Frequency–Based Slow Feature Analysis

Alexandros Doumanoglou[1,*], Nicholas Vretos[1], Petros Daras[1]

Abstract

Slow Feature Analysis (SFA) is an unsupervised learning algorithm which extracts slowly varying features from a temporal vectorial signal. In SFA, feature slowness is measured by the average value of its squared time-derivative. In this paper, we introduce Frequency-Based Slow Feature Analysis (FSFA) and prove that it is a generalization of SFA in the frequency domain. In FSFA, the low pass filtered versions of the extracted slow features have maximum energy, making slowness a filter dependent measurement. Experimental results show that the extracted features depend on the selected filter kernel and are different than the signals extracted using SFA. However, it is proven that there is one filter kernel that makes FSFA equivalent to SFA. Furthermore, experiments on UCF-101 video action recognition dataset, showcase that the features extracted by FSFA, with proper filter kernels, result in improved classification performance when compared to the features extracted by standard SFA. Finally, an experiment on UCF-101, with an indicative, simple and shallow neural network, being composed of FSFA and SFA nodes, demonstrates that the previously mentioned network, can transform the features extracted by a known Convolutional Neural Network to a new feature space, where classification performance through Support Vector Machine can be improved.

Keywords: Slow Feature Analysis, Slow Signals, Filtering, Fourier Spectrum, Action Recognition

*Corresponding author

Email addresses: aldoum@iti.gr (Alexandros Doumanoglou), vretos@iti.gr (Nicholas Vretos), daras@iti.gr (Petros Daras)

[1]The authors are with the Information Technologies Institute, Centre For Research and Technology - HELLAS. 6th Km Charilaou-Thermi Road, Thessaloniki, Greece

1. Introduction

By exploiting the slowness learning principle [1], [2], which is argued to be one of the potential principles that may drive the visual recognition ability of the human brain [3], the algorithm of Slow Feature Analysis (SFA) [4] is an unsupervised methodology for extracting slowly varying features from a rapidly changing, raw input, vector signal. More formally, SFA measures a signal's slowness by its average temporal squared derivative and extracts features that minimize this metric. Since the average temporal squared derivative of any signal is directly dependent on its scale, the formulated optimization problem avoids unfair comparisons between signals by imposing additional constraints that make the extracted features to have zero mean and unit variance. Optimizing under these criteria, SFA provides a closed form solution to the constraint optimization problem in the case where the extracted signals are considered to be a linear combination of the input signals' components in an expansion space.

In this work, namely Frequency-Based Slow Feature Analysis (FSFA), we are motivated to investigate the slowness criterion in the frequency domain by intuitively arguing that slow signals ought to have their energy concentrated at the lower ends of the frequency spectrum. We formulate an objective function that measures signal slowness by performing a weighted sum of the signal's energy spectrum, giving major importance to low frequencies and lower importance to higher ones. From this perspective, the newly introduced slowness metric is a parametric function of the preset weights. The proposed slowness criterion is directly affected by the signal's scale and offset. Thus, for an evenhanded approach to slow feature extraction, adopting the constraints of zero mean and unit variance was considered reasonable and practical. Eventually, by maximizing the aforementioned objective, in the search space of the linear combinations of the expanded input (as it is for standard SFA), we would expect slow feature extraction, under a new definition of slowness.

In the definition of FSFA's slowness, to enable slow feature extraction, the

preset weights ought to follow a monotonically decreasing formula with respect to frequency. From this point of view, FSFA's slowness metric can be seen as a low pass filter operation on the input signal and the previously mentioned preset weights constitute a filter kernel. The objective function optimized by FSFA, requires that the extracted features result into a signal of maximum energy when they are passed through this low-pass filter, essentially meaning that the features' energy is concentrated at low frequencies. In the general case, the usage of different filters can result into extracting features with different characteristics. At this point, while the new slowness metric seemed generalized and parameterizable, whether it has any relation to the slowness criterion of standard SFA was still an open question. In this paper, we prove that there is a specific filter kernel that turns FSFA equivalent to SFA, justifying that FSFA is a parameterized generalization of SFA in the frequency domain.

In summary, the novelties of this paper are the following:

- We generalize the notion of feature slowness by making it filter dependent.
- We formulate the generalized slowness optimization problem and provide its closed form solution.
- By solving the generalized optimization problem, we enable parameterized slow feature extraction through a preset filter.
- We prove the existence of a filter that makes FSFA equivalent to SFA and conclude that FSFA is a parameterized generalization of SFA.
- We discuss on the design of the filter kernel giving hints on its baseband bandwidth and study the implications that some common filters impose on the extracted output signals.
- We present experimental results on artificial and real-world data, justifying that the optimal FSFA features depend on the chosen filter kernel.
- We showcase that by choosing proper filters, classification performance can be improved compared to standard SFA in a video action recognition

context.

- We demonstrate that a shallow, simple, neural network that consists of four FSFA nodes utilizing different filters and a single SFA node, can be used as a post-processing step, in order to transform the features extracted by a known Convolutional Neural Network (CNN) to a new feature space, where classification performance, through a standard Support Vector Machine (SVM), can be improved.

The rest of the paper is structured as follows: in Section 2, related work to SFA is reviewed. In Sections 3, 4 and 5 we elaborate on the proposed method, its relation to SFA and filter design. In Section 6, experimental results are presented in both artificial and real datasets. Finally, Section 7 concludes the paper.

2. Related Work

SFA was firstly introduced as a novel method for learning invariant or slowly varying features from a vectorial input signal by Wiskott *et al* [4]. In their original paper, the authors introduced the slowness principle and proposed a mathematical formulation of signal's slowness, based on the average squared first order time derivative of the signal. The solution is searched in a finite function expansion space. An extensive study on the performance of SFA in different expansion spaces can be found in [5].

Since then, SFA has been applied in a number of applications. In [6] and [7], SFA was used to extract driving forces of non-stationary time series. In [1], SFA was employed for invariant object recognition while in [8], SFA was applied hierarchically for age and gender estimation from synthetic face images. Later, Zhang *et al* [9] proposed a variation of SFA, for supervised learning, called Discriminative SFA (D-SFA) applied to Human Action Recognition. A similar workflow like in [9] was also followed in [10] where SFA was used to detect violence in videos. In [11], SFA was used for hyperspectral anomaly

change detection while in [12], SFA was employed for gesture recognition from acceleration signals. In the work of Sun *et al* [13], SFA was combined with Deep Learning for Action Recognition. Furthermore, in [14], SFA was applied in Action Recognition from Motion Capture (MoCap) skeleton data. Finally, in [15] SFA is applied for change detection in multispectral imagery. A short survey on the capabilities of SFA in the application level can be found in [16].

Apart from the application level, in a theoretical basis the relation of SFA to other techniques has also attracted researchers' attention. In [17], it was shown that in the case of one time delay, the linear SFA is equivalent to second order Independent Component Analysis (ICA) and in [18] SFA was linked to Laplacian Eigenmaps. Moreover, Turner *et al* [19], showed an equivalence between SFA and Maximum Likelihood learning in a linear Gaussian state-space model, with an independent Markovian prior. SFA was also used in [20] for Nonlinear Blind Source Separation. Finally, in [21], it was shown that SFA can acquire the classification capability of Fisher's Linear Discriminant (FLD) for supervised learning when adjacent samples are likely to be from the same class.

Apart from its original form, variations of SFA have also appeared in the literature. In [22], Kompella *et al* presented the first online version of SFA based on incremental Principal Component Analysis (PCA) and Minor Component Analysis. Later, Liwicki *et al* [23] proposed an incremental SFA for change detection for online temporal video segmentation. Moreover, in [24], the same authors presented an online kernel variation of SFA with application in the same area. Additionally, in [25], Berkes introduced SFA to pattern recognition where the output of SFA provided a feature space suitable for classification. In the work of [26], Escalante *et al*, introduced an extension of SFA for supervised dimensionality reduction called graph-based SFA (GSFA). The newly introduced optimization problem generalized the notion of slowness from sequences of samples to training graphs. Furthermore, in [27], Escalante *et al* improved GSFA to preserve information in a hierarchical manner. Other variations of SFA include [28] and [29] with applications in audio/video and multitemporal remote sensing imagery, respectively. Finally, in [30], a novel deterministic SFA algorithm

able to identify linear projections that extract the common slowest varying features of two or more sequences, was presented. Additionally, an expectation maximization algorithm was proposed performing inference in a probabilistic formulation of SFA. Those algorithms were used for facial behavior analysis demonstrating their effectiveness.

In the important work of [3], an equivalence of the SFA optimization problem in the frequency domain is presented for the first time. It is shown that the SFA's minimization problem in the time domain, is equivalent to an energy maximization problem of low pass filtered versions of the input in the frequency domain. This work closely resembles the approach taken in the present paper. However, while in [3] the equivalence of SFA's optimization problem given in the frequency domain actually holds for continuous signals, in the present paper, we derive the appropriate equivalence in the discrete domain. On the side, in the present paper we provide a generalized closed form solution to the optimization problem in the frequency domain for any arbitrary filter, something that was out of the scope of [3].

Finally, in the literature, one may find other works on modeling the general temporal coherence principle. In [31], Hadsel et al. introduced Dimensionality Reduction by Learning an Invariant Mapping (DrLIM) for learning a globally coherent function that maps the data evenly to an output manifold. In [32], a deep learning architecture is illustrated that takes advantage of the temporal coherence principle to introduce a supervisory signal over unlabeled video recordings, improving the performance of a supervised task of interest. Futher, Wang et al. [33], used video input to a siamese-triplet Convolutional Neural Network (CNN) to enforce visual representations of patches to be the similar in deep feature space. Last, Jayaraman et al. [34] introduced "steady-feature" analysis that imposes a prior that higher order derivatives in the learned feature space must be small. While these works are relevant to the current paper, they do not overlap with the work presented here, as they are mainly different models for the temporal coherence principle. Contrariwise, this work generalizes SFA's slowness principle in the frequency domain.

3. Proposed Method

We begin by intuitively relating the slowness of a signal to its Fourier Spectrum. We argue, that among signals of the same total energy, the slowest ones should result into maximum remaining energy after they are passed through a low-pass filter.

Let $\mathbf{x}[t] = \big[x_1[t], x_2[t], ..., x_I[t]\big], t \in \{0, 1, ..., N-1\}, x_i[t] \in \mathbb{R}\ \forall\ i \in \{1, 2, ..., I\}$, $I \in \mathbb{N}^*$ and $N \in \mathbb{N}^*$, denote a multi-dimensional discrete signal. Similar to SFA, the FSFA's objective, is to find a function $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), ..., g_J(\mathbf{x})], g_j[\mathbf{x}] \in \mathbb{R}\ \forall\ j \in \{1, 2, ..., J\}$ generating the J-dimensional signal $\mathbf{y}[t] = \big[y_1[t], ..., y_J[t]\big]$ with $y_j[t] = g_j(\mathbf{x}[t])$, such as, the mono-dimensional signals $y_j[t]$ vary slowly across t. For FSFA, we propose, that the following objective is maximized:

$$\Delta_j = \sum_{k=0}^{N-1} |\mathbf{Y}_j[k]|^2 \mathbf{W}[k] \tag{1}$$

under the constraints:

$$\langle y_j \rangle = 0 \iff \mathbf{Y}_j[0] = 0 \quad \text{(zero mean)} \tag{2}$$

$$\langle y_j^2 \rangle = 1 \iff \langle \left|\mathbf{Y}_j\right|^2 \rangle = 1 \quad \text{(unit variance)} \tag{3}$$

$$\langle y_{j'} y_j \rangle = 0 \iff \langle \mathbf{Y}_{j'} * \mathbf{Y}_j \rangle = 0 \quad \text{(decorrelation)} \tag{4}$$

In (1), $\mathbf{Y}_j[k]$ refers to the Discrete Fourier Transform (DFT) of $y_j[t]$ at frequency k and $\mathbf{W}[k] \in \mathbb{R}$ denotes a filtering kernel (also referred as the "filter") that weighs the Fourier spectrum with different weights $0 \leq \mathbf{W}[k] \leq 1$. By choosing different filter kernels $\mathbf{W}[k]$ one gets different signals $\mathbf{y}[t]$. In order to perform fair comparisons between candidate features, constrains (2)-(4) are imposed to the problem.

