Scalable Classification in Large Scale Spatiotemporal Domains Applied to Voltage-Sensitive Dye Imaging

Igor Vainer[1], Sarit Kraus[1], Gal A. Kaminka[1] and Hamutal Slovin[2,3]
[1]Department of Computer Science
[2]The Leslie and Susan Gonda Multidisciplinary Brain Research Center
[3]The Mina and Everard Goodman Faculty of Life Sciences
Bar-Ilan University
Ramat Gan, 52900, Israel
iv@013.net, {sarit,galk}@cs.biu.ac.il, slovinh@mail.biu.ac.il

***Abstract*—We present an approach for learning models that obtain accurate classification of large scale data objects, collected in spatiotemporal domains. The model generation is structured in three phases: pixel selection (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. Model generation based on the combinations of techniques from each phase is explored. The introduced methodology is applied on datasets from the Voltage-Sensitive Dye Imaging (VSDI) domain, where the generated classification models successfully decode neuronal population responses in the visual cortex of behaving animals. VSDI currently is 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.**

***Index Terms*—classification; spatiotemporal; application; brain imaging; neural decoding; visual cortex;**

I. Introduction

Recently, there is much interest in applying machine learning in domains with large scale spatiotemporal characteristics. Examples range from discriminating cognitive brain states using functional Magnetic Resonance Imaging (fMRI) [1], [2], [3], [4], [5], to developing techniques for classification of brain signals in Brain Computer Interfaces (BCI) [6], [7], [8], [9], performing automated video classification [10] and more.

However, many existing techniques prone insufficient when the data is temporal (spanned over a time course) and spatially exhaustive (consists of a large number of locations in space). Classification often becomes computationally intensive and unfeasible. Raw data collected along the time course in a high-resolutional space results in hundreds of thousands of data-points, for which classical, straightforward machine learning approaches become practically ineffective. Studies presented in [3], [11] face these obstacles, discuss why the existing methods fail and present possible solutions (which either produced medium accuracy results, or were applied to moderately scaled datasets).

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 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 [12]. 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.

We apply 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)[13]. 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 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 [14], no machine learning methods were used—a proprietary 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.

II. Related work

Much research for decoding of cognitive brain states, employing machine learning methods, has been done in fMRI. 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. Such studies include picking the top n most active voxels based on t-test [15] or on average between the voxels [16]; picking the top n most discriminating voxels, based on training a classifier per each voxel [1]; or, picking the n most active voxels per Region Of Interest (ROI) [1]. While they manage to produce moderate to high accuracy results, they rely on relatively small resolutions of data (where training a classifier per voxel is admissable), or on expert knowledge (defining an ROI). The methods we present in our work are domain independent (require no prior knowledge), aimed at very high resolutional data, and exploit both temporal and spatial dimensions.

As for fMRI exploiting the temporal dimension, [2] employs the following heuristic: features are defined as voxel-timepoint pairs, ranked by how well they individually classify a training set, and the top 100 features for the final classifier are chosen. While individual training of classifiers for all time-space combinations is totally unfeasible in large scale domains, we do adopt the time-space combination approach in our work. Additional work that has inspired us is [3], in which one of the introduced techniques for feature selection is 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 are selected, and closeness of each pair of voxels' time series is measured. Despite the high reported success rates, the techniques in [3] are subject to be computationally expensive in large scale domains.

A different spatiotemporal domain which is fundamentally based on techniques for classification of brain-emitted signals is BCI. Here, brain-controlled computer systems are developed in order to operate the machine (e.g. prostheses, communication) by brain activity (e.g. imagining a hand movement will cause a prosthetic computer-controlled arm to move). For example, the method presented in [8] 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 Recursive Feature Elimination (RFE, as proposed in gene selection problem [17]) is used for feature subset selection of time-series datasets. Applying RFE in a similar manner on our type of data is computationally expensive—however, we do adopt the approach of correlation between spatial elements in our feature extraction.

