Deep Analysis of Facial Behavioral Dynamics

Lazaros Zafeiriou[⋆∗] Stefanos Zafeiriou[⋆‡]
Maja Pantic[⋆†]
[⋆]Imperial College London, UK [†]University of Twente, The Netherlands
[∗]AimBrain, UK
[‡]Center for Machine Vision and Signal Analysis, University of Oulu, Finland
[⋆]{l.zafeiriou12, s.zafeiriou, m.pantic}@imperial.ac.uk,[∗]lazaros@aimbrain.com, [†]PanticM@cs.utwente.nl

Abstract

Modelling of facial dynamics, as well as recovering of latent dimensions that correspond to facial dynamics is of paramount importance for many tasks relevant to facial behaviour analysis. Currently, analysis of facial dynamics is performed by applying linear techniques, mainly, on sparse facial tracks. In this, paper we propose the first, to the best of our knowledge, methodology for extracting low-dimensional latent dimensions that correspond to facial dynamics (i.e., motion of facial parts). To this end we develop appropriate unsupervised and supervised deep auto-encoder architectures, which are able to extract features that correspond to the facial dynamics. We demonstrate the usefulness of the proposed approach in various facial behaviour datasets.

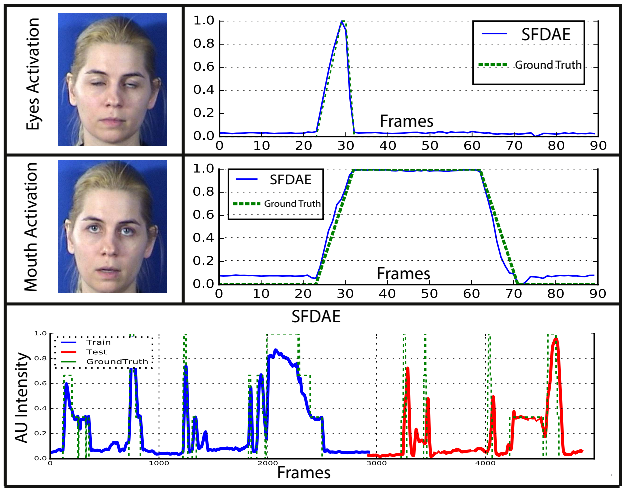

Figure 1: Some of the latent variables of the proposed unsupervised Deep Slow Feature Auto Encoder (SFDAE) which can be used in order to disentangle the dynamics of the facial expression (first two rows). Last row is the output features of the supervised SFDAE for predicting the AUs activations.

1. Introduction

One of the most important, yet understudied problems in automatic understanding of facial behaviour, is the automatic analysis of facial dynamics [35]. Facial dynamics are important, since they are required for precise detection of onsets, offsets, and the temporal envelope of facial emotional displays [35, 2, 25].

Automatic analysis of facial dynamics includes problems such as automatic discovering of latent dimensions that not only identify which parts of the face have moved in an expressive sequence but how they moved. That is, these dimensions can be used for (a) measuring how fast/slow a particular facial motion is, (b) segmenting the various temporal modes of facial motion (e.g., onset, offset, apex [25] etc.) and (c) for temporal alignment of behavioural sequences [36].

The current line of research on the unsupervised extraction of latent dimensions of facial dynamics revolves around linear deterministic and probabilistic methodologies [33, 32, 31, 13]. In particular, various general or special Linear Dynamical Systems (LDS) have been proposed for the task in [33, 13], as well as various extensions of deterministic Slow Feature Analysis (SFA) [31, 34]. In the recent past the supervised methodologies used to capture the temporal segments of facial behaviour mainly revolved around the use Hidden Markov Models (HMMs) [23, 18].

In this paper, we take a different direction and propose the first, to the best of our knowledge, non-linear deep methodologies for learning features that are able to describe the facial dynamics in behavioural sequences. In particular, we propose a novel deep unsupervised auto-encoder for the task. Then, in order to exploit the data labels that are available in some datasets, e.g. DISFA [12], we proposed a supervised version of the above auto-encoder that can be used in order to predict the dynamics of the motion of individual facial muscles, i.e., Facial Action Units (FAUs). Please see Fig. 1 for a motivation of the proposed approach. Summarising the contributions of the paper are:

- A novel slow-feature auto-encoder that can be used to extract the facial behavioural dynamics in an unsupervised manner.
- A novel slow-feature auto-encoder that can capitalise on the availability of labels (e.g., intensity of FAUs), which can be used for semi-supervised learning of behavioural dynamics. This auto-encoder can be used for predicting the dynamics of FAUs in image sequences.

