Using Mutual Independence of Slow Features for Improved Information Extraction and Better Hand-Pose Classification

Aditya Tewari*[†]
Aditya.Tewari@dfki.de

Bertram Taetz*
Bertram.Taetz@dfki.de

Didier Stricker*
Didier.Stricker@dfki.de

Frédéric Grandidier[†]
Frederic.Grandidier@iee.lu

ABSTRACT

We propose a Slow Feature Analysis (SFA) based classification of hand-poses and demonstrate that the property of mutual independence of the slow feature functions improves the classification performance. SFA extracts functions that describe trends in a time series data and is capable of isolating noise from information while conserving high-frequency components of the data which are consistently present over time or in the set of data points. SFA is a useful knowledge extraction method that can be modified to identify functions which are well suited for distinguishing classes. We show that by using the orthogonality property of SFA our information about classes can be increased. This is demonstrated by classification results on the well known MNIST dataset for hand written digit detection.

Furthermore, we use a hand-pose dataset with five possible classes to show the performance of SFA. It consistently achieves a detection rate of over 96% for each class. We compare the classification results on shape descriptive physical features, on the Principal Component Analysis (PCA) and the non-linear dimensionality reduction (NLDR) for manifold learning. We show that a simple variance based decision algorithm for SFA gives higher recognition rates than K-Nearest Neighbour (KNN), on physical features, PCA and non-linear low dimensional representation. Finally, we examine Convolutional Neural Networks (CNN) in relation with SFA.

Keywords

Slow Feature Analysis, Hand-Pose Identification, Knowledge extraction, Feature Learning

1 INTRODUCTION AND BACKGROUND

The hand is probably the most effective tool for indicating and gesticulating. Estimating the hand-pose in frames of a sequence to detect a gesture is a common step used in various gesture recognition approaches. Hand gesture recognition is steadily gaining popularity in tasks like navigation, selection and manipulation in Human Computer Interactions [BVBC04]. While complex applications like surgical simulation and training systems require dynamic hand gesture recognition [LTCK03], simpler command and control interfaces often employ hand-poses.

The hand-pose at each frame is treated as a feature in some approaches [CGP07], while some methods use this information to describe the states of a state machine [GMR$^+$02]. A sensor free, vision based detection of pose is a challenging task because of the large degree of freedom in the movement of hand parts and self occlusion that might occur, moreover the calculation of local edge or corner based features is prone to noise [CGP08]. Some methods use physical features of the hand like the gravitational center of the palm region and the finger location [RYZ11]. Other features include convexity that describes the curvature of the palm hull. The works of [PKK09, CLEL12] describe the use of geometrical descriptors for posture detection. We argue that because of occlusion and the high degree of freedom, high level features learnt from hand-pose data can help in improving the classification. In [LCP12] a method of manifold embedding for articulated hand configuration detection is proposed. This method learns one of the global description of data by identifying the manifold on which the data resides.

The SFA allows unsupervised learning of invariant or slowly varying features. It can learn translation, scale and rotational invariances [WS02]. The SFA technique has been modified to achieve supervised learning to

* DFKI, Augmented Vision, Trippstadter Str.122, Germany, 67663 Kaiserslautern.

[†] IEE-SA, Weiergewan, 11-rue Edmond Reuter, Luxembourg, 5326 Contern.

achieve classification [Ber05]. It provides mutually orthogonal features thus the prominent features carry independent information about the data even though they remain invariant to size, rotation and translation. Another important property of the SFA is the guaranteed optimisation to the slowest changing function which allows for easy extension when learning a new class. We propose to learn several slow feature functions for each class to improve classification further. To achieve this we employ the property of mutual orthogonality of features learnt from a class. The mutual orthogonality of SFA features result in aggregation of information thus it increases the effective information that a classifier receives.

Section 2.1 describes the basic ideas behind SFA, further we discuss its use as a classifier in section 2.2. In section 2.3 we explain the use of orthogonality to increase information and describe its effect on the classification task on the MNIST dataset in section 3.1. Finally in section 3.2 we apply the technique on hand-pose classification.

We ascertain the applicability of SFA as an information extraction method, by demonstrating better classification rates as compared to the standard PCA and manifold learning methods. The manifolds representation of data compensates for non-linearities. The better performance of SFA over manifold learning proves its strong capability of identifying the consistent properties of signals in a dataset. Apart from the comparison with PCA and manifold learning methods we also make comparisons to classification performed by using shape descriptors and geometrical features calculated from the hand-pose images. We report a substantial improvement over the classification done with these features. The improvement over physical features indicates that SFA is capable of information extraction while the improved classification compared to manifold embedding establishes the ability to handle non-linearities in a dataset.