As in [4], solving (1) when $g_j(\mathbf{x})$ belongs to an infinite function space, is a difficult problem of variational calculus. However, in case we constrain the functions g_j to be a linear combination of nonlinear functions belonging to a finite set, as it will be shown in this section and similar to what happens in standard SFA, the solution simplifies to an eigenvalue problem [35].

Let $\mathbf{h} = [h_1, ..., h_K], K \in \mathbb{N}^*, h_i \in \mathbb{R}, i \in \{1, ..., K\}$ a vector-valued function composed of a finite set of scalar valued functions h_i. Applying the vector function $\mathbf{h}$ to the input signal $\mathbf{x}[t]$ yields a nonlinearly expanded signal $\mathbf{z}[t] = \mathbf{h}(\mathbf{x}[t])$ where typically $K \gg I$. After this expansion, the problem can be treated as a linear one in the expanded space. Thus, $y_j[t] = g_j(\mathbf{x}[t]) = \mathbf{h}(\mathbf{x}[t])\mathbf{w}_j = \mathbf{z}[t]\mathbf{w}_j, \mathbf{w}_j \in \mathbb{R}^{K\times 1}$, with $\mathbf{z}[t]$ to be known, given the input $\mathbf{x}[t]$, and $\mathbf{w}_j$ to be the unknown. Let $\mathbf{z}[t] = \big[z_1[t], ..., z_K[t]\big]$. It is important to note that in order for a solution to fulfill the constraints (2) - (4), it is required that ${\mathbf{w}_j}^T\mathbf{w}_j = 1$ and the signal $\mathbf{z}[t]$ is centered and sphered, as in [4]. Moreover, to fulfill (4), the weight vectors $\mathbf{w}_j$ that correspond to different $y_j[t]$ need to form an orthonormal set.

Let $\mathbf{z}$ denote a matrix whose rows are equal to $\mathbf{z}[t]$ for different $t \in \{0, ..., N-1\}$, $\mathbf{z} \in \mathbb{R}^{N\times K}$ and $\mathbf{y}_j \in \mathbb{R}^{N\times 1}$ denote the column vector of $y_j[t]$ for all t. Then $\mathbf{y}_j = \mathbf{z}\mathbf{w}_j$. Let also $\mathbf{D} \in \mathbb{C}^{N\times N}$ the unitary DFT matrix. Then:

$$\mathbf{Y}_j = \mathbf{D}\mathbf{z}\mathbf{w}_j \in \mathbb{C}^{N\times 1} \tag{5}$$

The objective function (1) can now be rewritten as:

$$\Delta_j = \sum_k (\mathbf{Y}_j \circ \mathbf{Y}_j^* \circ \mathbf{W})_k \tag{6}$$

where $(\cdot)^*$ denotes the complex conjugate, $\circ$ the Hadamard product and summation takes place along all the elements of the resulting vectors. For three column vectors $\mathbf{A} \in \mathbb{C}^{N\times 1}$, $\mathbf{B} \in \mathbb{C}^{N\times 1}$ and $\mathbf{C} \in \mathbb{C}^{N\times 1}$, the following property holds for the summation of their Hadamard product:

$$\sum_i (\mathbf{A} \circ \mathbf{B} \circ \mathbf{C})_i = \mathrm{tr}(\mathbf{A}\mathbf{B}^T\mathbf{C}_d) \tag{7}$$

with $(\cdot)_d = \mathrm{diag}(\cdot)$, i.e. $(\cdot)_d$ is equal to a square diagonal matrix whose diagonal elements are equal to $(\cdot)$, tr is the matrix trace operator and $(\cdot)^T$ the matrix transpose. Now, the objective function (6), following (7), can be written as:

$$\Delta_j = \mathrm{tr}(\mathbf{Y}_j\mathbf{Y}_j^H\mathbf{W}_d) \tag{8}$$

$$\Delta_j = \mathrm{tr}(\mathbf{D}\mathbf{z}\mathbf{w}_j(\mathbf{D}^*\mathbf{z}\mathbf{w}_j)^T\mathbf{W}_d) \tag{9}$$

with $(\cdot)^H$ denoting the Hermitian matrix.

To maximize Δ_j we take the derivative of its Lagrangian function with respect to $\mathbf{w}_j$. The Lagrangian function, imposes ${\mathbf{w}_j}^T\mathbf{w}_j = 1$, which results into functions that fulfill (3) (when the signal is sphered):

$$L = \mathrm{tr}(\mathbf{D}\mathbf{z}\mathbf{w}_j{\mathbf{w}_j}^T\mathbf{z}^T\mathbf{D}^H\mathbf{W}_d) - \lambda({\mathbf{w}_j}^T\mathbf{w}_j - 1) \tag{10}$$

In order to compute the derivative of (10) with respect to $\mathbf{w}_j$, we use the following property from matrix theory [36]:

$$\frac{\partial}{\partial \mathbf{X}}\,\mathrm{tr}(\mathbf{A}\mathbf{X}\mathbf{X}^T\mathbf{C}) = \big(\mathbf{C}\mathbf{A} + (\mathbf{C}\mathbf{A})^T\big)\mathbf{X} \tag{11}$$

By setting $\mathbf{A} = \mathbf{D}\mathbf{z}$, $\mathbf{X} = \mathbf{w}_j$, $\mathbf{C} = \mathbf{z}^T\mathbf{D}^H\mathbf{W}_d$, letting $\mathbf{K} = \mathbf{C}\mathbf{A}$ and taking into account the unitary DFT matrix property $\mathbf{D}^H = \mathbf{D}^*$, we get:

$$\mathbf{K} = \mathbf{z}^T\mathbf{D}^*\mathbf{W}_d\mathbf{D}\mathbf{z} \tag{12}$$

and the derivative of (10) with respect to $\mathbf{w}_j$ can be written as:

$$\frac{\partial L}{\partial \mathbf{w}_j} = (\mathbf{K} + \mathbf{K}^T)\mathbf{w}_j - 2\lambda\mathbf{w}_j \tag{13}$$

$$\frac{\partial L}{\partial \mathbf{w}_j} = (\mathbf{K} + \mathbf{K}^*)\mathbf{w}_j - 2\lambda\mathbf{w}_j \tag{14}$$

Solving for $\frac{\partial L}{\partial \mathbf{w}_j} = 0$ yields:

$$\frac{\partial L}{\partial \mathbf{w}_j} = 0 \implies \frac{1}{2}(\mathbf{K} + \mathbf{K}^*)\mathbf{w}_j = \lambda\mathbf{w}_j \tag{15}$$

and the solution to (15) are the eigenvectors of the eigenvalue problem:

$$\mathbf{G}\mathbf{w}_j = \lambda\mathbf{w}_j \tag{16}$$

where,

$$\mathbf{G} = \frac{1}{2}(\mathbf{K} + \mathbf{K}^*) \tag{17}$$

The matrix $\mathbf{G}$ is both real, as the sum of a matrix and its conjugate, and symmetric as the sum of a matrix and its transpose. Thus, the eigenvalues of $\mathbf{G}$ are real and its eigenvectors $\mathbf{w}_j$ are orthogonal to each other and therefore, for

Symbol	Description
N	Total number of samples in the time domain of the input signal
$\mathbf{x}[t] \in \mathbb{R}^{1\times I}$	I-dimensional input signal
$\mathbf{y}[t] \in \mathbb{R}^{1\times J}$	J-dimensional slow signal extracted by FSFA
$\mathbf{Y}[k] \in \mathbb{C}$	The Discrete Fourier Transform of $\mathbf{y}[t]$ at frequency k
$\mathbf{W}[k] \in \mathbb{R}$	The FSFA's filter kernel at frequency k
$\mathbf{h}(\mathbf{x}[t]) \in \mathbb{R}^{1\times K}$	A K-dimensional vector function that transforms the input signal to an expansion space
$\mathbf{z}[t] = \mathbf{h}(\mathbf{x}[t]) \in \mathbb{R}^{1\times K}$	The transformed input signal to the expansion space
$\mathbf{w}_j \in \mathbb{R}^{K\times 1}$	The learned FSFA model that transforms the expanded signal to a new feature space where $y_j[t] = \mathbf{z}[t]\mathbf{w}_j$ varies slowly across time
$\mathbf{D} \in \mathbb{C}^{N\times N}$	The Discrete Fourier Transform unitary matrix
$\mathbf{W}_d \in \mathbb{R}^{K\times K}$	A diagonal matrix where each diagonal element equals $\mathbf{W}[k]$ for all $k \in 0, 1, ..., K-1$
Δ_j	FSFA's objective function for the j-th slowest signal

Table 1: Nomenclature of the most important symbols used throughout the paper

a sphered signal, the solution fulfills (4). At this point, it is important to stress the fact that the proposed method, as in [4], extracts signals from the multi-dimensional input that are instantaneous and not filtered versions of the input, despite the objective function involving filtering. The proposed procedure can be seen as follows: we seek signals which are linear combinations of the input in the expansion space, that all are of the same total energy (as imposed by the unit variance constraint) but the remaining energy after passing through the filter $\mathbf{W}[k]$ is maximum. Moreover, from another viewpoint, the filter $\mathbf{W}[k]$ can be seen as a function that imposes a weighting scheme to the extracted signals' frequency spectrum. Higher values of $\mathbf{W}[k]$ indicate frequencies of greater importance, while lower values of $\mathbf{W}[k]$ indicate frequencies with less objective function gain. For a low-pass filter $\mathbf{W}[k]$ we expect to get slow signals (signals that concentrate their energy at low frequencies), while for other types of filters we expect the extracted signals to be medium-slow, fast, or generally having a frequency spectrum close to the characteristics of the filter. The time complexity of FSFA, in case we utilize sparse matrix representation for $\mathbf{W}_d$, is the same as SFA's which is $\mathcal{O}(K^2N)$. Moreover, both methods in the test phase, have the same complexity: $\mathcal{O}(NK^2)$.