A last example from spatiotemporal domains is automated video genre classification [10]. In this case, the problem is investigated by first computing a spatiotemporal combined audio-visual "super" feature vector (of very high dimensionality). Then, the feature vector is further processed using Principal Component Analysis (PCA) to reduce the spatiotemporal redundancy while exploiting the correlations between feature elements. However, the PCA-based techniques in multivariate time-series datasets are known to be problematic in regard to scalability, which is more than evident in our domain.

III. Spatiotemporal classification modeling

In this section, we present the *three phase methodology* for building scalable models for spatiotemporal data classification. To describe our methodology, we first need to formalize the problem. A spatiotemporal domain contains n pixels that constitute the global pixel set $P = \{p_1, p_2, \dots, p_n\}$. Every pixel p_i, $i \in \{1, \dots, n\}$ 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, $i \in \{1, \dots, n\}$, $t \in \{1, \dots, m\}$ 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.

That being the case, the finite training samples set of size k in the spatiotemporal domain is defined as: $S = \{s_1, s_2, \dots, s_k\}$, where a single sample s_l, $l \in \{1, \dots, k\}$ is a set of vectors: $s_l = \{\overline{p_1}, \dots, \overline{p_n}\}$, where a vector $\overline{p_i} = \langle v_1^i, \dots, v_m^i \rangle$, $v_t^i \in \mathbb{R}$, $t \in \{1, \dots, m\}$ 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 $u = \{\overline{p_1}, \dots, \overline{p_n}\}$, $u \in U$, 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.

In the next subsections we describe each of the phases in detail. Subsection III-A presents a technique for selecting the pixels that have the most discriminative characteristics among the global pixel set P. Next, in III-B, we introduce two alternative techniques for extracting the features from the pixels selected by the first phase. The third phase described in III-C shows 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.

A. 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 extremelly 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 [12] for selecting by iterative refinement, the set of pixel subsets from which we can eventually derive the sought-after pixel set.

In Algorithm 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]. Each generated pixel subset's classification capabilities are roughly evaluated (Algorithm 2): pixel values in time are defined as features (step 2), as was done in [2]. Then an Information Gain (InfoGain) based feature selection [18] 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 insipired by [3], 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, [18]) to obtain an evaluation score (accuracy of the evaluated set). The scores are then ordered in a descending order, and the greedy phase begins.

During the greedy phase, we maintan 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 (Algorithm 3), and evaluated (using Algorithm 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's discarded. 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. The extraction of the pixel set from Γ (Algorithm 3) 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 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: $\Gamma \longleftarrow \left\{\tilde{P}^{d^{I_Z\left[1\right]}}\right\}$
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) $\Gamma^{'} \longleftarrow \Gamma \cup \left\{\tilde{P}^{d^{I_Z\left[j\right]}}\right\}$
    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 $P^* = \left\{p_1^*, p_2^*, ..., p_u^*\right\}$ (top $u$ spatial subspace representatives).
```

Algorithm 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 $z$.
```

B. Feature extraction

Methods described here are applied on the pixel selection results of the first phase (Subsection III-A). 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.

1) Features as pixels in time: 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 Subsection III-A for ranking pixel subsets and feature sets. While for simpler classification tasks this is satisfactory—fast, simple and effective (Section IV), a method described next is suggested for more complex tasks.

[1] From our experience, having $u \cdot r \approx 1.5n$, u and r being of about the same order of magnitude (see Table I), 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).

