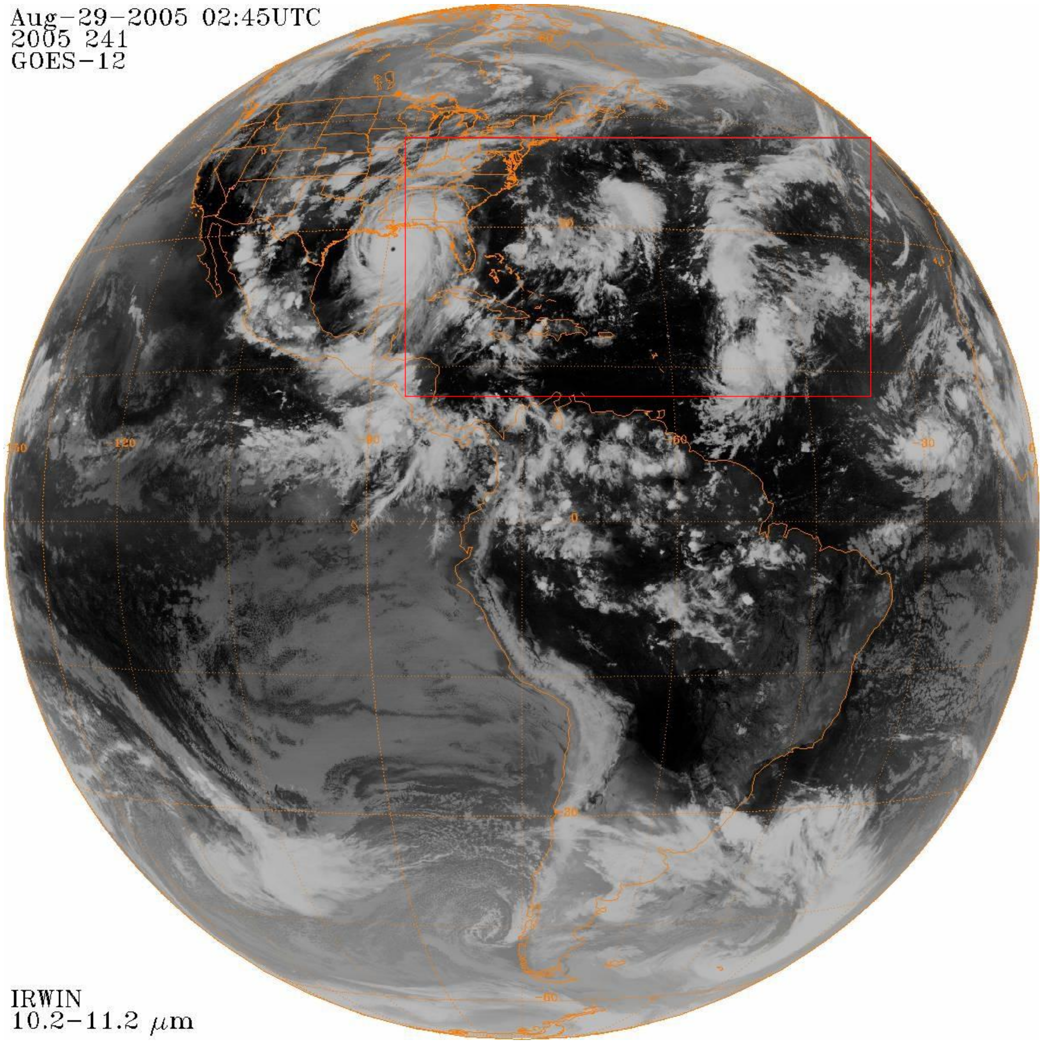
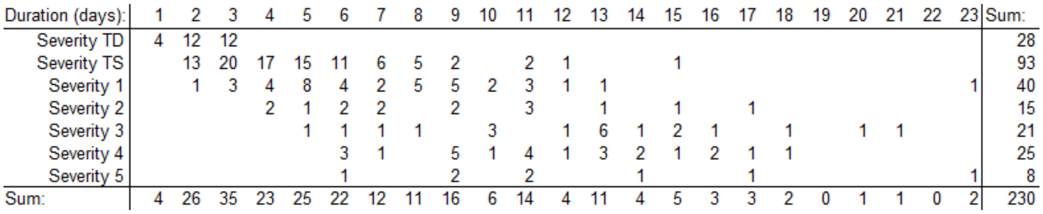


Figure 5.3: GOES-12 infrared (IR) image of the Atlantic region from Figure 5.1, with the area of interest defined by the red square.