4. Relation of the proposed method to SFA

In this section, we study standard SFA in the frequency domain and derive an equivalence of FSFA to SFA. Let $y[t], t \in \{0, 1, ..., N-1\}$, be a real valued signal with its N-Point Fourier transform $\mathbf{Y}[k], k \in \{0, 1, ..., N-1\}$. We are interested in relating the N-Point Fourier transform of its discrete time derivative $x[t] = y[t+1] - y[t], t \in \{0, 1, ..., N-2\}$ and $x[N-1] = 0$ with the N-Point Fourier transform of $y[t]$. Let $W_N = e^{-2\pi j/N}$, with $j = \sqrt{-1}$. Then, the Fourier transform $\mathbf{X}[k]$ of $x[t]$ is:

$$\mathbf{X}[k] = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} x[t] W_N^{tk} \tag{18}$$

$$\mathbf{X}[k] = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-2} y[t+1] W_N^{tk} - \frac{1}{\sqrt{N}} \sum_{t=0}^{N-2} y[t] W_N^{tk} \tag{19}$$

$$\mathbf{X}[k] = (W_N^{-k} - 1)\mathbf{Y}[k] + \frac{y[N-1] W_N^{(N-1)k} - y[0] W_N^{-k}}{\sqrt{N}} \tag{20}$$

$$\mathbf{X}[k] = (W_N^{-k} - 1)\mathbf{Y}[k] + \frac{(y[N-1] - y[0]) W_N^{-k}}{\sqrt{N}} \tag{21}$$

where $W_N^{Nk} = 1$ and $\mathbf{Y}[k] = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} y[t] W_N^{tk}$, the N-Point DFT of $y[t]$. SFA minimizes the following objective function:

$$\Delta_{\text{SFA}} = \sum_{t=0}^{N-2} x[t]^2 = \sum_{t=0}^{N-1} x[t]^2 = \alpha \sum_{k=0}^{N-1} |\mathbf{X}[k]|^2 \tag{22}$$

where in (22) we made use of Parseval's theorem and the fact that $x[N-1] = 0$ by definition, otherwise the sum that we used would be different than SFA's one. Moreover, $\alpha \in \mathbb{R}, \alpha \geq 0$ is a normalization constant depending on the type of DFT used and with out loss of generality it is left out from the rest of the discussion since it does not affect minimization of the objective. Substituting (21) in (22) we obtain:

$$\Delta_{\text{SFA}} = \sum_{k=0}^{N-1} |(W_N^{-k} - 1)\mathbf{Y}[k] + \frac{(y[N-1] - y[0]) W_N^{-k}}{\sqrt{N}}|^2 \tag{23}$$

In the case where $y[0] = y[N-1]$,

$$\Delta_{\text{SFA}} = \sum_{k=0}^{N-1} |(W_N^{-k} - 1)\mathbf{Y}[k]|^2 \tag{24}$$

which is the same as maximizing:

$$\Delta'_{\text{SFA}} = -\sum_{k=0}^{N-1} |(W_N^{-k} - 1)|^2 |\mathbf{Y}[k]|^2 \tag{25}$$

In the case of a centered signal with unit sample variance the term $\sum_{k=0}^{N-1} |\mathbf{Y}[k]|^2$ is constant. Let also $\gamma = |W_N^{N/2} - 1| = 2$. The objective function (25) can be written as:

$$\Delta'_{\text{SFA}} = -\frac{1}{\gamma^2}\left(\sum_{k=0}^{N-1} |W_N^{-k} - 1|^2 |\mathbf{Y}[k]|^2 - \gamma^2 \sum_{k=0}^{N-1} |\mathbf{Y}[k]|^2\right) \tag{26}$$

$$\Delta'_{\text{SFA}} = \sum_{k=0}^{N-1} \left(\frac{\gamma^2 - |W_N^{-k} - 1|^2}{\gamma^2}\right) |\mathbf{Y}[k]|^2 \tag{27}$$

Thus, FSFA is equivalent to SFA when for the signal under evaluation it is

$$y[0] = y[N-1] \tag{28}$$

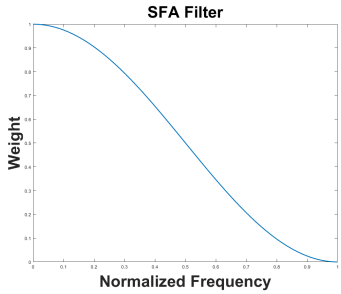
and the filter $\mathbf{W}[k]$ is

$$\mathbf{W}[k] = \frac{\gamma^2 - |W_N^{-k} - 1|^2}{\gamma^2} = 0.5 + 0.5\cos\left(2\pi\frac{k}{N}\right) \tag{29}$$

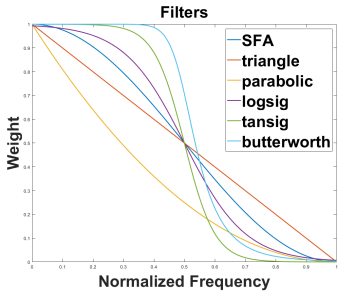
This filter $\mathbf{W}[k]$ is shown in Figure 1(a). The constant γ that was introduced previously was employed in order to normalize $\mathbf{W}[k]$ in the range $[0-1]$. From now on, this filter will be referred as the "SFA Filter".

On the condition of FSFA to SFA equivalence

As shown before in (27), there exist one filter that makes FSFA equivalent to SFA under a specific condition that requires the signal under evaluation $y[t], t \in \{0, 1, ..., N-1\}$ to have the same starting and ending value (28). As $y[t]$ deviates from satisfying this condition, FSFA, with the filter of (29) and SFA, conclude solving two different optimization problems with no direct equivalence between them.



(a) SFA Filter: The filter that makes FSFA equivalent to SFA.



(b) Various FSFA filters that are being used to study filter design.

Figure 1: FSFA Filters.

In this paragraph, we state that for any multidimensional input signal $\mathbf{y}[t], t \in \{0, 1, ..., N-1\}$ satisfying the condition $\mathbf{y}[0] = \mathbf{y}[N-1]$, FSFA can be made equivalent to SFA by applying the filter defined in (29). In fact, what we need to prove is that the condition of $\mathbf{y}[0] = \mathbf{y}[N-1]$ still applies after any instantaneous transformation and thus it still holds after creating the input's expansion space, applying centering and sphering and optimizing against linear combinations of the sphered signal in the expansion space. The proof of the aforementioned statement is actually straightforward, since for any instantaneous function $\mathbf{g}(\mathbf{y}[t])$ (i.e. any function that does not depend on past of future values of t) $\mathbf{y}[0] = \mathbf{y}[N-1] \implies \mathbf{g}(\mathbf{y}[0]) = \mathbf{g}(\mathbf{y}[N-1])$. Q.E.D.

5. Designing the filter kernel

In this section, we approach filter design from the FSFA's perspective. In subsections 5.1, 5.2, 5.3, we give generic facts around the filters of FSFA, while in subsections 5.4 and 5.5, we approach understanding slowness by investigating how FSFA's various filters would order candidate features of the input signal's expansion space in terms of slowness.

5.1. Filter's frequency area of interest

The objective function of FSFA for the signal $y[t] \in \mathbb{R}, t \in \{0, 1, ..., N-1\}$ can be written as follows:

$$\Delta_{\text{FSFA}} = \sum_{k=0}^{N_s-1} |\mathbf{Y}[k]|^2 \mathbf{W}[k] \tag{30}$$

with $N_s = \lceil N/2 \rceil$ (i.e. restricting the summation to the first half of the symmetric frequency spectrum) and with $\mathbf{Y}[k] \in \mathbb{C}$ denoting the DFT of $y[t]$ at frequency k. Let $y[t] = \sum_{i=1}^{K} w_i z_i[t], w_i \in \mathbb{R}$, i.e. $y[t]$ is a linear combination of signals $z_i[t] \in \mathbb{R}$ in an expansion space of dimensionality K. Then, the following inequality holds:

$$\big|\mathbf{Y}[k]\big| \leq \sum_{i=1}^{K} w_i \big|\mathbf{Z}_i[k]\big| \tag{31}$$

with $\mathbf{Z}_i[k]$ the DFT of the signal $z_i[t]$. From (31) it follows that if for all $\mathbf{Z}_i[k], i \in \{1, 2, ..., K\}$, $|Z_i[k]| = 0$ for $k \geq k_m$, then $|Y[k]| = 0$ for all $k \geq k_m$ (always for the first half of the frequency spectrum), i.e. $y[t]$ is band-limited with its baseband bandwidth being at most equal to the wider baseband bandwidth of its components in the expansion space. Thus, the FSFA filter should weight the frequency spectrum appropriately for $k \leq k_m$. In practice, k_m is computed as $k_m = \max(k_1, k_2, ..., k_K)$ with

$$k_i = \arg\min_k \Big(\sum_{n=0}^{k} |\mathbf{Z}_i[n]|^2 \geq \tau \sum_{n=0}^{N_s-1} |\mathbf{Z}_i[n]|^2\Big) \tag{32}$$

with a common choice for $\tau \in (0, 1]$ being $\tau = 1 - \epsilon$, with $\epsilon \in \mathbb{R}^+$ being a small constant. We name the frequency zone of $[0, k_m]$ as the filter's "frequency area of interest".

5.2. Filters with the same solution to the FSFA optimization problem

Let $\mathbf{W}_1[k]$ and $\mathbf{W}_2[k]$ denote two filter kernels with $\mathbf{W}_2[k] = \alpha \mathbf{W}_1[k] + \beta, \alpha \in \mathbb{R}_0^+, \beta \in \mathbb{R}$. Since the search space of FSFA consists of signals of the same total

energy, as imposed by (3), it is easy to prove that the maximization of the FSFA objective for $\mathbf{W}_1[k]$ and $\mathbf{W}_2[k]$ has the exact same solutions. In other words, any two filters that are a linear transformation of each other (with $\alpha > 0$) measure signal slowness in the exact same way.

5.3. Low-pass filter normalization

For a given input signal from which we need to extract slow signals, let $[0, k_m], k_m \in \mathbb{N}, k_m < N_s$ be the filter's frequency area of interest, as discussed in Section 5.1. When studying how different filters affect the slow signal output, it is all about realizing how the filters relate the different frequencies of the frequency spectrum of the extracted signal. In order to ease the study of those filters it is important that all the filters weight the frequency spectrum with values of the same interval. For this paper we chose this interval to be $[0, 1]$ and as already discussed in Section 3 we require $0 \leq \mathbf{W}[k] \leq 1$. Apart from those constraints and to aid with the study of different filters, in this subsection we additionally impose the requirement that the interval of $[0, 1]$ is entirely covered by the filter kernel. When the filter kernel entirely covers the interval $[0, 1]$ in the filter's frequency area of interest, we say that this filter is normalized.

More formally, let $\mathbf{W}[k]$ denotes a low-pass filter kernel for which $k_1 \leq k_2 \implies \mathbf{W}[k_1] \geq \mathbf{W}[k_2]$. We can normalize this filter in its frequency area of interest by choosing

$$\alpha = \frac{1}{\mathbf{W}[0] - \mathbf{W}[k_m]} \tag{33}$$

and

$$\beta = 1 - \alpha \mathbf{W}[0] \tag{34}$$

Then, the normalized filter $\mathbf{W}_n[k]$ is given by the following formula:

$$\mathbf{W}_n[k] = \alpha \mathbf{W}[k] + \beta \tag{35}$$

It is easy to show that $\mathbf{W}_n[0] = 1$ and $\mathbf{W}_n[k_m] = 0$. As already discussed in Section 5.2, $\mathbf{W}_n[k]$ and $\mathbf{W}[k]$ will have the same solution to the FSFA optimization problem. Filter normalization is very important. In the experimental result section (section 6) we normalize the SFA filter of (29) in the filter's frequency area of interest in order to better explain the resulted outcomes.

5.4. Slowness: A filter dependent measurement

In FSFA, the slowness of a signal depends on the selected filter. It is the selected filter that determines how slow a signal is and what is considered slow for one filter may be considered fast for another filter. Thus, in general, no direct comparison of signal slowness between different filters is possible, since the objective function is different for each filter and there is no meaning in comparing the values of the objective function for different filters. However, we can understand the behavior of each filter by observing how each one differentiates when applied to the same input.

We begin our study by picking 2 different signals $y_1[t] \in \mathbb{R}$ and $y_2[t] \in \mathbb{R}$, both containing two frequency components, namely f_1 and f_4 for $y_1[t]$ and f_2, f_3 for $y_2[t]$ with $f_1 \leq f_2 \leq f_3 \leq f_4$. Thus, $y_1[t] = \sin(2\pi f_1 t) + c \cdot \sin(2\pi f_4 t)$, $y_2[t] = \sin(2\pi f_2 t) + c \cdot \sin(2\pi f_3 t)$ with $0 \leq c \leq 1$. In other words, $y_1[t]$ contains one very low and one very high frequency components, while $y_2[t]$ contains two medium frequency components. In this study we are interested to investigate how different filters order $y_1[t]$ and $y_2[t]$ in terms of slowness.