Algorithm 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 $F$ over the sample set $S$:
       $F = \left\{p_j^t \mid t \in \{1,\ldots,m\}, \forall p_j \in P_i\right\}$
    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:
        i) $\rho\left[j\right] = \frac{\rho[j]\cdot\iota[j]+Z[i]\cdot\frac{\sum_{t=1}^{m} IG\left(p_j^t\right)}{m}}{\iota[j]+1}$
        ii) $\iota\left[j\right] = \iota\left[j\right] + 1$
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 $p_{I_\rho[l]} \in P$, $l \in \{1,\ldots,u\}$.
```

Algorithm 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) $R \longleftarrow R \setminus \{p\}$
    b) $G \longleftarrow \{p\}$
    c) $\Delta \longleftarrow \Delta \cup \{G\}$
7) Initialize feature-set $F^*$ over the sample set $S$, $F^* \longleftarrow \emptyset$.
8) for $\forall G \in \Delta$ do:
    a) for $t = 1$ to $m$ do:
        i) Define $f^t$—the average of values of all pixels in $G$ at time $t$: $f^t = \frac{\sum_{i=1}^{|G|} v_t^i}{|G|}$, $s.t. \overline{p_i} = \left\langle v_1^i, \ldots, v_m^i \right\rangle$, $v_t^i \in \mathbb{R}$, $t \in \{1, \ldots, m\}$, $\forall p_i \in G$
        ii) $F^* \longleftarrow F^* \cup \left\{f^t\right\}$
Output: Feature set $F^*$ over the sample set $S$.
```

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, and how to choose "similar" pixels in space, designated for averaging. The way we measure similarity is by employing Pearson's product moment coefficient [19] between pairs of pixels. We then 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.

In Algorithm 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 (III-A)—we refer to this set as "seeds". First, we calculate a correlation coefficient matrix C and a distances matrix D between all pixel pairs (step 3). 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'

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

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

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. The resulting features represent the average-in-time of similar pixels, as opposed to the pixel-in-time approach presented in III-B1. For seeds pixel set of size u, there will be at most $u \cdot m$ features (number of formed groups will not exceed the number of seeds, as each group must contain at least one seed).

C. *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 [18] is applied on the given feature set F, producing scores: $IG(f), \ \forall f \in F$. Then, only the features with positive InfoGain scores: $IG(f) > 0$ are selected. The motivation: the features produced in III-B are based on pixel selection from III-A, 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.

IV. Experimental evaluation

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 Section II. 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 additionaly compare our results to those achieved by an $Oracle$—a domain expert—faced with the same tasks, and validate their credibility.

A. *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 [13]. 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. 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.

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 (Fig. 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 consitutes 34.13% of the population (setting the baseline accuracy, i.e. ZeroR [18]).

2) Dataset 2: Gabors (complex): The monkey was presented with five different Gabor based orientations in space and a blank control image (Fig. 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 consitutes 18.95% of the population (ZeroR baseline accuracy).

3) Dataset 3: Contours (hard): The monkey was presented with four different Gabor-based Contours in space and a blank control image (Fig. 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 consitutes 23.39% of the population (ZeroR baseline accuracy).

B. *Experimental methodology*

As a part of the evaluation methodology for the pixel selection technique presented in III-A, we define the $Oracle$: a

[4] As our experimental evaluation shows (Section IV), in most cases the weakest formed groups are based on correlation coefficient of at least 0.4.

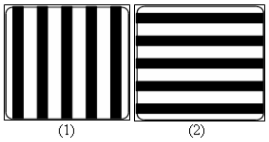

Figure 1. Stimuli for the Oriented Gratings dataset: (1) Drifted square gratings at vertical orientations; (2) Drifted square gratings at horizontal orientations; (3) Blank control image (not presented).