2. Background

2.1. Slow Feature Analysis

Slow Feature Analysis (SFA) [29] is an unsupervised component analysis technique which is principle consist of identifying slowly moving/changing factors in temporal/spatial data. Specifically, given an D-dimensional temporal sequence (e.g., T vectorized video frames) $\mathbf{X} \in \mathbb{R}^{D\times T}$, SFA seeks for a low-rank projection matrix $\mathbf{V} \in \mathbb{R}^{D\times M}$ with $M \ll D$ that extracts slowly varying features from the rapid varying input sequence $\mathbf{X}$ by solving the following optimization problem:

$$\underset{\mathbf{V}}{\operatorname{argmin}} \quad \operatorname{tr}[\mathbf{V}^T\mathbf{A}\mathbf{V}], \text{ s.t. } \mathbf{V}^T\mathbf{B}\mathbf{V} = \mathbf{I}. \tag{1}$$

where $\mathbf{A}$ is the covariance matrix of the first-order temporal derivative of $\mathbf{X}$ and $\mathbf{B}$ is the data covariance matrix. That is,

$$\mathbf{A} = \frac{1}{T-1}\dot{\mathbf{X}}\dot{\mathbf{X}}^T = \frac{1}{T-1}\mathbf{X}\mathbf{L}\mathbf{X}^T, \quad \mathbf{B} = \frac{1}{T}\mathbf{X}\mathbf{X}^T, \tag{2}$$

where $\mathbf{L} = \mathbf{P}\mathbf{P}^T$ and $\mathbf{P}$ is an $T \times (T-1)$ matrix with elements $p_{i,i} = -1$ and $p_{i+1,i} = 1$. The solution of (1) is found by the Generalized Eigenvalue Problem $\mathbf{AV} = \mathbf{BV\Lambda}$, where the columns of the projection matrix $\mathbf{V}$ are the generalized eigenvectors associated with the M lowest eigenvalues contained in the diagonal matrix $\mathbf{\Lambda}$.

2.2. Auto Encoders

Auto-Encoder [5, 6] is a special neural network, that learns to copy its input to its output by using a series of affine transformations. Specifically, given a set of T frames stacked as the columns of a matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{D\times T}$, the autoencoder first maps the input $\mathbf{X}$ to a feature representation using an encoder function $\mathbf{Z} = f(\mathbf{X}) \in R^{M\times T}$ and then a decoder function $\mathbf{G} = g(\mathbf{Z}) \in \mathbb{R}^{D\times T}$ is used to produce the reconstruction, mapping each hidden representation $\mathbf{z}_t$ back to the input space. The most commonly employed encoder and decoder functions use deterministic affine mappings as follows

$$\mathbf{Z} = f(\mathbf{X}) = s_f(\mathbf{W}_f\mathbf{X} + \mathbf{b}_f) \tag{3}$$
$$\mathbf{G} = g(\mathbf{Z}) = s_g(\mathbf{W}_g\mathbf{Z} + \mathbf{b}_g) \tag{4}$$

where s_f and s_g are the encoder and decoder activation functions and can be non-linear such as Rectified Linear Units (ReLU), sigmoid, hyperbolic tangent and softmax or linear. It is worth mentioning, that in case of linear activation functions and one layer of encoding and decoding process the autoencoder is reduced to Principal Component Analysis (PCA). Having defined the appropriate activation function for the task at hand, the parameters $\theta = \{\mathbf{W}_f, \mathbf{W}_g, \mathbf{b}_f, \mathbf{b}_g\}$ are then optimised by minimizing the following loss function

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{X}, g(f(\mathbf{X}))) \tag{5}$$

which can be effectively solved by using stochastic gradient descent based backpropagation approaches.

Training an autoencoder to just copy its input to its output may sound useless, since it does not learn something useful about the data representation. In essence, autoencoders are trained in a way that will result in hidden representation $\mathbf{Z}$ obtaining specific properties as a byproduct with the input copying task. One common way to increase the expressive power of the these representations is to constrain them to have smaller dimension than the training data ($M < D$). These autoencoders are called *undercomplete* and even though they learn to extract the most salient features of the training data, they fail to learn something useful about the dataset if the autoencoder is provided with too much capacity.