Firstly, we apply a hanning window [37] to those signals in order to force the FSFA to SFA equivalence constraint (28) without introducing new frequency content. We also normalize the signals to have zero mean and unit variance and thus the same energy. Since the signals $y_1[t]$ and $y_2[t]$ have both the same energy, they can be directly compared for their slowness, with respect to a selected filter, using the FSFA's objective function (30). For an arbitrary filter $\mathbf{W}[k]$, $0 \leq \mathbf{W}[k] \leq 1$, let Δ_1 indicate the FSFA's objective function score for the signal $y_1[t]$ and Δ_2 indicate the FSFA's objective function score for the signal $y_2[t]$, with the FSFA objective function score being given by the formula of (30). By letting k_1, k_2, k_3 and k_4 denote the corresponding frequencies of f_1, f_2, f_3 and f_4 in the discrete domain, $y_1[t]$ is classified slower than $y_2[t]$ for a given filter $\mathbf{W}[k]$ when $\Delta_1 > \Delta_2$. Since our engineered signals $y_1[t]$ and $y_2[t]$ are both composed by two sinusoids, comparing their slowness can be approximated by

the following formula:

$$\begin{aligned}
\Delta_1 > \Delta_2 \implies & \\
& |\mathbf{Y}_1[k_1]|^2\mathbf{W}[k_1] + |\mathbf{Y}_1[k_4]|^2\mathbf{W}[k_4] > \\
& |\mathbf{Y}_2[k_2]|^2\mathbf{W}[k_2] + |\mathbf{Y}_2[k_3]|^2\mathbf{W}[k_3] \implies \\
& |\mathbf{Y}_1[k_1]|^2(\mathbf{W}[k_1] + c^2\mathbf{W}[k_4]) > \\
& |\mathbf{Y}_2[k_2]|^2(\mathbf{W}[k_2] + c^2\mathbf{W}[k_3]) \implies \\
& \mathbf{W}[k_1] + c^2\mathbf{W}[k_4] > \mathbf{W}[k_2] + c^2\mathbf{W}[k_3] \quad (36)
\end{aligned}$$

where in (36) we used the following approximations: $|\mathbf{Y}_1[k_4]| = c|\mathbf{Y}_1[k_1]|$, $|\mathbf{Y}_2[k_3]| = c|\mathbf{Y}_2[k_2]|$ and $|\mathbf{Y}_1[k_1]| = |\mathbf{Y}_2[k_2]|$, with $\mathbf{Y}_1[k]$ denoting the DFT of $y_1[t]$ and $\mathbf{Y}_2[k]$ denoting the DFT of $y_2[t]$. By examining (36), we observe that whether $y_1[t]$ is slower than $y_2[t]$ depends on the weights that the filter kernel imposes to the frequencies k_1, k_2, k_3 and k_4 as well as the magnitude of the constant c.

For each filter and for each one of the experiments that we conduct, we pick specific values for f_1, f_2 and f_3 and search the maximum value of f_4, denoted by f_c, that makes the specific filter to consider $y_1[t]$ to be slower than $y_2[t]$. For any value of f_4 greater than f_c, $y_1[t]$ is considered faster than $y_2[t]$ from the perspective of the specific filter that is being studied. In the marginal case where $f_4 = f_c$, it will be $\Delta_1 = \Delta_2$ and thus the two signals are considered to be equally slow, while for $f_4 > f_c$ it will be $\Delta_1 < \Delta_2$.

By studying further (36) and for a monotonically decreasing filter $\mathbf{W}[k]$ (i.e. a low pass filter) we can draw some interesting conclusions:

$$\begin{aligned}
\Delta_1 > \Delta_2 \implies & \\
& \mathbf{W}[k_1] - \mathbf{W}[k_2] > c^2(\mathbf{W}[k_3] - \mathbf{W}[k_4]) \implies \\
& \frac{1}{c^2}(\mathbf{W}[k_1] - \mathbf{W}[k_2]) > \mathbf{W}[k_3] - \mathbf{W}[k_4] \implies \\
& \mathbf{W}[k_4] > \mathbf{W}[k_3] - \frac{1}{c^2}(\mathbf{W}[k_1] - \mathbf{W}[k_2]) \quad (37)
\end{aligned}$$

a) When $\mathbf{W}[k_2] \simeq \mathbf{W}[k_1]$ then $\mathbf{W}[k_c] \to \mathbf{W}[k_3]$ and when $\mathbf{W}[k]$ is monotonic,

$k_c \to k_3$ (where k_c was used to denote the discrete frequency corresponding to f_c).

b) In the case where k_3 is such that $\mathbf{W}[k_3] < \frac{1}{c^2}(\mathbf{W}[k_1] - \mathbf{W}[k_2])$, k_c can be arbitrary large and still $y_1[t]$ will be considered slower than $y_2[t]$.

c) If k_1 belongs to the filter's pass-band (i.e. $\mathbf{W}[k_1] \simeq 1$), then, no matter how large k_4 is, $y_1[t]$ is considered slower than all the signals $y_2[t]$ being composed from k_2 and k_3 for which $\mathbf{W}[k_2] < 0.5$ and $\mathbf{W}[k_3] < 0.5$. This observation holds for all filters $\mathbf{W}[k]$ and for any $0 \leq c \leq 1$.

5.5. Slowness: Experimentation with practical filters

In the current subsection, we conduct the previously mentioned experiment setup by using practical filters. The filters that we chose to experiment with are depicted in Figure 1(b). From this Figure, it can be noticed that the filters can be ordered with respect to their pass-band width from narrower to wider as follows: a) parabolic, b) triangle, c) SFA, d) logsig, e) tansig and f) butterworth. We execute three experiments in total and each of the experiment's results is illustrated in Table 2. The sampling frequency for all experiments is set to 50 Hz, the constant c is set to $c = 0.3$ and frequency normalization is performed by dividing each frequency value by half the sampling frequency.

For the first experiment, we choose $f_1 = 0.25$Hz (normalized: 0.01), $f_2 = 2.5$Hz, (normalized: 0.1) and $f_3 = 5$Hz (normalized: 0.2). As shown in Table 2, the application of rule **(a)** is observed. The wider the filter's pass-band the more $\mathbf{W}[k_2]$ is close to $\mathbf{W}[k_1]$ and thus, the more k_c tends to k_3, with the two extremes occurring in the butterworth/tansig and the parabolic filters. In the second experiment it is set $f_1 = 2.5$Hz (normalized: 0.1), $f_2 = 5$Hz (normalized: 0.2) and $f_3 = 7.5$Hz (normalized 0.3). Occurrences of rule **(b)** are noted for SFA, triangle and parabolic filters while rule **(a)** is fulfilled for logsig, tansig and butterworth filters similar to the previous experiment. Finally, in the third experiment, where $f_1 = 2.5$Hz (normalized: 0.1), $f_2 = 15$Hz (normalized: 0.6) and $f_3 = 17.5$Hz (normalized: 0.7), we see the application of rule **(c)**. All filters consider $y_1[t]$ to be slower than $y_2[t]$ no matter how large f_c is, because, as it

Frequency	Hz	Normalized
Experiment #1		
f_c (SFA)	6.3672	0.25469
f_c (triangle)	10.2734	0.41094
f_c (parabolic)	13.0078	0.52031
f_c (logsig)	5.8984	0.23594
f_c (tansig)	5.0391	0.20156
f_c (butterworth)	5.0391	0.20156
Experiment #2		
f_c (SFA)	23.7354	0.94941
f_c (triangle)	24.8975	0.9959
f_c (parabolic)	24.9658	0.99863
f_c (logsig)	11.7725	0.4709
f_c (tansig)	8.4912	0.33965
f_c (butterworth)	7.6709	0.30684
Experiment #3		
f_c (all filters)	24.9707	0.99883

Table 2: Slowness comparison between different filters. The maximum frequency f_c for which $y_1[t]$ is considered slower than $y_2[t]$. Results for experiments #1, #2 and #3.

can be seen in Figure 1(b), $\mathbf{W}[k] < 0.5$ for all filters and for all normalized frequencies above 0.6. The resulting signals for all the experiment are being depicted in Figures 2, 3 and 4. In those Figures we plot the signal $y_2[t]$ along with $y_1[t]$ for the value of $k_4 = k_c$, with k_c varying depending on the filter. Thus, we plot $y_1[t]$ once for each filter.

6. FSFA In Practice: Experimental Results

In this section, the experimental evaluation of the proposed method is detailed. We used 4 sets of experiments to pinpoint the different merits of the

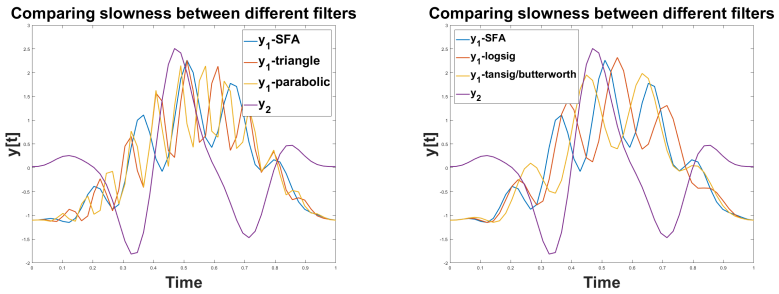


Figure 2: Comparing slowness between different filters. Experiment #1.

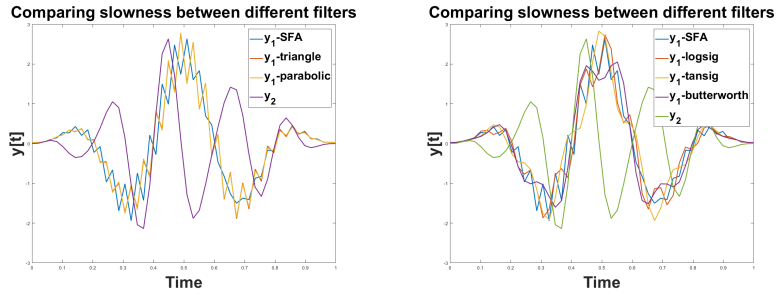


Figure 3: Comparing slowness between different filters. Experiment #2.

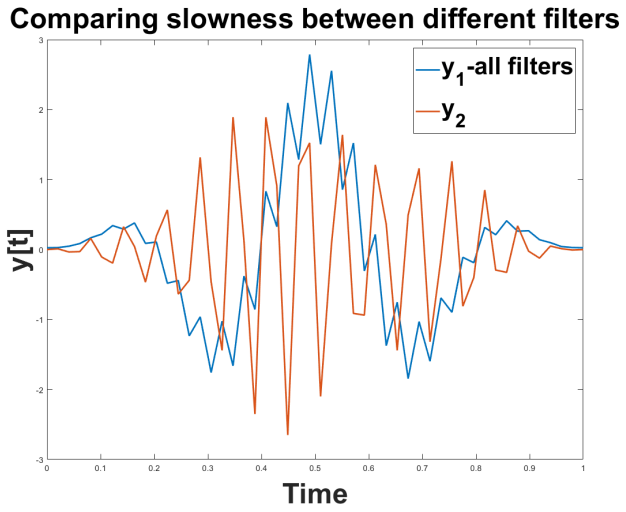


Figure 4: Comparing slowness between different filters. Experiment #3.

method. The first experiment is conducted on synthetic data and establishes the fact that FSFA extracts different slow features when using different low pass filters. A short theoretical discussion is also given that explains the experiment's results. In the second experiment, we apply FSFA in real world data obtained from a known CNN to showcase that various filters lead to different extracted features as in the synthetic case. In the third experiment, we provide a comparison on the classification performance of the features extracted by FSFA using various filters and features extracted by standard SFA. Finally, in the fourth experiment, we demonstrate that a simple, shallow, neural network comprised of four FSFA nodes and one SFA node can transform the features of the aforementioned CNN in a new feature space where classification performance in video action recognition can be improved.