Duration (days):	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Sum:
Severity TD	4	12	12																					28
Severity TS		13	20	17	15	11	6	5	2		2	1			1									93
Severity 1		1	3	4	8	4	2	5	5	2	3	1	1										1	40
Severity 2				2	1	2	2		2		3		1		1		1							15
Severity 3					1	1	1	1		3		1	6	1	2	1		1		1	1			21
Severity 4						3	1		5	1	4	1	3	2	1	2	1	1						25
Severity 5						1			2		2			1			1						1	8
Sum:	4	26	35	23	25	22	12	11	16	6	14	4	11	4	5	3	3	2	0	1	1	0	2	230

Figure 5.4: Hurricanes distribution (count) by duration (in days) and severity, for the Atlantic region, years 1995 to 2008 (inclusive).

appears missing or corrupt in an image, the pixels from this area at the corresponding image time are marked as missing. As such, they are not accounted when basic operations on pixels are done during the modeling process—only the existing pixels are taken into account. This includes, among the rest, feature set generation for all classification purposes with feature values marked as missing when using our selected SMO implementation of SVM [38] (which is capable of handling missing data), correlation calculation (by ignoring missing data) during the execution of Algorithm 3.4, the $IPCOSA$, and spatial averaging of pixel groups in $IPCOSA$ (feature values are marked as missing and ignored). In a case where a whole image is missing, we treat each of the pixels in the image as individually missing (an edge case of a missing area of pixels, that covers the whole image).

5.2 Experimental setup

With the guidelines outlined in Subsection 5.1.4, we have constructed a dataset consisting of 55 samples labeled as LOW and of 45 samples labeled as $HIGH$, from the years of 1995 to 2008 (inclusive). That being the case, the baseline accuracy of this dataset (i.e. ZeroR [38]) is 55%.

In resemblance to what we have done in Chapter 4, we have generated classification models using only the $GIRSS$ pixel selection technique—for the lack of an *Oracle*—in the first phase of our methodology, in combination with the two feature extraction techniques in the second phase: $\{GIRSS\} \times \{PIT, IPCOSA\}$, and with the application of the feature selection (Section 3.5) in the third phase. The resulting models were evaluated using a 5-fold cross-validation of the multi-class SMO implementation of SVM with linear kernel [38]. Using 5-fold cross-validation instead of using the standard 10-fold cross-validation is due to the long running times of each experiment. When the time allowed, model's evaluation was performed more than once (each trial yielding a different random 5-fold division), as specified in the results Table 5.1.

5.3 Preliminary results

The experimental results for the hurricanes dataset, presented in Table 5.1, show the accuracies resulted from combining the $GIRSS$ pixel selection method with each of the feature extraction techniques—the PIT and the $IPCOSA$ (refer to the table caption for the legend of the table entries).

Despite the apparently low accuracy of the results, they are statistically significant when compared to the baseline, and appear well above the chance level. When we take into account the principal differences between the VSDI and the hurricanes domains, and review the challenges listed in Subsection 5.1.3, we discover that in practice, even though the two domains share common spatiotemporal characteristics, these key differences between them are fundamental. Nevertheless, these initial, preliminary results, which were achieved in a considerably smaller amount of time than the amount of time invested in VSDI, imply that the potential of the methodology presented in our work is probably large enough to handle classification tasks that originate in extremely varied spatiotemporal domains.