The broad contributions made through this work are:

- The applicability of SFA for hand-pose classification using data obtained from a time of flight camera.
- SFA classification based on several slow feature functions and not just the principal slow feature.
- A comparison of classification based on physical features and SFA features that indicates the superior information extraction capability of SFA.
- The demonstration of improved classification performance on the MNIST hand written digit dataset and the hand-pose dataset using a modified SFA.

2 SLOW FEATURE ANALYSIS

2.1 Slow Feature Analysis as a Learning Problem

Low level features are short duration features and are often misleading. High level features of the data carry information that extends beyond small neighbourhoods. SFA learns functions that represent such high level features. These high level representation can better explain the property of the data space. A feature that does not vary rapidly, yet has a slow consistent change promises to describe the behaviour of a function in better detail [Föl91]. The slow features thus provide a consistent trend in the data. The SFA is originally designed for detection of trends in temporal data [WS02]. It has been modified to provide consistent trends within elements belonging to a static dataset [Ber05]. We first discuss the SFA procedure for temporal data and shall later explain the modifications for classification in static datasets.

If a vectorial input $\mathbf{X}(\mathbf{t}) \in \mathbb{R}^d$ is a time series, one of the slow features is the function $\mathbf{g}(\cdot)$, such that $\mathbf{y}(\mathbf{t}) = \mathbf{g}(\mathbf{X}(\mathbf{t}))$, varies as slowly as possible while avoiding trivial responses.

The problem is formally described by [Wis03] as minimising the absolute differential

$$\Delta(y_j) := \langle \dot{y_j}^2 \rangle. \tag{1}$$

Here y_j is the j^{th} component of $\mathbf{y}(\mathbf{t})$ and $\dot{y_j}$ is the derivative of y_j with respect to time t and $\langle \cdot \rangle$ denotes average over time. The absolute differential is minimised under the following conditions:

$$\langle y_j \rangle = 0 \tag{2}$$

$$\langle y_j^2 \rangle = 1 \tag{3}$$

$$\langle y_i y_j \rangle = 0 \quad i \neq j. \tag{4}$$

While the minimisation selects invariant features, (3) forces some variance and removes the possibility of obsolete solutions like a constant function and (4) forces independence among the calculated slow features. These constraints are forced by sphering the data [LZ98].

Sphering of $\mathbf{X} \in \mathbb{R}^d$ means we transform $\mathbf{X}$ such that the covariance matrix of the transformed random variable $\mathbf{X}^*(\mathbf{t})$ is an identity matrix. $\mathbf{X} = (x_1, x_2 ..., x_n)$, represents a data matrix and $x_1, x_2, x_3, ..., x_n$ are n vectors belonging to it. If $(\mathbf{X} - \mu)$ and Σ are respectively the centered data matrix and the covariance matrix, then the sphered data is expressed as:

$$\mathbf{X}^*(\mathbf{t}) = B_n(\mathbf{X} - \mu), \quad with \quad B_n^T B_n = \Sigma^{-1}. \tag{5}$$

The sphered data $\mathbf{X}^*(\mathbf{t})$ is projected into a quadratic space, resulting in data $\mathbf{Z}$. The derivative $Z(t+1) - Z(t)$, is represented by $\dot{Z}$. Let $\mathbf{W}$ be the eigenvectors of the covariance matrix of the derivative matrix $\dot{\mathbf{Z}}$,

$$\langle \dot{\mathbf{Z}}\dot{\mathbf{Z}}^{\mathbf{T}} \rangle \mathbf{W} = \lambda_{\mathbf{W}}. \tag{6}$$

The eigenvectors corresponding to the smallest eigenvalues are the direction of the slowest change in differential of the data. These eigenvectors compose the slow feature functions. These functions are the weighted linear sums over the components of the expanded signal, where weights are the components of eigenvectors w,

$$g_j(x) = w_j{}^T.Z(t). \tag{7}$$

Where w_j is the j^{th} column of the matrix $\mathbf{W}$. The m smallest eigenvalues correspond to the m primary slow feature functions: $g_1, g_2, g_3 \cdots g_m$.

2.2 Slow Feature Analysis for Classification

The slow features describe intrinsic features of a long time series. It is the property of slow features to conserve variations over time, this property can be exploited for classification. The data for classification is not temporal and thus the absolute differential described in (1) is modified to perform a supervised classification. To perform a supervised classification, functions resulting in minimum inter-element difference within each class are identified. As in case of time series SFA, the conditions of zero mean, constant variance and linear independence are imposed. Once again these conditions are satisfied by sphering the data. Furthermore, the optimisation process tries to increase the variance outside a class, to identify the slow feature functions.