6.1. Experiment A: Demonstrate FSFA output variation depending on the selected low-pass filter on synthetic data

Similar to SFA, FSFA's output depends on the input signal as well as the expansion space used. Additionally, FSFA's output depends on the chosen filter kernel. Our extensive experimentation with FSFA has shown that for simple input multi-dimensional signals, different low pass filters more or less extract the same slow signals. To demonstrate the generalized nature of FSFA we are forced to use a slightly more complex input signal than a signal being composed from very simple sinusoid components. However, we will use some simple mono-dimensional signals as a basis to construct a slightly more complex input multi-dimensional signal.

For the purposes of this experiment, let

$$r_i[t] = \sum_{j=1}^{4} A_{ij}[\sin(2\pi f_j t + \phi_{ij})]^{p_{ij}} \tag{38}$$

with $i = \{1, 2, 3\}$. The parameters $A_{ij}, \phi_{ij}, p_{ij}$ are all given in Table 4, while the frequencies f_j are set as: $f_1 = 0.1$Hz, $f_2 = 0.7$Hz, $f_3 = 1.2$Hz and $f_4 = 1.7$Hz. The multi-dimensional input signal $\mathbf{x}[t]$ that is used in this experiment is composed out of 3 components $x_1[t] \in \mathbb{R}, x_2[t] \in \mathbb{R}$ and $x_3[t] \in \mathbb{R}$ and is given

by the formula: $\mathbf{x}[t] = \left[x_1[t], x_2[t], x_3[t]\right]$ with $x_1[t] = r_2[t] \cdot r_3[t], x_2[t] = r_2[t] + r_3[t], x_3[t] = r_3[t] \cdot r_1[t]$. Moreover, we apply a hanning window [37] to the signal components $x_1[t], x_2[t]$ and $x_3[t]$, in order to fulfill the FSFA to SFA equivalence criterion of (28), without additionally introducing any new frequency content to the signal. Finally, we center the signals and make them of unit variance. The final input signal components are depicted in Figure 5(a). Thereafter, we ask FSFA to extract slow signals from the multi-dimensional input signal $\mathbf{x}[t]$ using different low-pass filters. The filters used in this experiment are: a) triangle, b) trapezoid c) SFA and d) tansig, all depicted in Figure 5(b). In Figure 6(a), a zoomed in version of the filters in their frequency "area of interest" is given, while their normalized versions are depicted in Figure 6(b). The expansion space used in this experiment is "SExp" [5] which for a 3-component input signal has dimensionality 6, and thus the number of slow signals extracted by FSFA is 6.

The extracted slow signals for each filter along with their power spectrum are illustrated in Figures 7 and 9, 10. The filter's frequency area of interest for this input signal and in this expansion space is approximately the interval $[0, 0.16]$, computed using (32) with $\tau = 0.999$. This interval is given as a normalized frequency range where 0 corresponds to the 0 frequency and 1 corresponds to half the sampling frequency of 150Hz. From all the provided Figures and especially from Figures 7, and 9(a), 9(b), it is made clear that FSFA's slow signal extraction significantly depends on the chosen filter kernel. In the rest of the section we provide comments that are being reasoned from examining Figure 7 (i.e the slowest extracted signal for all filters).

By closely looking at Figure 6(b), it is observed that the tansig filter has the wider pass-band of all the filters participating in the experiment (the band where $\mathbf{W}[k] \simeq 1$) and it is the one that mostly resembles the ideal low pass filter. The signal extracted by tansig has the narrowest baseband bandwidth of all the signals extracted using the rest of the filters, as shown in Table 3, where for the computation of the signals' baseband bandwidth we used (32) with $\tau = 0.9999$. The tansig filter does not pick any of the signals extracted with the other filters because the other extracted signals leak energy to higher frequencies (> 0.06)

which mainly lies in the tansig's filter transition or stop band where $\mathbf{W}[k] \ll 1$, and thus there is lesser objective function gain at those frequencies. (See also a zoomed version of the power spectrum in Figure 8.)

On the other hand, the triangle filter, being the filter with the narrowest pass-band, is very picky in extracting a slow signal with a strong frequency component at a very low frequency. This result aligns very well with the observation rules that we presented in Section 5 about filter design, where for a low frequency k_1, $y_1[t]$ was always considered slower than all the signals $y_2[t]$ containing frequencies k_2 and k_3 with $k_1 \leq k_2 \leq k_3$, no matter how much "energy leakage" occurred in the higher frequencies which was controlled by k_4.

If the tansig filter (widest pass-band) and the triangle filter (narrowest pass-band) constitute the two extremes, then the SFA and trapezoid filters stands in between. This can be clearly seen in Figure 7. If the signals extracted by the triangle and tansig filters are the two reference signals, then the signals extracted by SFA and trapezoid filters, look like a signal that is a smooth transition from the one reference signal to the other. Moreover, as depicted in Figure 6(b), for the normalized frequency range of $[0, 0.03]$ it is $\mathbf{W}_{\text{trapezoid}}[k] \leq \mathbf{W}_{\text{SFA}}[k]$, with $\mathbf{W}_{\text{trapezoid}}[k]$ denoting the trapezoid filter and $\mathbf{W}_{\text{SFA}}[k]$ denoting the SFA filter. In other words, the trapezoid filter is pickier for lower frequency components than SFA.

Table 3 also gives us an interesting observation. The pickier a filter is about low frequency components, the higher the baseband bandwidth of the extracted signal. While this is certainly not necessary always the case, it is likely to happen. To explain this interesting fact, we remind the reader that all signals extracted by FSFA (and SFA) are of the same total energy. If we carefully examine our intentions when using a very picky filter (a filter with a very narrow pass-band) we would actually discover that this way we are coercively looking to extract signals that contain very low frequency components. In the very common case where not all the energy can actually be concentrated in the lowest frequencies, what we get in fact, is a signal with a strong low frequency component and with some other energy being leaked in higher frequencies which

Filter	Triangle	Trapezoid	SFA	Tansig
Baseband Bandwidth	0.16	0.1428	0.11607	**0.0982**

Table 3: Experiment A: Slow component #1 signal baseband bandwidth computed by (32) with $\tau = 0.9999$ and normalized to the frequency range of [0-1] (with 0 corresponding to the zero frequency and 1 corresponding to half the sampling frequency).

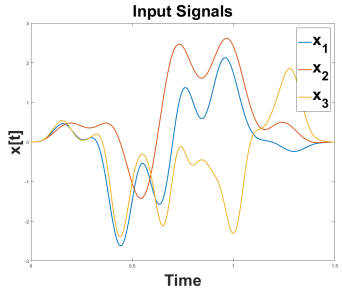
$A_{11} = 0.4$	$A_{12} = 0.7$	$A_{13} = 0$	$A_{14} = 0.5$
$A_{21} = 0$	$A_{22} = 0$	$A_{23} = 0.6$	$A_{24} = 0.3$
$A_{31} = 0.3$	$A_{32} = 0$	$A_{33} = 0$	$A_{34} = 0.7$
$\phi_{11} = 0$	$\phi_{12} = \pi/2$	$\phi_{13} = 0$	$\phi_{14} = 0$
$\phi_{21} = 0$	$\phi_{22} = 0$	$\phi_{23} = \pi/6$	$\phi_{24} = 0$
$\phi_{31} = 0$	$\phi_{32} = 0$	$\phi_{33} = 0$	$\phi_{34} = \pi/8$
$p_{11} = 1$	$p_{12} = 1$	$p_{13} = 1$	$p_{14} = 3$
$p_{21} = 1$	$p_{22} = 1$	$p_{23} = 1$	$p_{24} = 1$
$p_{31} = 1$	$p_{32} = 1$	$p_{33} = 1$	$p_{34} = 2$

Table 4: Experiment A: Parameters A_{ij}, ϕ_{ij}, and $p_{ij}, i \in \{1, 2, 3\}, j \in \{1, 2, 3, 4\}$.

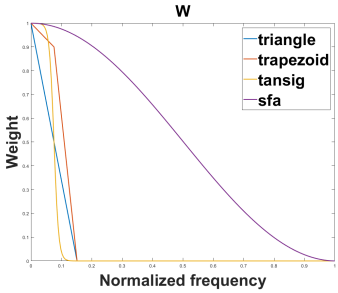
in many cases results into signals with high baseband bandwidth. The more we relax our constraint to be picky at low frequencies, and thus selecting filters with wider pass-bands, we actually give room to the energy to be concentrated at low and mid frequencies potentially avoiding leakage to the higher ends of the spectrum.

6.2. Experiment B: Demonstrate FSFA output variation depending on the selected low-pass filter on real-world data

In this subsection, we apply FSFA, with various filters, on real-world data obtained by a well-known CNN (C3D [38]) and justify that the extracted features depend on the choice of the filter kernel. For completeness, we also provide the features extracted by SFA for the same input. Regarding the data, we use the standard split #1 of the UCF-101 [39] video action recognition dataset. In sub-

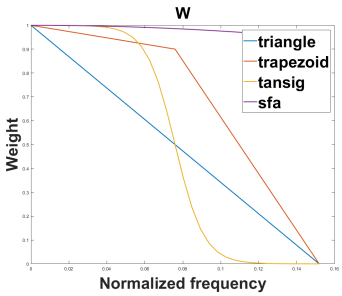


(a) Experiment A: Input signals $x_1[t]$, $x_2[t]$ and $x_3[t]$.

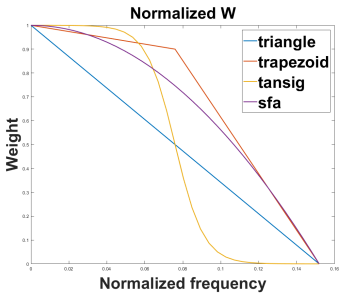


(b) Experiment A: Different FSFA low pass filters.

Figure 5: Experiment A: Input signals and FSFA low pass filters.

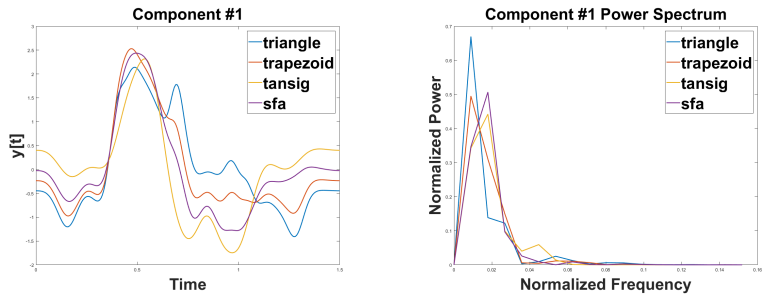


(a) Experiment A: Illustration of the filters of Figure 5(b) by zooming in the frequency area of interest.



(b) Experiment A: Normalized FSFA low-pass filters used to drive the experiment, in the frequency area of interest.

Figure 6: Experiment A: FSFA low-pass filters used to drive the experiment.



(a) Signals in time domain.

(b) Signals' power spectrum.

Figure 7: Experiment A: Slow component #1.