As with the results of the Oriented Gratings dataset from VSDI, presented in Section 4.4, we see that the $IPCOSA$ spatial averaging produces accuracies lower than the ones produced by using the PIT approach. The reason for this behavior in VSDI was related to high spatial frequency of the data, for which averaging over space has caused the loss of signal. In the case of the hurricanes, this hypothesis for a possible explanation has yet been verified.

Table 5.1: The results of applying each of the combinations: $\{GIRSS\} \times \{PIT, IPCOSA\}$ on the hurricanes dataset. The numbers in brackets for u (number of pixels in a random pixel subset) and r (number of random pixel subsets) are their respective values. The results of the form $\mu \pm \sigma\%\,(n)$ have μ representing the average accuracy between the trial runs, σ representing the standard deviation and n representing the number of trial runs. The entry in bold represents the best accuracy obtained.

HURRICANES

BASELINE: 55%

		$GIRSS$	
	$u\,(1620)\,,r\,(150)$	$u\,(2430)\,,r\,(100)$	$u\,(3300)\,,r\,(70)$
PIT	$\mathbf{66.5 \pm 1.5\%\,(2)}$	$60.0 \pm 0.0\%\,(1)$	$63.0 \pm 0.0\%\,(1)$
$IPCOSA$	$63.0 \pm 1.0\%\,(2)$	$56.0 \pm 0.0\%\,(1)$	$60.0 \pm 0.0\%\,(1)$

Chapter 6

Discussion

In this chapter, we first discuss the implications of our work on the neuroimaging field of study. We then draw conclusions about the results of the presented work. Finally, we introduce the directions we foresee and present the ideas we have for the future work based on our research.

6.1 Neuroimaging implications

Some questions arise in light of these results with respect to the neuroimaging perspective and neural decoding in particular. Our results show that machine learning can definitely be applied on fields such as VSDI for decoding and possibly other tasks. Without prior knowledge in neuroimaging, we can successfully classify (to some extent) different neuronal population responses with respect to the provoking stimuli. We can support neuroimaging researchers in revealing the dominant areas in the brain responsible for visual processing. Can our results shed new light on the dynamics of the neuronal populations? We believe it can, for two reasons. First, the support of our domain expert, who believes that these results look interesting and promising, and that a further and deeper study is necessary in order to advance in their interpretation. Second, by analyzing the differences revealed between the expert's ROI pixel sets to the ones selected by our technique. Not only that the pixels selected by a non-expert technique provide at least as good results as the expert's ROIs, but they also provide new findings on their significance.

6.2 Conclusions and future work

We presented a combination of methods that employ machine learning techniques to handle large scale spatiotemporal data. Initially, our techniques were developed based on the field of VSDI, and were successfully evaluated there. We consider this work as pioneering, in terms of combining these two perspectives to produce an interdisciplinary AI research, applied for the first time to the VSDI domain. VSDI technology is novel and revolutional, from the viewpoint of its spatial and temporal resolutions, therefore its potential has yet to be explored to its full extent. With this advanced neuroimaging technology and our proposed tools, we foresee further progress in the development of visual perception decoding algorithms to aid in decoding novel visual stimulus, such as movies or real-time streaming visual data. We plan to compare different decoding mechanisms over different cortical areas and behavioral conditions. Having advanced decoding abilities will allow greater understanding of the visual pathway functionality and will allow progress in the revelation of cognitive brain states—helping promote computer-aided control of artificial prostheses by patients' brain activity, diagnosing cognitive activity in paralyzed patients, or directly contributing to development of vision prostheses.

Furthermore, we have examined the applicability of our methodology to the hurricanes domain. Although at first we have expected our techniques to yield analogous behavior in both the VSDI and the hurricanes fields—which apparently share common spatiotemporal characteristics—we have later revealed a great deal of principal differences between the two domains. Nevertheless, in applying our methods to the hurricanes domain, despite being of a very different nature, we have managed to achieve initial results that show a promising direction—even in a field as different as this one. The kind of application we have explored here demonstrates the extent of the potential of our methodology.