For the dataset $\mathbf{X}$, we define a matrix $\mathbf{Z}$, such that $\mathbf{Z}$ is the quadratic expansion of the sphered transform of $\mathbf{X}$. Accordingly, the differential term for a vector z^{el} belonging to the expanded dataset $\mathbf{Z}$ is represented as:

$$\nabla_{el} := \sum_{C=1}^{N} \sqrt{\sum_{n=1}^{N_C} (z_C^n - z^{el})^2}. \tag{8}$$

Thus average differential for the data $\mathbf{Z}$ can be rerepresented as:

$$\nabla := \langle \nabla_{el} \rangle. \tag{9}$$

Where, z^{el} is the vector corresponding to the element for which the differential is calculated. z_C^n is the n^{th} element of a class C, N is the number of classes and N_C is the number of datapoints in the class C. We now minimise the value of ∇. This minimisation condition returns functions that forces slow variance within classes. Each of the slow features correspond to one of the classes, to further improve the extracted feature functions, (9) is extended to maximise the variance between classes while minimising it within the class [ZT12].

To achieve this we subtract the average of the absolute difference of the in-class element with elements outside the class (∇_{el}^o) from the average differential within the class (∇_{el}), that yields

$$\nabla_{el}^o \quad := \quad \sum_{C=1}^{N} \sqrt{\sum_{\{c=1,c\neq C\}}^{N} \sum_{n=1}^{N_c} (z_c^n - z^{el})^2}. \tag{10}$$

The calculation of the slow feature function is modified to minimising the cost function O, where O is defined as:

$$O = \langle \nabla_{el} \rangle - \langle \nabla_{el}^o \rangle. \tag{11}$$

2.3 Using Orthogonality to Increase Information

The classification process described above returns (N=number of classes) functions. These functions are learnt from the entire dataset using the optimisation function of (11). This procedure results in a set of functions which provide low variance response. The constraint of decorrelation between different slow features creates the possibility of learning many functions corresponding to one class.

The ready availability of features after doing an SFA procedure, and there mutual independence motivates us to find more features within a class. Thus we calculate multiple slow features corresponding to each class. Rather than learning slow features over the entire dataset we learn a set of function for every class. Slow features are learnt by restricting the dataset to elements of one class, this is repeated for all classes.

As each function is orthogonal, we have more than one function representing intrinsic properties of the specific class. These linear functions are decorrelated on the expanded space. Learning slow features in every class requires a larger training dataset, meanwhile it also results in adding information for classification. The optimisation function (11) is further modified to minimise variance within a class, while maximising out-of-class variation using all other classes (13). This modification extends (10) as follows:

$$\nabla_{el_C} := \sqrt{\sum_{n=1}^{N_C} (z_C^n - z^{el_C})^2}, \tag{12}$$

$$\nabla_{el_C}^o := \sqrt{\sum_{\{c=1,c\neq C\}}^{N} \sum_{n=1}^{N_c} (z_c^n - z^{el_C})^2}, \tag{13}$$

$\nabla_{el_C}^o$ in (13) is the sum of out-of-class variances calculated over the training dataset.

$$O_C = \langle \nabla_{el_C} \rangle - \langle \nabla^o_{el_C} \rangle. \tag{14}$$

el_C represents that the calculation for the differential is done for elements belonging to the class C. The optimisation for class C is achieved by minimising O_C.

The functions are collected as matrix $\mathbf{W_C}$ where C is the class for which these functions are learnt. $w_{\lambda_{C_j}}$ is the vector corresponding to the j^{th} eigenvalue λ_{C_j} of class matrix $\mathbf{W_C}$. For a test input vector $\mathbf{P}$ the functional $\mathbf{G}$ returns an output vector $\mathbf{G}(\mathbf{W},\mathbf{P})$. The functional $\mathbf{G}$ has m linear functions in the space corresponding to the dimension of vector expanded in data space,

$$G(\mathbf{W_C},\mathbf{P}) = \mathbf{P}\cdot\bar{\mathbf{W}}_{\mathbf{c}}^T. \tag{15}$$

The variance for the output of the function is calculated as,

$$Var_C \quad = \sum_j (\mathbf{P}\cdot \bar{w}_{\lambda_{C_j}})^2 \quad = \sum G(\mathbf{W_C},\mathbf{P})^2. \tag{16}$$

The final classification is performed as follows:

$$class = \underset{C}{\operatorname{argmin}}(Var_C). \tag{17}$$

While doing an N class classification using m functions for each class, we have Nm functions. Some of these functions are very similar even though they belong to separate classes. This does not affect the minimum variance choice, because of aggregation.