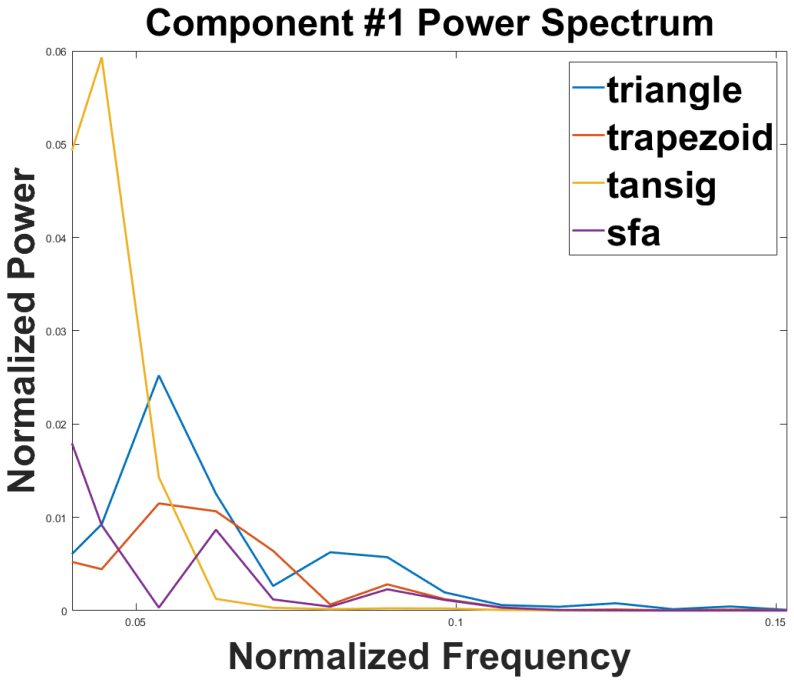
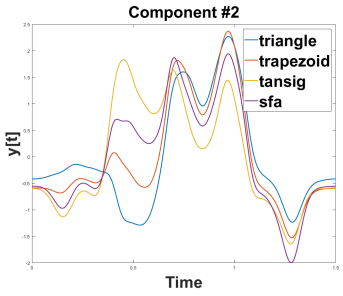
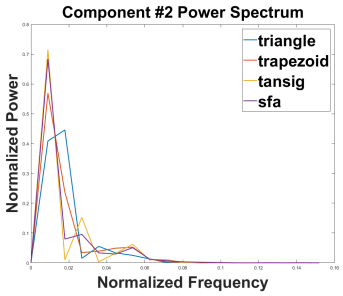


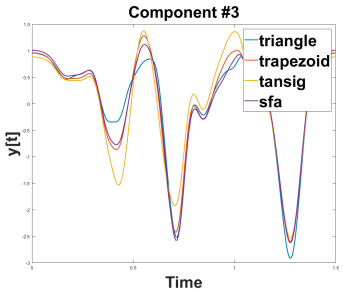
Figure 8: Experiment A: Slow component #1 power spectrum zoom in at higher frequencies.



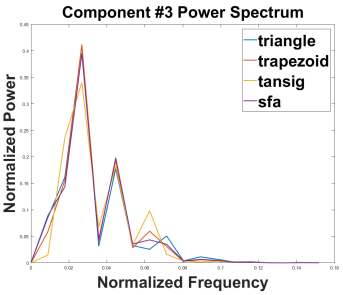
(a) Signals in time domain.



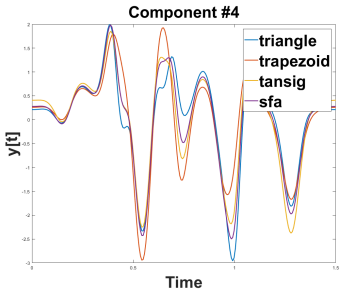
(b) Signals' power spectrum.



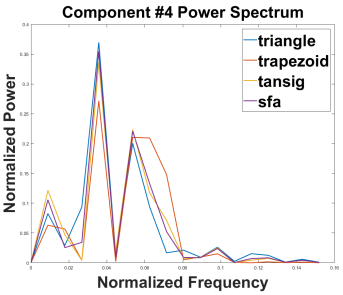
(c) Signals in time domain.



(d) Signals' power spectrum.

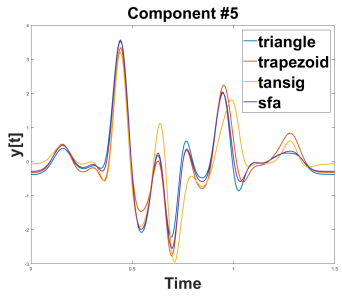


(e) Signals in time domain.

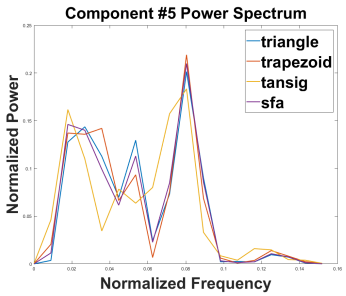


(f) Signals' power spectrum.

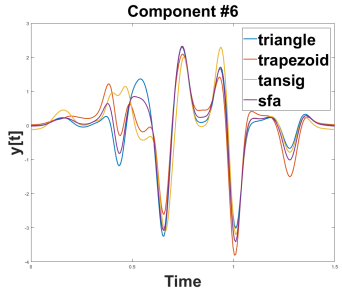
Figure 9: Experiment A: Slow components #2 - #4.



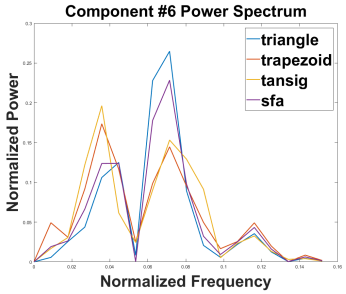
(a) Signals in time domain.



(b) Signals' power spectrum.

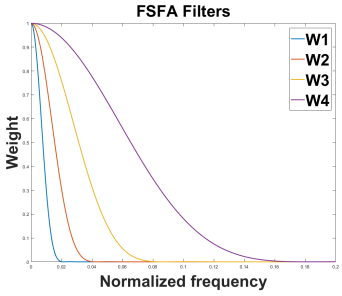


(c) Signals in time domain.

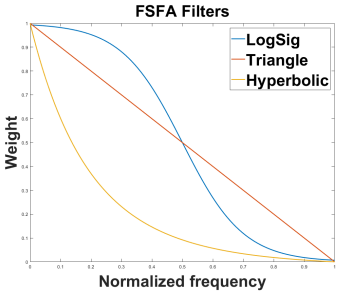


(d) Signals' power spectrum.

Figure 10: Experiment A: Slow components #5 - #6.

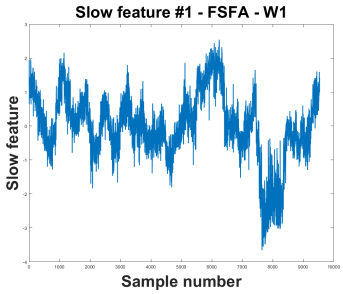


(a) Experiments B and D: The 4 FSFA filters used (W1, W2, W3, W4).

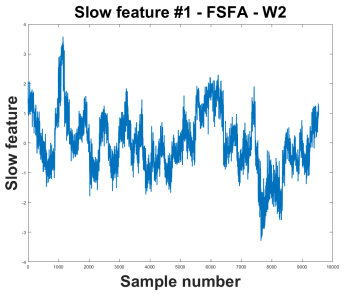


(b) Experiment C: The 3 FSFA filters used (LogSig, Triangle and Hyperbolic).

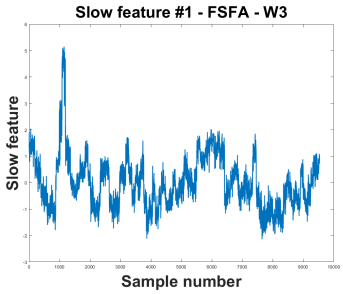
Figure 11: The FSFA filters used in Experiments B, C and D.



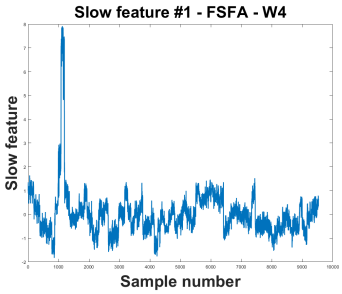
(a) FSFA's slowest feature for filter W1



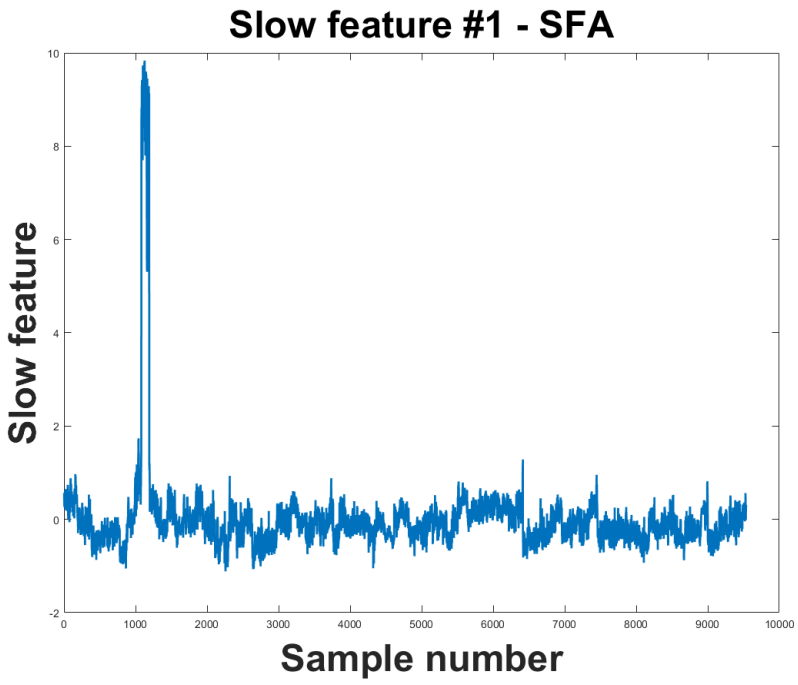
(b) FSFA's slowest feature for filter W2



(c) FSFA's slowest feature for filter W3

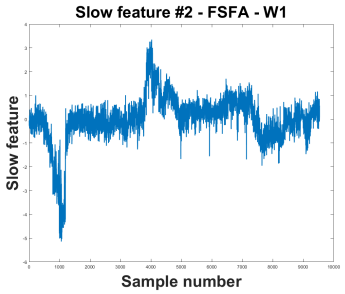


(d) FSFA's slowest feature for filter W4

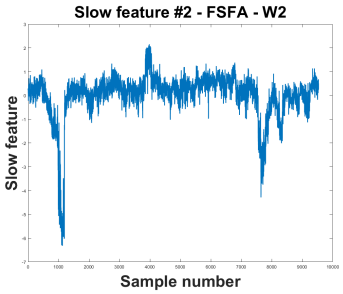


(e) SFA's slowest feature

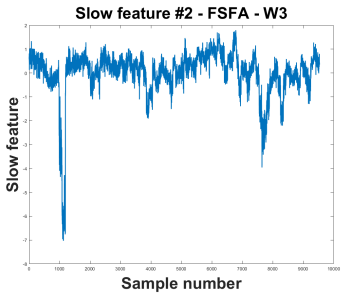
Figure 12: Experiment B: The slowest feature extracted by FSFA using 4 filters (W1, W2, W3, W4) and the slowest feature extracted by standard SFA.