To alleviate the aforementioned limitation a bunch of autoencoders has been proposed, known as *regularized* autoencoders, that use a cost function forcing the model to acquire specific properties in addition to the ability to learn its identity function. To this category belong the Sparse autoencoders [15, 9], the Contractive autoencoders [17, 16] and the Denoising autoencoders (DAE) [26, 27, 28]. Sparse autoencoders impose a $L1$ sparsity penalty on the hidden code $\mathbf{Z}$ in addition to the reconstruction error. The extracted features from these autoencoders are suitable for another task such as classification. In Contractive autoencoders an analytic contractive penalty is used, in order to enforce the derivatives of f to be as small as possible by minimising

$$\sum_t \mathcal{L}(\mathbf{x}_t, g(f(\mathbf{x}_t))) + \lambda||\frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t}||_F^2 \tag{6}$$

in order to make the representations as resistant as possible with respect to infinitesimal perturbations in input. Finally, the Denoising autoencoders receive as input a corrupted version of the data $\tilde{\mathbf{X}}$ contaminated by some form of noise through a corruption process $C(\tilde{\mathbf{X}}|\mathbf{X})$. Common choices include binomial noise (switching pixels or on off) or uncorrelated Gaussian noise. The autoencoder then is trained to undo this corruption by first finding the hidden representation $f(\tilde{\mathbf{X}}) = s_f(\mathbf{W}_f\tilde{\mathbf{X}}+\mathbf{b}_f)$ and then the reconstruction of the original input $\mathbf{G} = g(\mathbf{Z}) = s_g(\mathbf{W}_g\mathbf{Z}+\mathbf{b}_g)$.

Finally, the most recent and relevant to our proposed autoencoder is the Graph Regularized AutoEncoder (GAE) [10] which captures the geometrical structure of the data by incorporating a graph regularized constraint.

2.3. Convolutional Networks

CNNs are one of the most powerful and popular feature extractors over the last years, since they have shown exceptional performance in various computer vision problems such as image classification [8, 20, 21] and object detection [4, 3]. The architecture of a convolutional layer consists of three basic building blocks. In the first block, the layer performs several convolutions in parallel to produce a set of linear activations, in the second block (detector stage) these activations run through a nonlinear activation function (eg. Rectified Linear activation) and in the final block a further modification of the output is performed of via a pooling operation. Among the pooling operations the most popular one is the *max pooling* [19] and reports the maximum output within a rectangular neighborhood. Finally, the main difference between convolutional and fully connected layers is that the former introduce a parameter sharing property, where the weights are shared among all locations, maintaining the spatial locality. Therefore, convolution is much more efficient than dense matrix multiplication in terms of the memory requirements and statistical efficiency improving thus the performance of a machine learning system.

3. System Architectures

Here in we introduce our Slow Feature Denoising convolutional Auto-Encoder (SFDAE) and describe the proposed architectures for the task of temporal segmentation of facial expressions in both unsupervised and supervised manner.

3.1. Unsupervised Architecture

In Fig. 2 is depicted the detailed configuration for the entire unsupervised deep network consisting of four parts, convolution-fully connected-fully connected-deconvolution. The convolution part is responsible for extracting the features which are then encoded through the fully connected layers into low dimensional representations.

For the convolutional and deconvolutional layers we used the leaky ReLU[11] activation function, in order to alleviate the problems we experienced due to the hard 0 activation of the ReL units. The leaky ReLU activation function is given by

$$l(x) = \begin{cases} x & ,x \geq 0 \\ \frac{x}{a} & ,x < 0 \end{cases}$$

where a is a fixed parameter in range $(1, \infty)$. Although the authors in original paper suggest to set a to a large numbers (e.g 100), in our task we found that the optimal was $a = 10$. Finally, for the fully connected layers we used the sigmoid activation function as $\sigma(x) = 1/(1 + e^{-x})$.

The learning process proceeds by amplifying the input frames with a D-dimensional zero-mean Gaussian-distributed noise as follows

$$\tilde{\mathbf{X}} = \mathbf{X} + N(\mathbf{0}, \sigma^2\mathbf{I}) \tag{7}$$

We should point out that the purpose of the corruption process is not the denoising task, but instead denoising is investigated as a training criterion for extracting more robust and stable higher level representations. Then these representations are assessed by measuring how well can capture the temporal dynamics of the facial expressions without supervision. The equations that formulate the four parts of our auto encoder are given by