The value of functions corresponding to a class when applied to an element from the same class is centred around a constant value. When a function is applied on a mismatched class, the result is random. This randomness likely results in a wrong identification.

In the case of multiple centred functions, corresponding to a class, the resulting output for a matching sample has all the function outputs centred around zero. Some functions from non-matching classes may return centred responses close to zero but, the aggregated variance for a mismatch element is higher, resulting in clearer distinction from the matching class.

3 EXPERIMENTS

3.1 Effect of Increased Information on MNIST Dataset

MNIST dataset [LC12] is one of the most popular dataset for evaluating classification problems. The Lecun network [LJB+95] has achieved an error rate of less than 0.3% on the MNIST dataset. [Ber05] also describes the original classification technique on the MNIST Hand written digit dataset. We further tested and compared both methods of using SFA for classification described earlier on the same dataset. Each datapoint in the MNIST dataset is a 28x28 pixel image. We reduce it to a 35 dimensional vector by employing PCA and then project it into a quadratic space. The quadratic expansion of the 35 dimensional PCA vector results in a vector of size 630. We calculate 10 slow features functions for the full dataset. Also, we calculate 10 slow feature functions for each class. It was observed that the identification performance for every class improved when we used the property of orthogonality to calculate slow feature functions. The comparative results are listed in Table 1.

class	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Full Dataset	81	93	79	83	77	72	77	80	73	84
Class Separation	91	96	82	85	79	81	89	91	83	84

Table 1: Classification accuracy in % for each digit mentioned on the top row. The second row values are accuracy percentages when slow feature functions are learnt from the entire dataset, the third row shows the accuracy percentages when several functions are learnt independently for each class

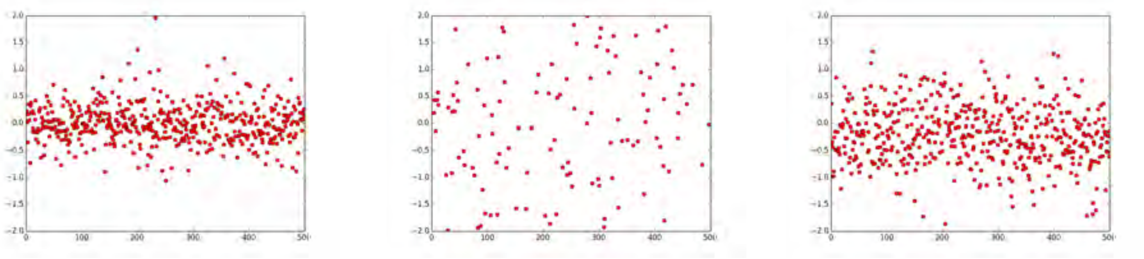
Figure 1 and Figure 2 show the difference between the two methods for classification. Figure 1 is based on identification of feature function from the entire data while Figure 2 is based on the classification approach where multiple corresponding functions are learnt from each class. The Y axis represents the distance of the response from the mean response calculated during the training stage, the X axis marks the index of input element on the dataset. The input elements are stacked in order of the classes that they belong to.

Figure 1 shows the centred response of the first three classes to the function corresponding to class with digit 0. The deviation of elements of class '0' from the origin are smaller as compared to other classes. This fits our hypothesis that SFA looks for feature functions that minimise the in-class variance. The Figure 2 shows the response of each data point to three functions learnt for class 0. The response of the data points of each class is shown in the same figure, with dark blue (the first cluster) representing class 0. The lower variance of function value to the matching class is clearly visible in these figures, the aggregation of function 1, 2 and 3 results in a deviation which is smaller for the matching class, but higher for mismatch. Averaging over these function values reduces the likely possibility of error in the first method because of randomness of non matching function response.