In addition to our intent of doing advanced VSDI decoding—as discussed at the beginning of this section, our immediate focus is to achieve more accurate classification results with the hurricanes. As we have just begun initial experimentation in the hurricanes domain, we believe that producing substantially higher accuracies is possible by examining a broader variety of dataset generation techniques. On top of all that, we also plan to apply our methodology in other spatiotemporal domains with resembling characteristics. We are excited to see how well it will behave in new domains.

We ought to mention that our methods do not treat the time dimension as a dimensionality threat, thus not taking an effort to effectively reduce it. However, we did preliminary attempts to apply various sliding window techniques for temporal reduction, but without any apparent advantage (as expected with potential data loss; details provided in Section 4.6). Expecting future data to have a much higher temporal resolution obligates temporal reduction. For this purpose, we believe that using the methods for dimensionality reduction of time series discussed in Subsection 2.2.2—such as Discrete Fourier Transform (DFT) or Discrete Wavelet Transform (DWT), as reported in [20, 34]—will help us find a lower dimensionality time-course representation that preserves the original information.

Appendix A

Machine learning glossary

Presented in this appendix are brief descriptions of the concepts and the machine learning tools discussed and utilized in our work.

A.1 Supervised learning and classification

Classification as a *supervised learning* task is a machine learning technique [18] for learning a function from a classified (labeled) training data, which is capable of predicting the class of an unclassified sample (whose origin is outside of the training data set).

Given a training sample set S and a class label set C, where each training sample $s \in S$ is labeled with a class label $c \in C$, and an infinitely large universal set U of all possible unlabeled samples $u \in U$, the standard classification problem is to build a model that approximates classification functions of the form $f : U \longrightarrow C$, which map unclassified samples from U to the set of class labels C.

A.2 Support Vector Machines

The Support Vector Machine (SVM) [38] is a systematic and reproducible supervised learning approach, properly motivated by statistical learning theory. The training of SVM involves optimization of a convex cost function. SVMs are well-known for using the idea of kernel substitution (*kernel methods*). The SVM and the kernel methodology are well-suited for machine learning and data mining tasks. With SVMs, overfitting is unlikely to occur, and no false local minima complicates the learning process. The classification task using the SVM approach produces elegant mathematical models, both *geometrically intuitive* and *theoretically well-founded.* SVMs are based on two alternative principles: the *convex hull* approach, or finding the *maximum margin hyperplane.*

The *convex hull* of a set of points is the smallest convex set containing the points. For two linearly separable classes, we can examine the convex hull of each class' training data, and then find the closest points in the two convex hulls. Then we can construct the plane that bisects these two points, resulting in finding a classifier. The closest points in the two convex hulls can be found by solving a quadratic problem (QP). The example for this approach is presented in Figure A.1.

As an alternative, we can find the plane furthest from both sets of points, also known as the *maximum margin hyperplane*—the one that gives the greatest separation between the classes. To do this, we need to maximize the distance (the margin) between the support planes for each class. The support planes are "pushed" apart until they "bump" into a small number of data points (the support vectors) from each class. Maximizing the distance (the margin) between the supporting planes can also be found by solving a quadratic problem. Figure A.2 presents an example for this principle.

The solutions found by both presented methods are identical. In the maximum margin method, the solution depends on the support vectors. In the convex hull method, same data points determine the closest points in the two convex hulls. This is due to the mathematical programming concept of duality. Either of the quadratic problems (primal or dual) of the two methods can give the same solution. These optimization problems (also known as constrained quadratic optimization problems) are relatively straightforward from a mathematical programming perspective (belong to a well-studied class of convex quadratic programs). Many effective robust algorithms for such tasks exist.