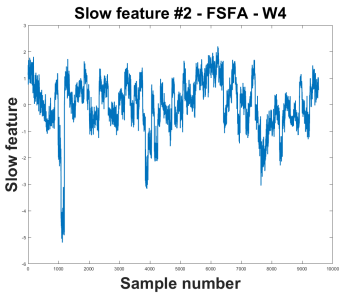
(a) FSFA's 2nd slowest feature for filter W1



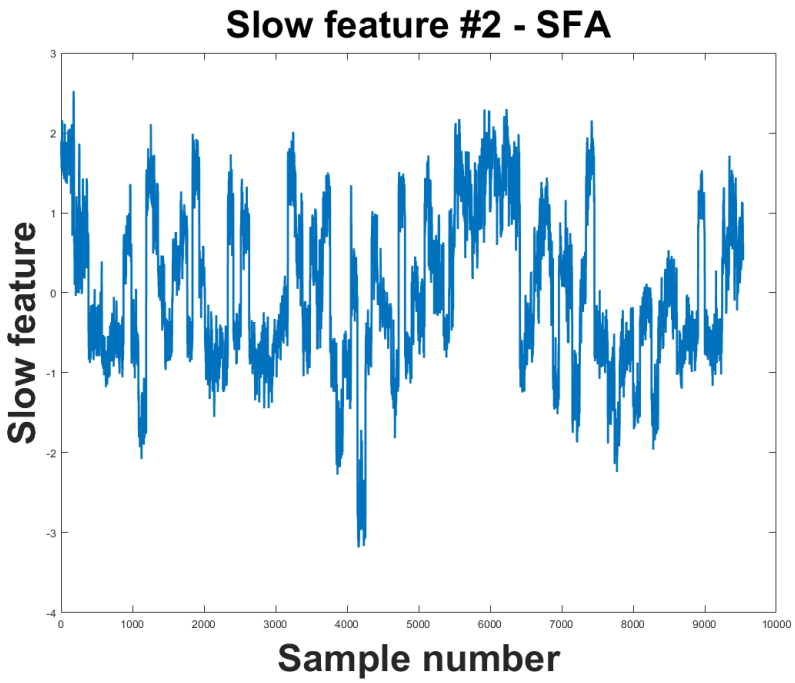
(b) FSFA's 2nd slowest feature for filter W2



(c) FSFA's 2nd slowest feature for filter W3



(d) FSFA's 2nd slowest feature for filter W4



(e) SFA's 2nd slowest feature

Figure 13: Experiment B: The second slowest feature extracted by FSFA using 4 filters (W1, W2, W3, W4) and the second slowest feature extracted by standard SFA.

section 6.2.1 we describe the data pre-processing procedure that we used, while in subsection 6.2.2 we provide figures depicting the slowest extracted features.

6.2.1. Data pre-processing

In order to apply the unsupervised methodology of SFA/FSFA in a classification task, such as the task of Video Action Recognition, we follow the procedure described in [21] along with some pre-processing steps. In particular, for each video of the dataset, we extract C3D features with standard video clip size of 16 and frame overlapping of 12 frames (i.e. a frame stride of 4 steps). This procedure, extracts T feature vectors of size 1×4096 for each video, where T corresponds to the number of clips inside the video. We average those features across T, as it is described in standard C3D, and perform L2 normalization, to produce one final feature vector per video, of size 1×4096. Subsequently, we perform PCA dimensionality reduction, in order to reduce the videos' feature vector size from 1×4096 to 1×608, which is the dimensionality ($D = 608$) that explains the 85% of the features' variance. Since we are performing this experiment on split #1, which has $N = 9537$ videos in its training set, we construct a new multi-dimensional signal $\mathbf{x} \in \mathbb{R}^{N \times D}$ containing all the final video features.

As described in [21], we construct $\mathbf{x}$, by placing features corresponding to videos of the same class in adjacent positions. In this way, $\mathbf{x}$ can be seen as a multi-dimensional signal where the class identity (i.e. the action being performed in the video) becomes the slowest varying latent feature, across its first dimension (i.e. $\mathbf{x}$'s rows). Thus, the input to FSFA/SFA is the aforementioned matrix $\mathbf{x}$, with the class identity being implicitly encoded in the order of the video features. In that case, and according to the theory of [4], the latent variable corresponding to the class identity as a function of the video's feature can be extracted by the SFA/FSFA algorithm. In the above procedure, the dimensionality reduction step through PCA, is mandatory for SFA/FSFA systems, because the number of samples in the training set ($N = 9537$) are quite close to the dimensionality of the original C3D feature vector (4096), and in that case, the SFA/FSFA algorithms easily overfit to their optimal solutions.

	Baseline C3D with our SVM	SFA	FSFA LogSig Filter	FSFA Triangle Filter	FSFA Hyperbolic Filter
Train	99.86%	99.91%	99.92%	99.89%	99.92%
Test	80.2%	81.6%	81.39%	**81.68**%	**82.08**%

Table 5: SFA vs FSFA classification accuracy on the UCF-101 dataset, split #1.

6.2.2. Feature visualization

After applying the steps described in subsection 6.2.1, we give the matrix $\mathbf{x}$ as input to FSFA and SFA in order to extract slowly varying features. For the FSFA algorithm, we use the 4 filters depicted in Figure 11(a), namely W1, W2, W3 and W4. In Figure 12, we provide the slowest feature extracted by FSFA using each one of the 4 filters, as well as the slowest feature extracted by standard SFA, while in Figure 13 the second slowest feature for the same cases is depicted. The expansion type used in all of our experiments is "Identity"[5].

By inspecting the figures, it is evident that FSFA extracts different features depending on the chosen filter. The variation of FSFA's output with respect to the chosen filter, in the real-world dataset, is even more noticeable than in the simple synthetic case. Conclusively, the slow feature extraction conducted by FSFA, certainly depends on the chosen filter kernel, no matter the source type of the input data (i.e synthetic or real-world).

6.3. Experiment C: Comparing classification performance between FSFA and SFA in Video Action Recognition

In this section, we set up an experiment in order to evaluate the classification performance of the features extracted by FSFA and compare against the features extracted by standard SFA. Once again, for the dataset of this experiment, we use standard UCF-101 split #1 and for SFA and FSFA, we conduct all the pre-processing steps described in subsection 6.2.1. During training, we

present the multidimensional signal $\mathbf{x}$ to SFA/FSFA and learn the functions that transform the input signal $\mathbf{x}$ to a new space where the extracted features vary slowly. During testing, the learned functions are applied to the C3D video feature instantaneously in order to bring it to the space that was learned during training. The expansion space used for SFA & FSFA, is "Identity"[5], while after applying the slow feature extraction algorithms we reduce the dimensionality of the original feature space ($D = 608$), to about its half, by always keeping the 300 slowest features. Finally, we perform L2-Normalization on the 1×300 sized slow features, as in standard C3D, and perform classification through a standard linear SVM.

In this experiment, the feature transformation pipeline is exactly the same for both SFA and FSFA. For the baseline comparison, we use directly the C3D features extracted by the pre-trained network provided by the authors of [38] (with the single net option), average them across time for each video, perform L2 normalization in each one of them, and pass them to the linear SVM as it was done in [38]. The linear SVM that we use for classification, is not tuned in any way for any of the methods and its configuration is kept the same across evaluations. For FSFA, we used the 3 filters depicted in Figure 11(b). The classification performance (training and test accuracies) of SFA, FSFA and the C3D baseline is illustrated in Table 5. Both slow feature extraction methods improve classification performance over the standard baseline. Comparing between them, we find that SFA outperforms FSFA in the case of the "LogSig" filter, while for the other two filters, i.e. "Triangle" and "Parabolic", the opposite is true. This justifies that not all FSFA filters perform better than original SFA. In our experimentation, we found that pickier low pass filters (filters with a strong scheme, favoring lower frequencies) perform better, as in the current case.

Concerning the C3D baseline, we have to note, that we could not reproduce the exact performance reported in [38], which for UCF-101 and for the single net C3D is 82.3%. Based on the knowledge that we have, and since we have used the exact implementation provided by the respective authors for the C3D

	Baseline C3D (reported in [38])	Baseline C3D with our SVM	Our network - only FSFA nodes	Our network - FSFA & SFA nodes
Train	-	99.86%	99.80%	99.88%
Test	82.3%	80.2%	82.52%	**83.19**%

Table 6: Classification accuracy of a shallow network employing SFA and FSFA nodes on the UCF-101 dataset, split #1.

network, we believe that this shortcoming ought to be due to some misalignment between the parameters of our SVM implementations. However, the scope of the present paper is mostly about introducing a generic theoretical concept rather than tuning the SVM parameters to achieve state of the art results. In this paper, FSFA and SFA are used as post-processing steps on the top of C3D (and potentially on the top of any other feature extractor) and our arguments for improving the baseline are to be taken from this point of view.

6.4. Experiment D: Using a simple, shallow, FSFA & SFA network to improve classification performance in Video Action Recognition

In this experiment, we combine four FSFA nodes with the filters presented in Figure 11(a) and a single SFA node, into a shallow neural network that transforms the features extracted by C3D [38] to a new feature space, where classification performance in the Video Action Recognition dataset UCF-101 [39], split #1, using a liner SVM, is further improved.

Initially, we follow all the pre-processing steps described in subsection 6.2.1. Subsequently, the input matrix $\mathbf{x}$ is given independently to each FSFA node, performing slow feature extraction using its assigned filter (i.e. W1, W2, W3 or W4). The expansion space for all FSFA nodes is set to "Identity"[5]. In the training phase, the network aims to learn the functions that transform the input signal to a new space where the features of the same class are close to each other, since they are spatially adjacent and are optimized to vary slowly

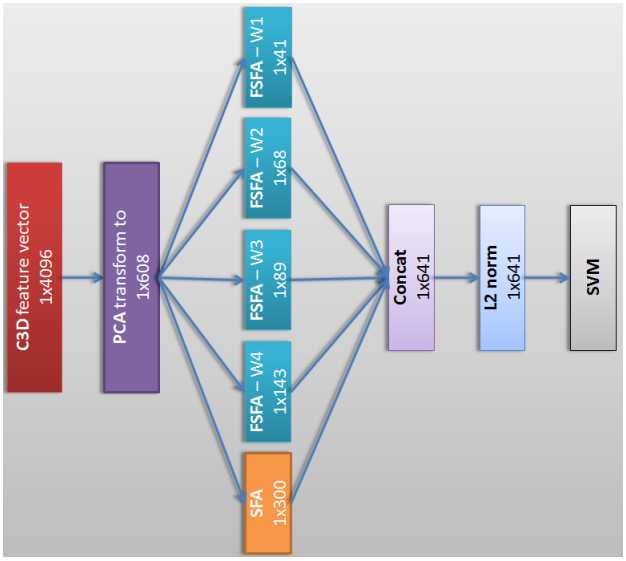


Figure 14: The architecture of the proposed neural network composed of FSFA & SFA nodes.

across the rows of $\mathbf{x}$. We then use the following normalized metric of slowness, in order to set a limit on the number of slow features that we keep from each node:

$$\Delta_n = \frac{\sum_{k=0}^{N-1} |\mathbf{Y}[k]|^2 \mathbf{W}[k]}{\sum_{k=0}^{N-1} |\mathbf{Y}[k]|^2} \tag{39}$$

Note that $0 \leq \Delta_n \leq 1$ and that the denominator of (39) is constant for all the extracted features in the training set, since they all fulfill the unit variance constraint. We set a threshold on Δ_n, $\Delta_n \geq 0.5$. (the same for all FSFA nodes) and each node keeps the learned functions that lead to the slowest features fulfilling this requirement.

All FSFA nodes operate on the same input $\mathbf{x}$ (i.e. the proposed network is not deep). We additionally employ a single SFA node that operates on $\mathbf{x}$ as well, and we keep the learned functions for its 300 slowest features (as we did in subsection 6.3). The expansion space for the SFA node is set to "Identity", similar to FSFA nodes. To complete the training process, we concatenate the features extracted by each one of the network's nodes, perform L2 normalization and train a linear SVM for classification. During testing, the C3D features

undergo the same SFA/FSFA transformations that were learned by the network during training, they are concatenated, L2 normalized and being fed to the SVM for final classification. The final feature vector size, at the network's output, is 641.

The architecture of the network is depicted in Figure 14, while the classification results (training and test accuracies) are shown in Table 6. We also provide classification results for when using only the FSFA nodes without the SFA node. (The classification performance of the SFA node alone is the same as it was reported in subsection 6.3). The implementation and parameterization of the linear SVM is kept the same as in section 6.3. In this experiment, we see that FSFA can harmonize with SFA in order to transform the features extracted by C3D to a new feature space where classification performance is further increased, compared to the single node operation that was described in subsection 6.3.