3.2 Hand-pose Experiments

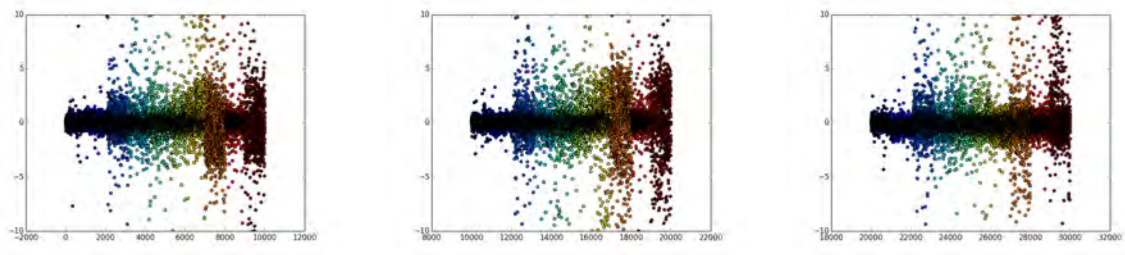
3.2.1 Hand-Pose Data Collection

A 3D Time-of-light, PMD-Nano camera has been used to collect a dataset of hand-poses. The camera is fixed vertically above the palm. The output of the PMD-Nano time of flight camera is an 120x165x2 image. The two channels of the image are the amplitude



(a) class:0 (b) class:1 (c) class:2

Figure 1: Response of class 0, 1 and 2 to function learnt from class 0.



(a) function:1 (b) function:2 (c) function:3

Figure 2: Response of all classes to the first 3 functions learnt from class 0.

value and the depth map image data. We cover the arm region with absorbent clothing and use the reflectance of skin to identify the palm. The reflectance constraint does not entirely remove the background and thus the closest contour greater than a threshold area is chosen as the palm region. The segmented palm region is then converted into a binary image which is further used for hand-pose identification.

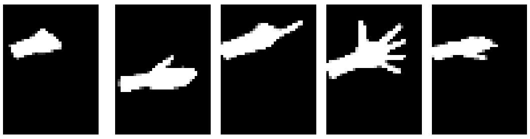
We then learn slow feature functions for five hand-pose classes labelled as "Fist", "Flat", "Index", "Open" and "Grab", see Figure 3. Slow features or invariances are learnt from a dataset of 3,000 frames of each class from 3 subjects. 1000 frames in each class are randomly selected and rotated in either direction, by an angle between 10° and 20°. These rotated frames are added to the training dataset along with the original frames. Note that, this spreads the poses such that they cover the whole rotational axis, it also increases the dataset and generates samples which train the SFA for rotational invariances.

Three hundred frames are selected for each class through random partitioning of the original dataset. These samples are used as test dataset, while the remaining original dataset is used for training. The preprocessing follows the same procedure as described for the training dataset.

3.2.2 Hand-pose Identification

Before learning slow features from the dataset of segmented hands, the image is scaled down to one-third of its original size. This is followed by a PCA which reduces each image to a 35 dimension vector that is projected to its quadratic space to allow the learning of non-linear invariances in the principal components of the training data.

During the SFA learning process the covariance matrix of the differential data as well as the eigenvectors corresponding to the largest eigenvalues are recorded. The eigenvectors corresponding to the ten largest eigenvalues correspond to the linear functions used for classification. Each function is centred around the mean values learnt during the training process. It is observed that the samples of matching classes are tightly spread around the mean values of the classes. The class which corresponds to the function has much smaller variance as compared to other classes. Figure 4 shows the response of the test dataset on the most prominent function of the "Fist" class. The data points for each class are represented by a unique color. The "Fist" class which is represented by blue in the figure has relatively tight packing of the data-points as compared to any other class. Like in the previous figures the X axis of the plot represents the data points which are arranged by their labels, and the Y axis represents the centered value of the learnt function.



(a) Fist (b) Flat (c) index (d) Open (e) Grab

Figure 3: The hand-pose samples.

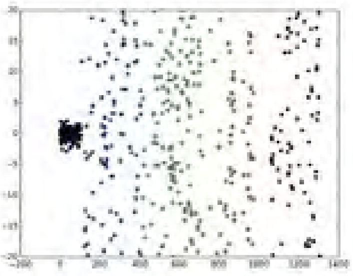


Figure 4: Slow function response for class 'Fist'.

This pattern is visible over the entire set of *Nm* functions, 5x10=50 in the present case. Table 2 shows the response of each data-point to the first five of the 10 learnt functions of each class. Each rows represents a set of functions corresponding to the class.

It can be observed that the functions learnt for one class have lower variance in the same class, while higher variance in other classes. This observation is used to differentiate classes. Thus we calculate the variance of the function response over all the functions calculated for a class.

The three hundred frames of each class in the dataset are used for evaluations. While learning models that are saved include, PCA mapping for each class, the sphering matrix, *m* eigenvectors and the covariance matrices for each class.

4 RESULTS AND COMPARISONS

We compare the results of the classification using slow feature analysis with results from KNN on physical features extracted from each frame. The physical features include coordinates of the tip of the finger (or the tip of the palm), the coordinates of the palm centroid, the convex ratio and the concave depth of the image and the polar and azimuth angle of the finger [RYZ11, PKK09, CLEL12] . We also compare the results to KNN applied on the PCA of the data and the low dimension manifold of the raw binary image [LCP12].