However, for two datasets that are not linearly separable, the convex hull strategy will fail. The solution for this is to use reduced convex hulls to restrict the influence of each point, in such way that the convex hulls would not intersect. An appropriate modification

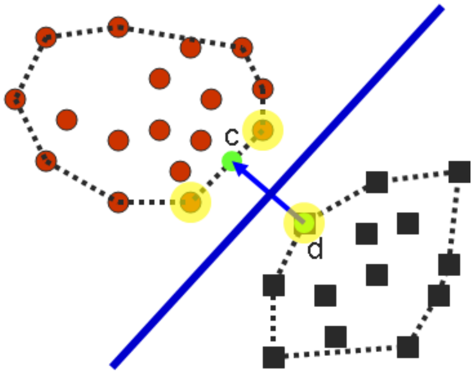


Figure A.1: The convex hull. The red circles and the black squares represent two different classes. The dotted lines are convex hulls of each class. *c* and *d* are the closest points in the convex hulls. The three boldly circled points are the support vectors. The bisecting plane is the classifier.

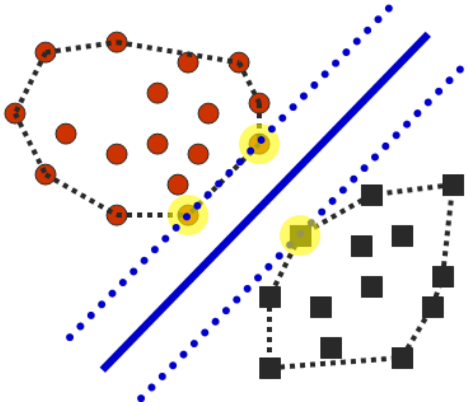


Figure A.2: The maximum margin hyperplane. The red circles and the black squares represent two different classes. The blue dotted lines are the support planes for each class. The solid blue line is the plane farthest from both sets—the classifier. The three boldly circled points are the support vectors.

for the QP problem needs to be applied. The supporting plane method will also fail in this case. To cope with the infeasibility of the QP task here, the constraints that insure that each point is on the appropriate side of its supporting plane need to be relaxed. Since any point falling on the wrong side of the supporting plane is considered to be an error, we want to simultaneously maximize the margin and minimize the error, thus introducing minor changes to the supporting plane QP problem.

Nevertheless, in some cases even that is not enough. If the linear discriminants are inappropriate for the data set—resulting in high training set errors—SVM methods would not perform well. In this case, the SVM approach needs to be generalized to construct highly nonlinear classification functions. To convert a linear classification algorithm into a nonlinear classification algorithm, the principle is to add additional attributes to the data, that are nonlinear functions of the original data. Then, existing linear classification algorithms can be applied to the expanded dataset in feature space, producing nonlinear functions in the original input space.

While doing this will cause the exponential explosion of the dimensionality of the feature space, causing the computation to become impractical, using the *kernel methods* to substitute the original dot products of the QP problems will solve the problem. We can get different highly nonlinear classifiers by employing various kernels. Robust, efficient algorithms, that have no problem with local minima, can be used for training highly nonlinear classification functions. Using kernel substitution allows turning a linear algorithm, only capable of handling separable data, into a general nonlinear algorithm.

The WEKA machine learning framework [38] we employ in the implementation of the techniques presented in our work, makes use of the SMO variation of the SVM. SMO implements the Sequential Minimal Optimization algorithm for training a support vector classifier, using polynomial or Gaussian kernels. Missing values are replaced globally, nominal attributes are transformed into binary ones, and attributes can be normalized. Pairwise classification is used for multiclass problems, where the predicted probabilities are coupled pairwise. Please refer to [38] for a more detailed description of SVM, and SMO in particular.

A.3 Information Gain (InfoGain) based filtering

The Information Gain (InfoGain) measure [38] is borrowed from the Information Theory field. It is based on measuring the relative reduction in entropy. The classic *entropy* measure is defined as: $E\left(S\right) = \sum_{c \in C} -\frac{|S_c|}{|S|} log_2 \frac{|S_c|}{|S|}$, where S is the entire training dataset, C is the class label set and S_c is a subset of S in which the training samples are labeled with class $c \in C$.