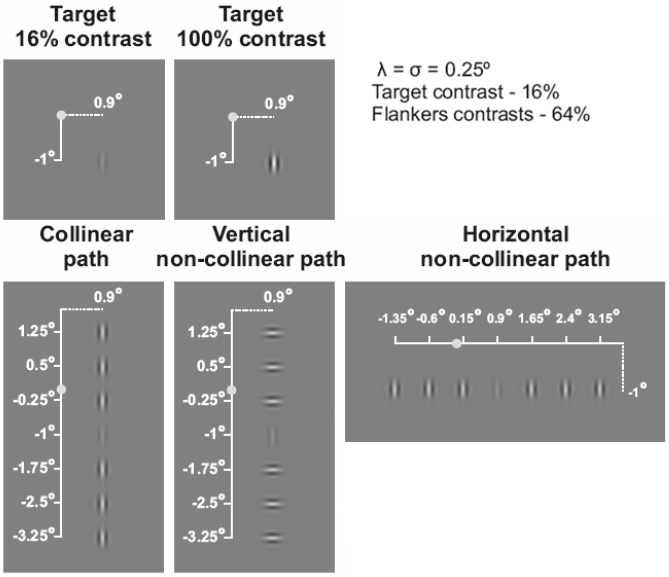

Figure 2. Stimuli for the Gabor dataset (the numbers and the degrees on the white axes are not part of the stimuli; blank control image not presented). The circle represents the fixation point location.

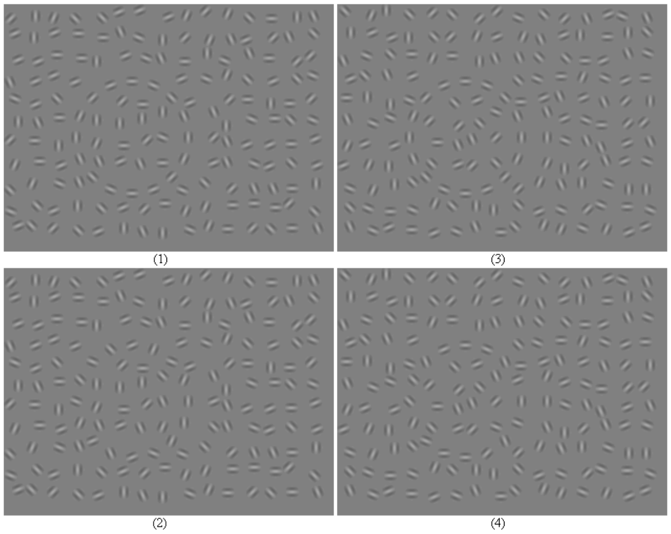

Figure 3. Example of stimuli for the Contours dataset: (1) Circle 1; (2) Masked circle 1; (3) Circle 2; (4) Masked circle 2; (5) Blank control image (not presented).

pixel selection method, a best-effort attempt by a human expert who was asked 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 human expert, or the $Oracle$, manually picked 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 success rates of using the pixel set selected by $GIRSS$, our pixel selection technique.

In the experimental setup, the domain expert was requested to provide an ROI of pixels for each dataset. In the case of Gabors, we were given an improved ROI, based on the results of using the original ROI[5]. In the case of Contours, three different ROIs of pixels were given in advance, each for individual evaluation by our techniques. We built models using both pixel selection techniques as the first phase, in combination with the two feature extraction methods as the second: $\{Oracle, GIRSS\} \times \{PixelInTime, IPCOSA\}$, with application of the feature selection phase (III-C). The resulting models were evaluated using a 10-fold cross-validation of the multi-class SMO implementation of SVM with linear kernel [18]. 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 I.

C. Results

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've shown that for two types of data (Oriented Gratings, Contours), our pixel selection technique is capable of producing pixel sets having as good dicriminative 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. In this case, we see major difference when our selected pixel sets are compared to the ROI pixel sets (please refer to Fig. 4 and 5 for example comparison). While both of the ROI sets 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.

[5] See results of ROI_2 pixel set of Gabors dataset in Table I, compared to the results of ROI_1.

Table I
THE RESULTS OF APPLYING EACH OF THE COMBINATIONS: $\{Oracle, GIRSS\} \times \{PixelInTime\,(^{1}), IPCOSA\,(^{2})\}$ 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.