The KNN models for the physical features are generated using 1500 samples from each class and are modelled by simple euclidean distances. The Manifold is learned by Isomap algorithm [TDSL00] and the learning is done by the same training data as used for slow feature analysis.

Slow Feature Analysis based classification works better than the physical feature based classification evaluated in the KNN model. It also outperforms the KNN evaluation done with 35-dimensional (35-D) PCA and 9-dimensional (9-D) manifold representation of the dataset. We chose 35-D PCA because it is used as the basis for SFA calculation and 9-D isomap because the classification by KNN performs best for it. Table 3 shows the confusion matrix for the SFA based classification, Table 4 shows the confusion matrix for classification on KNN model trained on the hand crafted physical features. Table 5 is the confusion matrix for classification results from KNN model trained on the 35-D PCA representation of the image data. While classifying on the 9-D element vector received from the isomap done on the palm region as described earlier, the results are improved as compared to KNN on physical features and PCA based KNN. Table 6.

The results from the SFA are considerably better than the results from the physical features. These features are carefully selected for hand-pose estimation. This underlines the ability of the method to search for relevant features in a class. This improvement also suggests that SFA is capable of reducing the effect of local noise and distortion.

We compare SFA with KNN on the lower dimension representation of the data computed by PCA. The confusion matrices of Tables 3 and 5 clearly demonstrate that SFA performs far better. Thus the process of calculating the slow feature functions after doing PCA on the data further refines the knowledge that we are able to extract from the dataset.

SFA classification also performs better than a KNN model trained on manifold representation of the dataset. While the identification of the "Flat" hand-pose is better than the SFA in case of the isomap representation, the overall performance of SFA is superior. It is notable that KNN is a far more complex model as compared to simpler variance based classification of SFA. This result suggests that SFA is capable of managing nonlinearities in the data, this can be attributed to the step in which the PCA data is projected onto a quadratic space.

The improvement from PCA to isomap modelling is a result of better handling of non-linearities in the data. The KNN model based on euclidean distances suffers from the inability to compensate for non-linearities, this is overcome when we use the isomap projection. It is also important to note that while the KNN model is learnt over the isomap projection, SFA classification provides better results by simple variance calculations. It is worth mentioning that the performance improvement in the quality of classification was minimal when we scaled the palm region by distances. This observation can be attributed to the characteristic of SFA that, it explores multidimensional linear functions which encompasses the invariances over the data points.

Discussion

The SFA, as demonstrated in the last section, performs well for the classification task. Even though the total labelled data available to us was small, we compared the performance of SFA classification for hand-pose with CNN. The CNNs have resulted in exceptional classification results. As mentioned earlier Lecun network