The InfoGain rank (IG) is defined as: $IG\left(S, f\right) = E(S) - \sum_{v \in V(f)} \frac{|S_v|}{|S|} E\left(S_v\right)$, and shows how much information is gained by splitting the dataset relative to the feature f. Here, $V\left(f\right)$ is the set of unique values of the feature f and S_v is the subset of S in which the value of the feature f is v.

The feature evaluation based on the Information Gain measure evaluates features by measuring their information gain with respect to the class. The filtering approach implemented in WEKA [38] discretizes numeric attributes first using the Minimum Description Length (MDL)-based discretization method. The techniques we have presented in Chapter

3 use this InfoGain based filtering method.

A.4 Stratified cross-validation

The *stratified cross-validation* method [38] is used for predicting the error rate of a learning technique—and specifically, for evaluating a classification model—given a fixed, limited amount of data.

Reserving a certain amount of data for testing, and using the remainder for training, is known as the *holdout* method. The sample data used for training or testing might not be representative. Hence, each class in the full dataset should be represented in approximately the right proportion in the training and testing sets. The random sampling needs to be done in such a way that it is guaranteed that each class is properly represented in both training and testing sets—procedure known as *stratification*.

Repeating the whole process (training and testing) several times with different random samples is a general way to mitigate any biases caused by any particular samples of data chosen for holdouts. In each iteration, a certain proportion of the data is randomly selected for training, and the remainder is used for testing. In a k-fold *cross-validation* procedure, the data is randomly partitioned into k parts, or *folds*, in each of which the class is represented in approximately the same proportions as in the full dataset. Each part is held out in turn, and the clasification model is built using the remaining $k - 1$ parts. The error rate of the held out data part is calculated. The learning procedure is executed a total of k times on the different training sets. Finally, the k error estimates are averaged to yield an overall error estimate.

Setting the k to a value of 10, i.e. using a 10-fold cross-validation, is in fact the standard way of predicting the error rate of a learning technique [38], and specifically of a classification model. By adding stratification, the stratified 10-fold cross-validation has in practice become the standard evaluation technique. We have employed this technique in Chapters 3 and 4 to evaluate our classification models.

A.5 Pearson's product moment coefficient

The correlation coefficient [7] between vectors a and b of length n is defined as follows: $\rho = \frac{\sum_{i=1}^{n}(a_i - \mu_a)(b_i - \mu_b)}{(n-1)\sigma_a \sigma_b}$, where μ_a and μ_b are the respective means of vectors a and b, and σ_a and σ_b are their respective standard deviations. The coefficient ρ represents how strongly the variables imply each other. If $\rho > 0$, a and b are positively correlated—the values of a increase as the values of b increase. Symmetrically, if $\rho < 0$, a and b are negatively correlated. The coefficient ρ ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), whereas 0 indicates no correlation (in which case a and b are independent).

A.6 ZeroR as baseline accuracy

The ZeroR (or 0-R) described in [38] is a basic classification rule: it predicts the test data's majority class (if nominal) or average value (if numeric). We have employed ZeroR in our experimental evaluation in Chapter 4, for setting the baseline classification accuracy to which our results are compared.

Bibliography

[1] Geoffrey M. Boynton. Imaging orientation selectivity: decoding conscious perception in V1. *Nature Neuroscience*, 8(5):541–542, 2005.

[2] Yuzhi Chen, Wilson S. Geisler, and Eyal Seidemann. Optimal decoding of correlated neural population responses in the primate visual cortex. *Nature Neuroscience*, 9(11):1412–1420, October 2006.

[3] C. Davatzikos, K. Ruparel, Y. Fan, D. G. Shen, M. Acharyya, J. W. Loughead, R. C. Gur, and D. D. Langleben. Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. *Neuroimage*, 28(3):663–668, November 2005.

[4] Mehrdad Fatourechi, Gary Birch, and Rabab Ward. Application of a hybrid wavelet feature selection method in the design of a self-paced brain interface system. *Journal of NeuroEngineering and Rehabilitation*, 4(1):11, 2007.