ORIENTED GRATINGS

BASELINE: 34.13%

	$Oracle$	$GIRSS$
	$ROI_1\,(154)$	$u\,(154)\,, r\,(20)$
1	$95.4 \pm 0.4\%\,(10)$	$94.9 \pm 1.3\%\,(10)$
2	$79.3 \pm 3.2\%\,(10)$	$88.5 \pm 4.0\%\,(10)$

GABORS

BASELINE: 18.95%

		$Oracle$	
	$ROI_1\,(104)$	$ROI_2\,(218)$	
1	$55.0 \pm 1.5\%\,(10)$	$68.8 \pm 1.1\%\,(10)$	
2	$57.2 \pm 3.3\%\,(10)$	$71.0 \pm 2.4\%\,(10)$	
		$GIRSS$	
	$u\,(100)\,, r\,(150)$	$u\,(100)\,, r\,(125)$	$u\,(100)\,, r\,(100)$
1	$79.1 \pm 1.7\%\,(10)$	$78.5 \pm 1.8\%\,(6)$	$78.6 \pm 1.5\%\,(7)$
2	$81.8 \pm 1.4\%\,(10)$	$81.4 \pm 2.7\%\,(6)$	$80.6 \pm 1.8\%\,(7)$

CONTOURS

BASELINE: 23.39%

		$Oracle$	
	$ROI_1\,(151)$	$ROI_2\,(227)$	$ROI_3\,(155)$
1	$44.9 \pm 2.3\%\,(10)$	$50.6 \pm 2.4\%\,(10)$	$73.3 \pm 1.6\%\,(10)$
2	$40.2 \pm 3.1\%\,(10)$	$47.6 \pm 3.0\%\,(10)$	$65.7 \pm 1.8\%\,(10)$
		$GIRSS$	
	$u\,(151)\,, r\,(100)$	$u\,(500)\,, r\,(100)$	
1	$71.9 \pm 2.8\%\,(10)$	$69.6 \pm 2.8\%\,(10)$	
2	$72.4 \pm 2.7\%\,(10)$	$73.1 \pm 2.1\%\,(10)$	

The high accuracy of the Oriented Gratings dataset is somehow expected due the apparent differences between its visual stimuli, but it's not for granted considering the baseline of 34.13%. 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 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.

As for the feasibility of construction and evaluation of our models—all early attempts to handle the data before basing our pixel selection on random subspace [12], such as employing techniques that base their feature extraction, selection and classification on the full spatial range (resembling methods proposed in [15], [16], [1], [2]), ended with impractical running times (waiting for weeks with no end in prospect) and memory requirements. However, $GIRSS$ and $IPCOSA$ 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. Truly, our proposed models are not only feasible, but practical.

D. 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 observed the responses of the same neuronal populations, solely before the onset of the stimuli, we wouldn't expect to have the ability to discriminate between them—simply because of the fact that the behaviour 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 III, 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 $Oracle$ wasn't 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 redundant.

That being the case, we would expect the classification results to be close to baseline accuracies of each of the datasets. Indeed, we can safely say that the results of this stage were as expected—roughly the same as the chance level. In our validation procedure, the time course was reduced to the first 10 points in time before the visual stimulus onset for the Oriented Gratings and the Gabors datasets, and to the first 3 points in time for the Contours. For the Oriented Gratings, $GIRSS$ with $u\,(154)\,, r\,(20)$, applied with $PixelInTime$, produced the average accuracy of 31.6%; the application of $IPCOSA$ instead of $PixelInTime$ yielded the average accuracy of 34%. In the Gabors case, $GIRSS$ with $u\,(100)\,, r\,(150)$ and $PixelInTime$, produced the average accuracy of 17.5%, while using the $IPCOSA$ generated roughly the same outcome. Applying $GIRSS$ with $u\,(151)\,, r\,(100)$ on the Contours produced the accuracy of 21% when used along with the $PixelInTime$, and the accuracy of 22.7% when used with the $IPCOSA$. Using $GIRSS$ with $u\,(500)\,, r\,(100)$