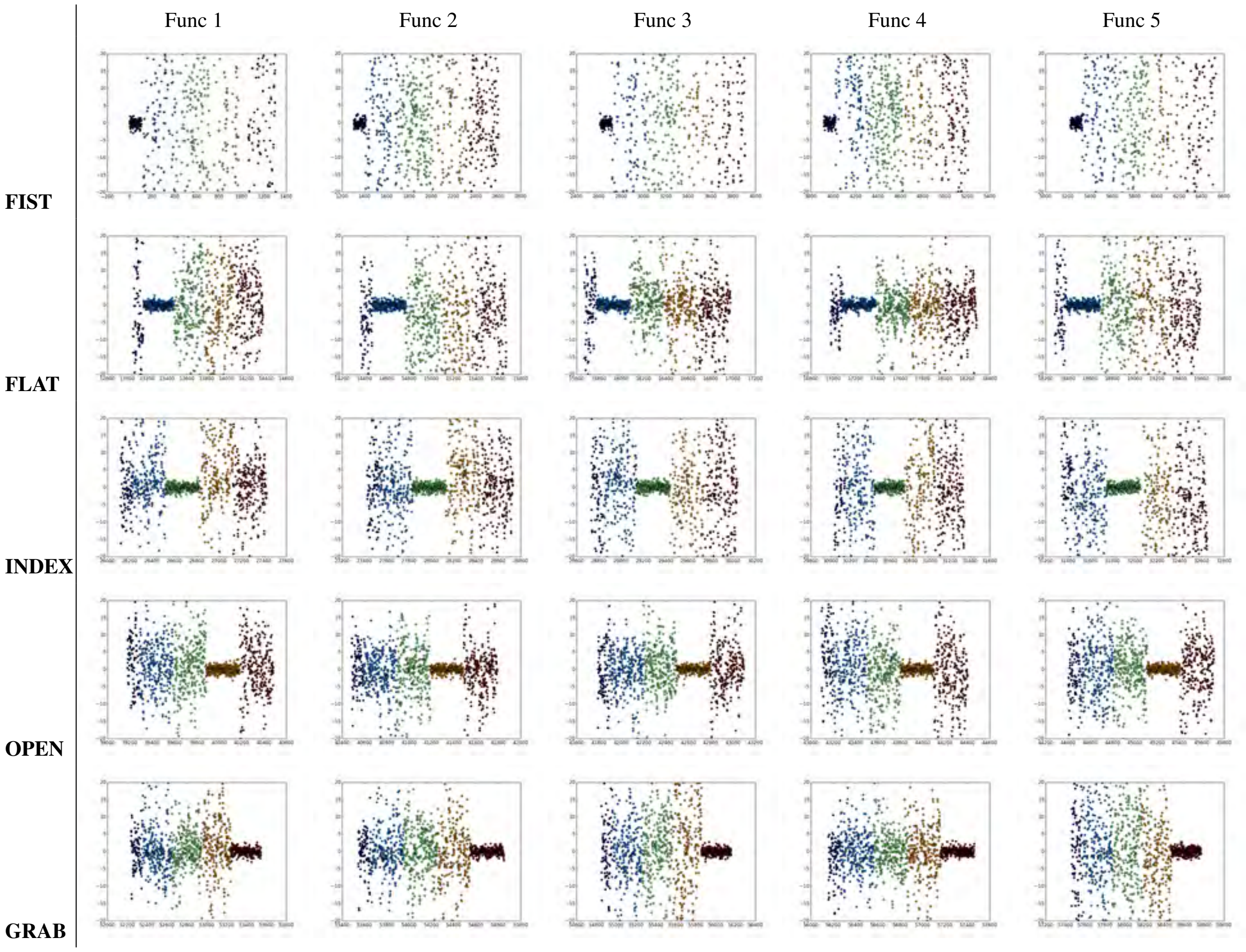


Table 2: Scatter map showing the value for 5 SFA functions for every class on the test dataset, different colors represent different classes.

based on CNN has achieved an error rate of less than 0.3% on the MNIST dataset, this compares favourably with human accuracy. We tested our hand-pose dataset for training a CNN with two convolution layers and two Max pooling layers. Using 15000 data-points after rescaling. The accuracy of classification reached over 98% after 30,000 iteration with a batch size of 50 images. Although, it was observed that because of the relatively small amount of data the CNN model starts over-fitting. The use of easily available, less specific hand-pose datasets for pre-training the CNN is one of the possible methods of overcoming the problem of over-fitting with the present data. SFA also requires a large dataset but lesser than CNN, we demonstrate that it is capable of learning functions for each class of hand-poses with 3000 data points. It can be argued that the SFA learning process results in learning of information that defines the class of the dataset, but the convolutional features learnt by a CNN using the classification based method contain information that distinguishes different classes. SFA results in lesser classification accuracy than CNN on a large dataset, but SFA gives interpretation about the nature of the class independent, which seems to be harder to identify in a CNN model.

5 CONCLUSION

In this paper we used SFA for classification on two datasets. SFA was tested on MNIST dataset and a hand-pose dataset. We approached the classification by training SFA separately for each class and demonstrated that, the property of orthogonality of SFA helps in extracting more information about the class. We showed that SFA outperforms hand picked physical features for hand-pose classification. This confirms the recent trend of preferring global features which are learnt from the data over extracting features by intuition. Training and test data has considerable variances of rotation and scale, in our experiments SFA remains robust to such variances.

The use of slow feature analysis also reduces the on line processing required on the test sample. SFA based classification requires a relatively large dataset for training, additionally it employs an expensive batch learning algorithm which requires large computer memory to run. Yet, it displays a remarkable ability to extract information and identify trends in a dataset. Usually calculating

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	1.0	0.0	1.7	0.3
FLAT	0.0	**96.7**	2.3	1.0	0.0
INDEX	0.0	0.0	**98.7**	1.3	0.0
OPEN	1.0	0.0	1.3	**97.6**	0.0
GRAB	0.7	2.3	0	0.3	**96.7**

Table 3: Confusion matrix for SFA classification. Bold values are accuracy values for the class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	0.7	1.3	0	1.0
FLAT	0.7	**95.7**	3.0	0	0.7
INDEX	2.7	5.7	**91.7**	0.3	0.0
OPEN	3.0	2.3	0	**94.3**	0.3
GRAB	0.7	4.7	0	0.3	**94.3**

Table 4: Confusion Matrix for KNN classification based on physical features. Bold values are accuracy values for class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**78.3**	12.2	2.9	3.8	2.9
FLAT	1.3	**80.7**	6.3	6.6	5.0
INDEX	0.0	3.3	**81.7**	2.0	14.0
OPEN	0.0	7.7	4.7	**85.3**	2.3
GRAB	0.3	3.3	7.7	3.0	**85.7**

Table 5: Confusion Matrix for KNN classification results on 35-D PCA. Bold values are accuracy values for the class corresponding to the respective row.