[5] Mathieu Fauvel, Jocelyn Chanussot, and Jon Atli Benediktsson. A joint spatial and spectral SVM's classification of panchromatic images. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS'07)*, pages 1497–1500, July 2007.

[6] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[7] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*, chapter 3, page 121. Morgan Kaufmann, 2000.

[8] John-Dylan Haynes and Geraint Rees. Predicting the orientation of invisible stimuli from activity in human primary visual cortex. *Nature Neuroscience*, 8(5):686–691, April 2005.

[9] John-Dylan Haynes and Geraint Rees. Predicting the stream of consciousness from activity in human visual cortex. *Current Biology*, 15(14):1301–1307, July 2005.

[10] John-Dylan Haynes and Geraint Rees. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7(7):523–534, July 2006.

[11] Yukiyasu Kamitani and Frank Tong. Decoding the visual and subjective contents of the human brain. *Nature Neuroscience*, 8(5):679–685, April 2005.

[12] Yukiyasu Kamitani and Frank Tong. Decoding seen and attended motion directions from activity in the human visual cortex. *Current Biology*, 16(11):1096–1102, June 2006.

[13] Kenneth R. Knapp. Scientific data stewardship of International Satellite Cloud Climatology Project B1 global geostationary observations. *Journal of Applied Remote Sensing*, 2(1):023548, November 2008.

[14] S. LaConte, S.C. Strother, V. Cherkassky, and X. Hu. Predicting motor tasks in fMRI data with support vector machines. In *Proceedings of the 11th Scientific Meeting of the International Society for Magnetic Resonance in Medicine*, 2003.

[15] Carmen Lai, Marcel J. Reinders, and Lodewyk Wessels. Random subspace method for multivariate feature selection. *Pattern Recognition Letters*, 27(10):1067–1076, July 2006.

[16] Hyekyung Lee and Seungjin Choi. PCA+HMM+SVM for EEG pattern classification. In *Proceedings of the 7th International Symposium on Signal Processing and Its Applications*, volume 1, pages 541–544 vol.1, July 2003.

[17] Tom Mitchell, Rebecca Hutchinson, Marcel Adam Just, Radu S. Niculescu, Francisco Pereira, and Xuerui Wang. Classifying instantaneous cognitive states from fMRI data. In *Proceedings of the 2003 Americal Medical Informatics Association Annual Symposium*, page 469, 2003.

[18] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[19] Tom M. Mitchell, Rebecca Hutchinson, Radu Stefan Niculescu, Francisco Pereira, Xuerui Wang, Marcel Just, and Sharlene Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57(1-2):145–175, 2004.

[20] Fabian Mörchen. Time series feature extraction for data mining using DWT and DFT. Technical Report 33, Department of Mathematics and Computer Science, University of Marburg, Germany, 2003.

[21] Janaina Mourao-Miranda, Karl J. Friston, and Michael Brammer. Dynamic discrimination analysis: A spatial-temporal SVM. *NeuroImage*, 36(1):88–99, May 2007.

[22] Mark Palatucci. Temporal feature selection for fMRI analysis. Working paper (unpublished), February 2007.

[23] Mark Palatucci and Tom M. Mitchell. Classification in very high dimensional problems with handfuls of examples. In *Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, volume 4702 of *Lecture Notes in Computer Science*, pages 212–223. Springer-Verlag, 2007.

[24] Indrayana Rustandi. Classifying multiple-subject fMRI data using the hierarchical Gaussian Naïve Bayes classifier. In *13th Conference on Human Brain Mapping*, June 2007.

[25] Pradeep Shenoy, Kai Miller, Nathan Evans, Jeffrey Ojemann, and Rajesh Rao. Robust classification of Electrocorticographic Signals for BCI. 2008.

[26] Svetlana V. Shinkareva, Hernando C. Ombao, Bradley P. Sutton, Aprajita Mohanty, and Gregory A. Miller. Classification of functional brain images with a spatio-temporal dissimilarity map. *NeuroImage*, 33(1):63–71, October 2006.