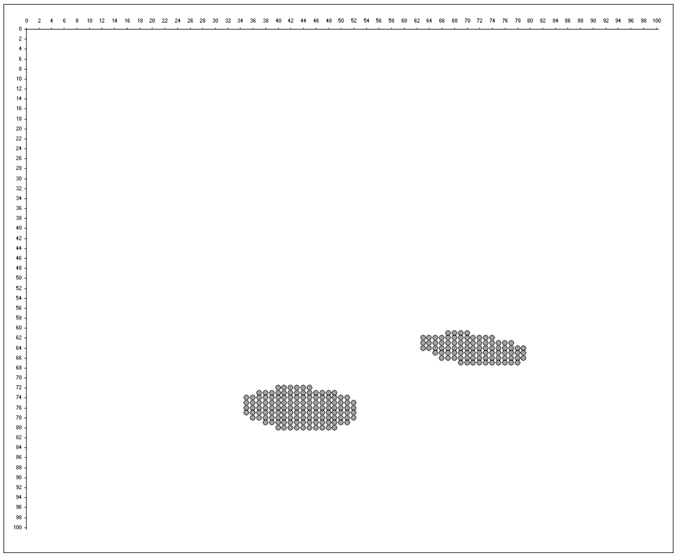

Figure 4. Gabors dataset, $ROI_2(218)$—the best performing $Oracle$'s ROI pixel set. The imaged area of pixels is depicted on the grid (all pixels are in V1, the primary visual cortex).

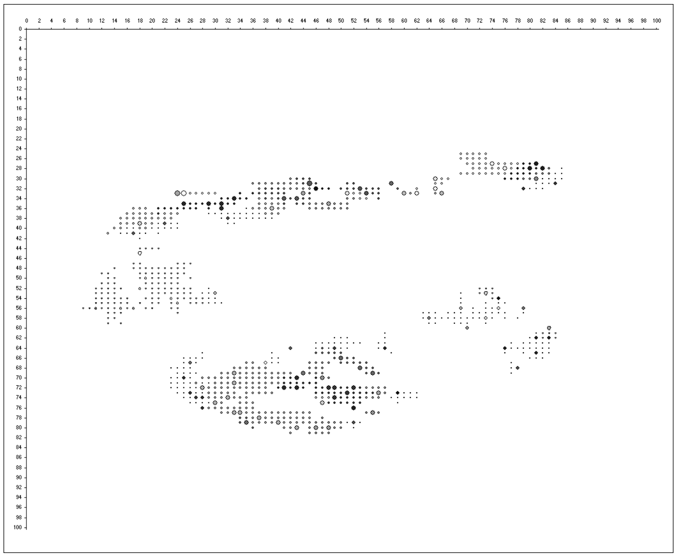

Figure 5. Gabors dataset, sample fold result—the imaged area of pixels depicted on the grid. Applied $GIRSS$ with $u(100)$ and $r(125)$ to produce "seeds" pixel set (large circles). Applied $IPCOSA$ to improve the accuracy of $PixelInTime$ from 87.67% to 100%. Neighborhoods of pixels for averaging are formed around the seeds (small circles, having the seeds' shades). The different sizes of pixels between the neighborhoods express the inter-correlation strength within each neighborhood, compared to the other ones.

for the Contours dataset, generated the results of 19% and 22.4% when using the $PixelInTime$ and the $IPCOSA$, respectively.

E. 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.

V. Conclusions and future work

In this work, we presented a combination of methods that employ machine learning techniques to handle vast spatiotemporal VSDI data. In addition to the results and implications discussed in Section IV, 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. With 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. Thanks to the fact that our techniques are domain independent, we also plan to apply them in other spatiotemporal domains with resembling characteristics.

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). Expecting future data to have a much higher temporal resolution obligates temporal reduction. For this purpose, we believe that using Discrete Fourrier Transform (DFT) or Discrete Wavelet Transform (DWT) for dimensionality reduction of time-series, as reported in [20], [21], will help us find a lower dimensionality time-course representation that preserves the original information—describing the original shape of the time-series data as closely as possible.