7. Conclusion

In the algorithm of Slow Feature Analysis (SFA) [4] signal slowness is measured by its average squared time-derivative. In this paper, we study the notion of slowness in the frequency domain, introducing Frequency-Based Slow Feature Analysis (FSFA). In the proposed method, the slowness objective takes a parametric filter representation, making slowness a filter dependent measurement. We find out, that the aforementioned new notion of slowness can be equivalent to the one used in standard SFA for a specific parameter value. This makes FSFA's slowness criterion a generalization of the one used in standard SFA. The new parametric optimization problem that is introduced, is directly given in the frequency domain and its closed form solution is derived.

Synthetic and real-world experiments show that, the features extracted by FSFA depend on the chosen filter. Moreover, a real world video action recognition experiment shows that, for proper filter choices, the features extracted by FSFA can lead to improved classification performance compared to the fea-

tures extracted by SFA, while there are other filter choices that perform worse, justifying the generalized nature of the propose method. Quantitatively, in our experiments, the classification performance of FSFA compared to SFA, in the real world video action recognition experiment, was ranging from -0.21% to +0.48%. Additionally, we showcase that a simple, shallow, neural network that is composed of four FSFA nodes, employing different filters, and one SFA node, can transform the features extracted by a CNN, to a new feature space, where classification performance, in the task of video action recognition, can be improved up to +2.9%. Apart from the concrete examples on the performance of FSFA that we provide in this paper, we believe that FSFA, also combined with standard SFA, can be used as a post-processing step in the fashion presented in [21] and that was followed in this paper, to potentially improve the classification performance of any modern feature extractor. Finally, potential future work could expand on eliminating the condition (28) under which FSFA equals SFA and study FSFA's optimal free responses as it was done for SFA in [40].

Acknowledgements

The research leading to these results has been supported by the EU funded project FORENSOR (GA 653355).

References

[1] M. Franzius, N. Wilbert, L. Wiskott, Invariant object recognition with slow feature analysis, in: Artificial Neural Networks - ICANN 2008: 18th International Conference, Prague, Czech Republic, September 3-6, 2008, Proceedings, Part I, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 961–970. doi:10.1007/978-3-540-87536-9_98.
URL http://dx.doi.org/10.1007/978-3-540-87536-9_98

[2] P. Berkes, Temporal slowness as an unsupervised learning principle - self-organization of complex-cell receptive fields and application to pattern recognition, Ph.D. thesis (2006).

[3] H. Sprekeler, C. Michaelis, L. Wiskott, Slowness: An Objective for Spike-Timing-Dependent Plasticity?, PLoS Computational Biology 3 (6) (2007) e112.

[4] L. Wiskott, T. Sejnowski, Slow feature analysis: Unsupervised learning of invariances, Neural Computation 14 (4) (2002) 715–770. doi:10.1162/089976602317318938.
URL http://dx.doi.org/10.1162/089976602317318938

[5] A. Escalante-B, L. Wiskott, Heuristic evaluation of expansions for non-linear hierarchical slow feature analysis, in: Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, Vol. 1, 2011, pp. 133 – 138.

[6] L. Wiskott, Estimating driving forces of nonstationary time series with slow feature analysis, arXiv.org e-Print archive, http://arxiv.org/abs/cond-mat/0312317/ (Dec 2003).

[7] W. Konen, P. Koch, The slowness principle; SFA can detect different slow components in non stationary time series, International Journal of Innovative Computing and Applications 3 (1) (2011) 3–10. doi:10.1504/IJICA.2011.037946.
URL http://dx.doi.org/10.1504/IJICA.2011.037946

[8] A. Escalante-B, L. Wiskott, Gender and age estimation from synthetic face images, in: Computational Intelligence for Knowledge-Based Systems Design: 13th International Conference on Information Processing and Management of Uncertainty, IPMU 2010, Dortmund, Germany, June 28 - July 2, 2010. Proceedings, Springer Berlin Heidelberg, 2010, pp. 240–249. doi:10.1007/978-3-642-14049-5_25.
URL http://dx.doi.org/10.1007/978-3-642-14049-5_25

[9] Z. Zhang, D. Tao, Slow feature analysis for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (3) (2012)

436–450. doi:10.1109/TPAMI.2011.157.
URL http://dx.doi.org/10.1109/TPAMI.2011.157

[10] K. Wang, Z. Zhang, L. Wang, Violence video detection by discriminative slow feature analysis, in: Pattern Recognition: Chinese Conference, CCPR 2012, Beijing, China, September 24-26, 2012. Proceedings, Springer Berlin Heidelberg, 2012, pp. 137–144. doi:10.1007/978-3-642-33506-8_18.
URL http://dx.doi.org/10.1007/978-3-642-33506-8_18

[11] C. Wu, L. Zhang, B. Du, Hyperspectral anomaly change detection with slow feature analysis, Neurocomputing 151, Part 1 (2015) 175 – 187. doi:http://dx.doi.org/10.1016/j.neucom.2014.09.058.
URL http://www.sciencedirect.com/science/article/pii/S0925231214012740

[12] P. Koch, W. Konen, K. Hein, Gesture recognition on few training data using slow feature analysis and parametric bootstrap, in: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, 2010, pp. 1–8.

[13] L. Sun, K. Jia, T. Chan, Y. Fang, G. Wang, S. Yan, DL-SFA: deeply-learned slow feature analysis for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, pp. 2625–2632. doi:10.1109/CVPR.2014.336.
URL http://dx.doi.org/10.1109/CVPR.2014.336

[14] Y. Shan, Z. Zhang, K. Huang, Learning skeleton stream patterns with slow feature analysis for action recognition, in: Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III, Springer International Publishing, Cham, 2015, pp. 111–121. doi:10.1007/978-3-319-16199-0_8.
URL http://dx.doi.org/10.1007/978-3-319-16199-0_8

[15] C. Wu, B. Du, L. Zhang, Slow feature analysis for change detection in multispectral imagery, IEEE Transactions on Geoscience and Remote Sensing 52 (5) (2014) 2858–2874. doi:10.1109/TGRS.2013.2266673.

[16] A. Escalante-B, L. Wiskott, Slow feature analysis: Perspectives for technical applications of a versatile learning algorithm, KI - Künstliche Intelligenz 26 (4) (2012) 341–348. doi:10.1007/s13218-012-0190-7.
URL http://dx.doi.org/10.1007/s13218-012-0190-7

[17] T. Blaschke, P. Berkes, L. Wiskott, What is the relation between slow feature analysis and independent component analysis?, Neural Computation 18 (10) (2006) 2495–2508. doi:10.1162/neco.2006.18.10.2495.
URL http://dx.doi.org/10.1162/neco.2006.18.10.2495

[18] H. Sprekeler, On the relation of slow feature analysis and laplacian eigenmaps., Neural Computation 23 (12) (2011) 3287–3302.

[19] R. Turner, M. Sahani, A maximum-likelihood interpretation for slow feature analysis, Neural Computation 19 (4) (2007) 1022–1038. doi:10.1162/neco.2007.19.4.1022.
URL http://dx.doi.org/10.1162/neco.2007.19.4.1022

[20] T. Blaschke, L. Wiskott, Independent slow feature analysis and nonlinear blind source separation, in: Independent Component Analysis and Blind Signal Separation: Fifth International Conference, ICA 2004, Granada, Spain, September 22-24, 2004. Proceedings, Springer Berlin Heidelberg, 2004, pp. 742–749. doi:10.1007/978-3-540-30110-3_94.
URL http://dx.doi.org/10.1007/978-3-540-30110-3_94

[21] S. Klampfl, W. Maass, Replacing supervised classification learning by slow feature analysis in spiking neural networks, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2009, pp. 988–996.
URL http://papers.nips.cc/paper/3672-replacing-supervised-classification-learning-by-sl
pdf

[22] V. Kompella, M. Luciw, J. Schmidhuber, Incremental slow feature analysis: Adaptive low-complexity slow feature updating from high-dimensional

input streams, Neural Computation 24 (11) (2012) 2994–3024. doi:10.1162/NECO_a_00344.
URL http://dx.doi.org/10.1162/NECO_a_00344

[23] S. Liwicki, S. Zafeiriou, M. Pantic, Incremental slow feature analysis with indefinite kernel for online temporal video segmentation, in: Computer Vision – ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part II, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 162–176. doi:10.1007/978-3-642-37444-9_13.
URL http://dx.doi.org/10.1007/978-3-642-37444-9_13

[24] S. Liwicki, S. Zafeiriou, M. Pantic, Online kernel slow feature analysis for temporal video segmentation and tracking, IEEE Transactions on Image Processing 24 (10) (2015) 2955–2970. doi:10.1109/TIP.2015.2428052.
URL http://dx.doi.org/10.1109/TIP.2015.2428052

[25] P. Berkes, Pattern recognition with slow feature analysis, Cognitive Sciences EPrint Archive (CogPrints) 4104.

[26] A. Escalante-B, L. Wiskott, How to solve classification and regression problems on high-dimensional data with a supervised extension of slow feature analysis, Journal of Machine Learning Research 14 (2013) 3683–3719.
URL http://jmlr.org/papers/v14/escalante13a.html

[27] A. Escalante-B, L. Wiskott, Improved graph-based SFA: information preservation complements the slowness principle, Computing Research Repository (CoRR).
URL http://arxiv.org/abs/1601.03945

[28] W. Böhmer, S. Grünewälder, H. Nickisch, K. Obermayer, Regularized sparse kernel slow feature analysis, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I, Springer

Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 235–248. doi:10.1007/978-3-642-23780-5_25.
URL http://dx.doi.org/10.1007/978-3-642-23780-5_25

[29] L. Zhang, C. Wu, B. Du, Automatic radiometric normalization for multitemporal remote sensing imagery with iterative slow feature analysis, IEEE Transactions on Geoscience and Remote Sensing 52 (10) (2014) 6141–6155. doi:10.1109/TGRS.2013.2295263.

[30] L. Zafeiriou, M. Nicolaou, S. Zafeiriou, S. Nikitidis, M. Pantic, Probabilistic slow features for behavior analysis, IEEE Transactions on Neural Networks and Learning Systems 27 (5) (2016) 1034–1048. doi:10.1109/TNNLS.2015.2435653.
URL http://dx.doi.org/10.1109/TNNLS.2015.2435653

[31] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 1735–1742. doi:10.1109/CVPR.2006.100.
URL http://dx.doi.org/10.1109/CVPR.2006.100

[32] H. Mobahi, R. Collobert, J. Weston, Deep learning from temporal coherence in video, in: L. Bottou, M. Littman (Eds.), Proceedings of the 26th International Conference on Machine Learning, Omnipress, Montreal, 2009, pp. 737–744.

[33] X. Wang, A. Gupta, Unsupervised learning of visual representations using videos, 2015 IEEE International Conference on Computer Vision (ICCV) (2015) 2794–2802.

[34] D. Jayaraman, K. Grauman, Slow and steady feature analysis: Higher order temporal coherence in video, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 3852–3861.

[35] D. Lay, Linear Algebra and Its Applications, 4th Ed, Addison-Wesley.

[36] K. B. Petersen, M. S. Pedersen, The matrix cookbook (Nov 2012).
URL http://www2.imm.dtu.dk/pubdb/p.php?3274

[37] F. Harris, On the use of windows for harmonic analysis with the discrete fourier transform, Proceedings of the IEEE 66 (1) (1978) 51–83.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15, IEEE Computer Society, Washington, DC, USA, 2015, pp. 4489–4497. doi:10.1109/ICCV.2015.510.
URL http://dx.doi.org/10.1109/ICCV.2015.510

[39] K. Soomr, A. R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, Computing Research Repository (CoRR) abs/1212.0402.
URL http://arxiv.org/abs/1212.0402

[40] L. Wiskott, Slow feature analysis: A theoretical analysis of optimal free responses, Neural Computation 15 (9) (2003) 2147–2177. arXiv:https://doi.org/10.1162/089976603322297331, doi:10.1162/089976603322297331.
URL https://doi.org/10.1162/089976603322297331