%	FIST	FLAT	INDEX	OPEN	GRAB
FIST	**97.0**	0.3	2.0	0.7	0
FLAT	0.3	**98.3**	1.3	0	0.3
INDEX	3.7	0.3	**96.0**	0	0
OPEN	1.7	0.0	1.0	**96.3**	1.0
GRAB	2.7	0.3	0.3	0.7	**96.0**

Table 6: Confusion Matrix for KNN classification on 9-D isomap on raw images. Bold values are accuracy values for the class corresponding to the respective row.

features at run time is a hard task, it consumes considerable computing and development effort. Whereas, SFA requires few linear operations to calculate the slow features. Thus, it does not only improve the robustness towards the data but also improves the performance of the machine when compared with processes that use physical features.

We showed the performance on global SFA features in this work and compared it to physical (local) features. Note that when we tested the SFA for classification of fixed length time series sequences, local features like peaks and inflexion, when combined with slow features, improved the classification performance. Classification was made using a logistic regression classifier. However, this fusion requires online feature calculation and a more complex classifier model.

It will be interesting to further study and quantify the effect of noise and poor segmentation on these features. Also further experiments with various data sources and the influence of an increasing number of classes on the orthogonality property of SFA will be of interest. We plan to extend the present approach of pose detection to gesture recognition. The batch learning approach is not suitable for the gesture classification and recently developed incremental SFA [KLS11] is a promising solution to the problem.

6 ACKNOWLEDGMENT

This work is supported by the National Research Fund, Luxembourg, under the AFR project 7019190.

7 REFERENCES

[Ber05] Pietro Berkes. Pattern recognition with slow feature analysis, February 2005.

[BVBC04] Volkert Buchmann, Stephen Violich, Mark Billinghurst, and Andy Cockburn. Fingartips: gesture based direct manipulation in augmented reality. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 212–221. ACM, 2004.

[CGP07] Qing Chen, Nicolas D Georganas, and Emil M Petriu. Real-time vision-based hand gesture recognition using haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–6. IEEE, 2007.

[CGP08] Qing Chen, Nicolas D Georganas, and Emil M Petriu. Hand gesture recognition using haar-like features and a stochastic context-free grammar. *Instrumentation and Measurement, IEEE Transactions on*, 57(8):1562–1571, 2008.

[CLEL12] J-F Collumeau, Rémy Leconge, Bruno Emile, and Hélène Laurent. Hand gesture recognition using a dedicated geometric descriptor. In *Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on*, pages 287–292. IEEE, 2012.

[Föl91] Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

[GMR$^+$02] Namita Gupta, Pooja Mittal, S Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Developing a gesture-based interface. *Journal of the Institution of Electronics and Telecommunication Engineers*, 48(3):237–244, 2002.

[KLS11] Varun Raj Kompella, Matthew D Luciw, and Jürgen Schmidhuber. Incremental slow feature analysis. In *IJCAI*, volume 11, pages 1354–1359, 2011.

[LC12] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits, 1998, 2012.

[LCP12] Chan-Su Lee, Sung Yong Chun, and Shin Won Park. Articulated hand configuration and rotation estimation using extended torus manifold embedding. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 441–444. IEEE, 2012.

[LJB$^+$95] Yann LeCun, LD Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, UA Muller, E Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276, 1995.

[LTCK03] Alan Liu, Frank Tendick, Kevin Cleary, and Christoph Kaufmann. A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599–614, 2003.

[LZ98] Guoying Li and Jian Zhang. Sphering and its properties. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 119–133, 1998.

[PKK09] Giorgio Panin, Sebastian Klose, and Alois Knoll. Real-time articulated hand detection and pose estimation. In *Advances in Visual Computing*, pages 1131–1140. Springer, 2009.

[RYZ11] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 1093–1096, New York, NY, USA, 2011. ACM.

[TDSL00] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Wis03] Laurenz Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

[WS02] Laurenz Wiskott and Terrence Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[ZT12] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):436–450, 2012.