Acknowledgment

This research was supported in part by ISF grants #859/05 (to H. S.) and #1357/07. We thank Elhanan Meirovithz for his help in VSDI data aquisition, and Ofer Garnett for his helpful advice. As always, thanks to K. Ushi and V. Ricki for their support.

References

[1] T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman, "Learning to decode cognitive states from brain images," *Machine Learning*, vol. 57, no. 1-2, pp. 145–175, 2004.

[2] M. Palatucci, "Temporal feature selection for fMRI analysis," February 2007, working paper.

[3] L. Zhang, D. Samaras, D. Tomasi, N. Alia-Klein, L. Cottone, A. Leskovjan, N. D. Volkow, and R. Goldstein, "Exploiting temporal information in functional magnetic resonance imaging brain data," in *MICCAI*, ser. Lecture Notes in Computer Science, vol. 3749. Springer, 2005, pp. 679–687.

[4] J. Mourao-Miranda, K. J. Friston, and M. Brammer, "Dynamic discrimination analysis: A spatial-temporal SVM," *NeuroImage*, vol. 36, no. 1, pp. 88–99, May 2007.

[5] H. Yoon and C. Shahabi, "Feature subset selection on multivariate time series with extremely large spatial features," in *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*. IEEE Computer Society, 2006, pp. 337–342.

[6] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 823–830.

[7] S. Singh, "EEG data classification with localized structural information," in *ICPR*, 2000, pp. 2271–2274.

[8] K. Yang, H. Yoon, and C. Shahabi, "A supervised feature subset selection technique for multivariate time series," in *Proceedings of the Workshop on Feature Selection for Data Mining: Interfacing Machine Learning with Statistics, 92-101*, 2005.

[9] Q. Zhao and L. Zhang, "Temporal and spatial features of single-trial EEG for brain-computer interface," *Intell. Neuroscience*, vol. 2007, no. 1, pp. 4–4, 2007.

[10] L.-Q. Xu and Y. Li, "Video classification using spatial-temporal features and PCA," in *ICME '03: Proceedings of the International Conference on Multimedia and Expo*, vol. 3, 2003, pp. III–485–8 vol.3.

[11] M. Fatourechi, G. Birch, and R. Ward, "Application of a hybrid wavelet feature selection method in the design of a self-paced brain interface system," *Journal of NeuroEngineering and Rehabilitation*, vol. 4, no. 1, p. 11, 2007.

[12] C. Lai, M. J. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1067–1076, July 2006.

[13] H. Slovin, A. Arieli, R. Hildesheim, and A. Grinvald, "Long-term voltage-sensitive dye imaging reveals cortical dynamics in behaving monkeys," *Journal of Neurophysiology*, vol. 88, no. 6, pp. 3421–3438, December 2002.

[14] Y. Chen, W. S. Geisler, and E. Seidemann, "Optimal decoding of correlated neural population responses in the primate visual cortex," *Nature Neuroscience*, vol. 9, no. 11, pp. 1412–1420, October 2006.

[15] T. Mitchell, R. Hutchinson, M. A. Just, R. S. Niculescu, F. Pereira, and X. Wang, "Classifying instantaneous cognitive states from fMRI data," in *Proceedings of the 2003 Americal Medical Informatics Association Annual Symposium. Washington D.C*, 2003, p. 469.

[16] X. Wang, R. Hutchinson, and T. M. Mitchell, "Training fMRI classifiers to discriminate cognitive states across multiple subjects," in *NIPS*, 2003.

[17] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[18] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[19] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000, ch. 3, p. 121.

[20] M. Vlachos, J. Lin, E. Keogh, and D. Gunopulos, "A wavelet-based anytime algorithm for K-Means clustering of time series," in *Proceedings Workshop on Clustering High Dimensionality Data and Its Applications*, 2003, pp. 23–30.

[21] F. Mörchen, "Time series feature extraction for data mining using DWT and DFT," Department of Mathematics and Computer Science, University of Marburg, Germany, Tech. Rep. 33, 2003.