$$\begin{aligned}
\mathbf{H} = h(\tilde{\mathbf{X}}) = l_h(\mathbf{W}_h * \tilde{\mathbf{X}} + \mathbf{b}_h) \quad (8)\\
\mathbf{Z} = f(\mathbf{H}) = \sigma_f(\mathbf{W}_f\mathbf{H} + \mathbf{b}_f) \quad (9)\\
\mathbf{Q} = q(\mathbf{Z}) = \sigma_q(\mathbf{W}_q\mathbf{Z} + \mathbf{b}_q) \quad (10)\\
\mathbf{G} = g(\mathbf{Q}) = l_g(\mathbf{W}_g * \mathbf{Q} + \mathbf{b}_g) \quad (11)
\end{aligned}$$

where $*$ denotes the convolution operator. In addition to the denoising effect in order increase the expressive power of our representations we need to maintain the temporal coherence of the input frames. To this end, we introduce a slowly varying constraint that penalizes the temporal difference between successive hidden representations. Then based on (14) - (15) we optimize the SFDAE as follows

$$\sum_t \mathcal{L}(\mathbf{x_t}, g(q(f(h(\tilde{\mathbf{x}}_\mathbf{t}))))) + \lambda \sum_t ||f(h(\tilde{\mathbf{x}}_{t+1})) - f(h(\tilde{\mathbf{x}}_t))||_F^2 \tag{12}$$

where λ is a hyper-parameter that controls the strength of the slowness. The unsupervised cost function (12) can be written in a more compact form as follows

$$\underset{\theta}{\text{argmin}}(||\mathbf{X} - \mathbf{G}||^2 + \lambda\, \text{tr}(\mathbf{ZLZ}^T)) \tag{13}$$

where $\mathbf{L}$, described in section 2.1, is the matrix coding the slowly varying properties, $\text{tr}(.)$ denotes the trace of a matrix and the term $\text{tr}(\mathbf{ZLZ}^T)$ is the slow feature regularizer.

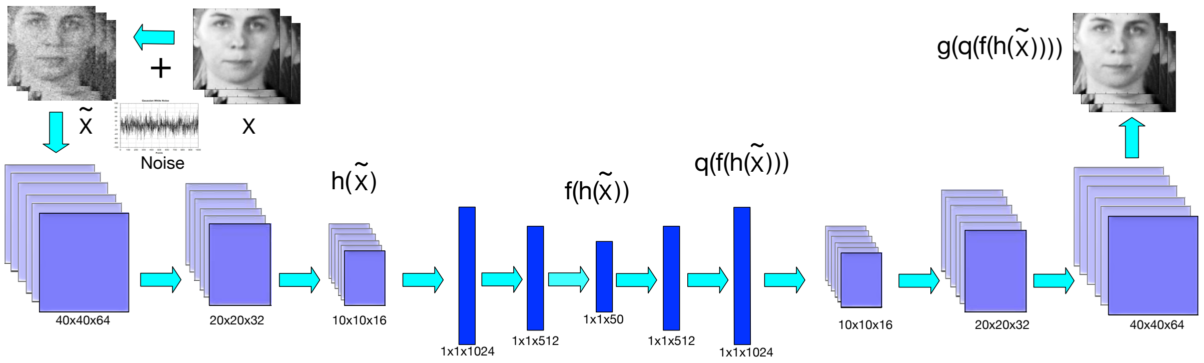

Figure 2: Overall architecture for the unsupervised discovery of latent representations. The learning process commences by corrupting the input which is then transformed into a low dimensional representation through the convolutional and fully connected layers. Finally, the clean input is reconstructed with a series of fully connected and deconvolutional layers.