[27] Robert H. Simpson and Herbert Riehl. *The Hurricane and Its Impact*. Louisiana State University Press, Baton Rouge, 1981.

[28] Sameer Singh. EEG data classification with localized structural information. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00)*, pages 2271–2274, 2000.

[29] Vishwajeet Singh, Krishna P. Miyapuram, and Raju S. Bapi. Detection of cognitive states from fMRI data using machine learning techniques. In *Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 587–592, 2007.

[30] H. Slovin, A. Arieli, R. Hildesheim, and A. Grinvald. Long-term voltage-sensitive dye imaging reveals cortical dynamics in behaving monkeys. *Journal of Neurophysiology*, 88(6):3421–3438, December 2002.

[31] Le Song, Alex Smola, Arthur Gretton, Karsten M. Borgwardt, and Justin Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*, pages 823–830. ACM, 2007.

[32] Jonathan Stoeckel and Glenn Fung. SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 410–417, 2005.

[33] Dima Stopel, Zvi Boger, Robert Moskovitch, Yuval Shahar, and Yuval Elovici. Improving worm detection with artificial neural networks through feature selection and temporal analysis techniques. *International Journal of Computer Science and Engineering*, 15:202–208, 2006.

[34] Michail Vlachos, Jessica Lin, Eamonn Keogh, and Dimitrios Gunopulos. A wavelet-based anytime algorithm for K-Means clustering of time series. In *Workshop on Clustering High Dimensionality Data and Its Applications at the 3rd SIAM International Conference on Data Mining*, pages 23–30, 2003.

[35] Xuerui Wang, Rebecca Hutchinson, and Tom M. Mitchell. Training fMRI classifiers to discriminate cognitive states across multiple subjects. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[36] Yongmei Michelle Wang, Robert T. Schultz, R. Todd Constable, and Lawrence H. Staib. Nonlinear estimation and modeling of fMRI data using spatio-temporal support vector regression. In *Information Processing in Medical Imaging (IPMI'03)*, pages 647–659. Springer Berlin / Heidelberg, 2003.

[37] Wikipedia. Tropical cyclone — Wikipedia, The Free Encyclopedia, 2009. [Online; accessed 14-October-2009].

[38] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, second edition, 2005.

[39] Li-Qun Xu and Yongmin Li. Video classification using spatial-temporal features and PCA. In *Proceedings of the 2003 International Conference on Multimedia and Expo (ICME'03)*, volume 3, pages III–485–8 vol.3, 2003.

[40] Kiyoung Yang and Cyrus Shahabi. A PCA-based kernel for kernel PCA on multivariate time series. In *Proceedings of ICDM 2005 Workshop on Temporal Data Mining: Algorithms, Theory and Applications held in conjunction with The Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 149–156, 2005.

[41] Kiyoung Yang, Hyunjin Yoon, and Cyrus Shahabi. A supervised feature subset selection technique for multivariate time series. In *Proceedings of International Workshop on Feature Selection for Data Mining: Interfacing Machine Learning with Statistics (FSDM) in conjunction with 2005 SIAM International Conference on Data Mining (SDM'05)*, pages 92–101, 2005.

[42] Hyunjin Yoon and Cyrus Shahabi. Feature subset selection on multivariate time series with extremely large spatial features. In *Proceedings of the 6th IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, pages 337–342. IEEE Computer Society, 2006.

[43] Lei Zhang, Dimitris Samaras, Dardo Tomasi, Nelly Alia-Klein, Lisa Cottone, Andreana Leskovjan, Nora D. Volkow, and Rita Goldstein. Exploiting temporal information in functional magnetic resonance imaging brain data. In *Proceedings of the 8th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'05)*, volume 3749 of *Lecture Notes in Computer Science*, pages 679–687. Springer, 2005.

[44] Qibin Zhao and Liqing Zhang. Temporal and spatial features of single-trial EEG for brain-computer interface. *Computational Intelligence and Neuroscience*, 2007(1):4–4, 2007.