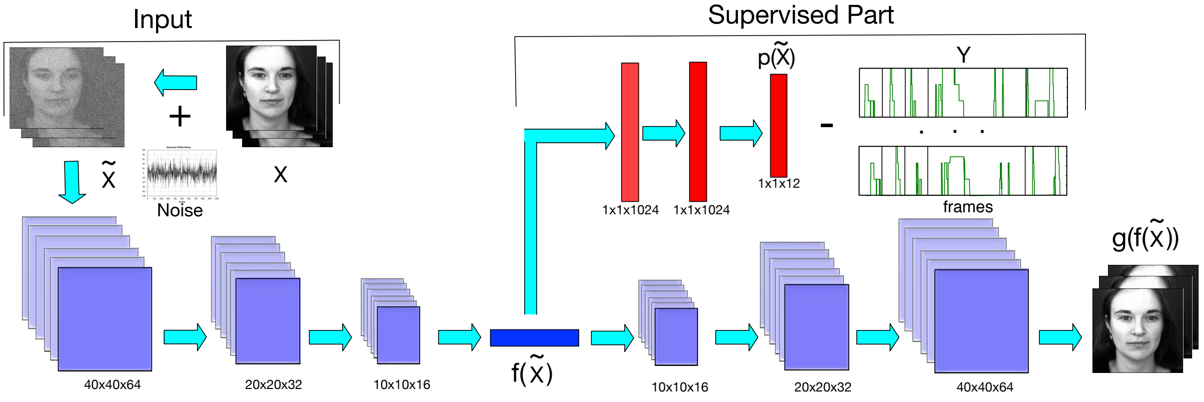

Figure 3: Overall architecture for the supervised discovery of latent representations. The learning process commences by corrupting the input which is then transformed into a hidden code $f(\tilde{\mathbf{X}})$ through the convolutional layers. Then the hidden code is transformed into a low dimensional representation that will represent the labels of the input data.

3.2. Supervised Architecture

In this section we describe the architecture that take into account the label information in cases where an abundance of labeled data is available. In order to incorporate the label information, we modify the architecture of Fig. 2 by replacing the fully connected parts with one flat vector. Then this flat vector is reshaped in order to be fed back to the decoder and is transformed into a low dimensional representation $\mathbf{P} = p(\tilde{\mathbf{X}})$ of size equal to the number of AUs that each frame is annotated, via two fully connected layers of the same number of hidden units. This architecture is illustrated in Fig. 3 and the equations of the encoder and the decoder are given by

$$\mathbf{Z} = f(\tilde{\mathbf{X}}) = l_f(\mathbf{W}_f * \tilde{\mathbf{X}} + \mathbf{b}_f) \tag{14}$$

$$\mathbf{G} = g(\mathbf{Z}) = l_g(\mathbf{W}_g * \mathbf{Z} + \mathbf{b}_g) \tag{15}$$

The aim of the supervised part is to force each one of the hidden units of the low dimensional representation to represent the evolution of the temporal events as accurately as possible by adding the following supervised loss

$$\underset{\theta}{\operatorname{argmin}}(1 - \frac{\mathbf{P}^T\mathbf{Y}}{||\mathbf{P}||||\mathbf{Y}||}) \tag{16}$$

More formally, given a set of frames $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{D\times T}$, with their corresponding labels $\mathbf{Y} = [\mathbf{y}_1, \ldots \mathbf{y}_T]\mathbb{R}^{C\times T}$, where C corresponds to the number of AUs that each frame is annotated, our supervised SF-DAE minimizes end to end the following loss function

$$\underset{\theta}{\operatorname{argmin}}(||\mathbf{X}-\mathbf{G}||^2 + \lambda\,\mathrm{tr}(\mathbf{ZLZ}^T) + 1 - \frac{\mathbf{P}^T\mathbf{Y}}{||\mathbf{P}||||\mathbf{Y}||}) \tag{17}$$

which is the complete loss function for the supervised architecture and can be solved by using gradient or stochastic based methods like (RMSprop[22] or Adam[7] optimizers). Finally, our SFDAE does not require weight-tying constraint ($\mathbf{W}_f = \mathbf{W}_g^T$), therefore we use distinct weights for encoder and decoder.

4. Experimental Results

In this section, we demonstrate the conducted experimental results in order to validate the performance of our proposed architectures in both unsupervised and supervised learning tasks.

4.1. Experiments in Unsupervised Learning

For assessing the performance of the proposed unsupervised architecture we employed the MMI [14, 24] and the UvA-Nemo Smile (UNS) [1] databases. The MMI consists of videos with deliberate FAUs while the UNS contains videos with deliberate and spontaneous smiles. Specifically, the MMI contains more than 2000 where 351 of them are annotated in terms of FAUs and the temporal segments in which a subject performs one or more FAUs in terms of neutral-onset-apex-offset-neutral indicators. We used all of the 351 videos and we tracked 68 facial landmarks using a variant of the Supervised Descent Method (SDM) [30]. The tracked landmarks were used in order to align and scale the frames to a fixed size template of 80×80 pixels. The relevant FAUs used for each region of the face are as follows:

- **Mouth**: Upper Lip Raiser, Nasolabial Deepener, Lip Corner Puller, Cheek Puffer, Dimpler, Lip Corner Depressor, Lower Lip Depressor, Chin Raiser, Lip Puckerer, Lip stretcher, Lip Funneler, Lip Tightener, Lip Pressor, Lips part, Jaw Drop, Mouth Stretch and Lip Suck

- **Eyes**: Upper Lid Raiser, Cheek Raiser, Lid Tightener, Nose Wrinkler, Eyes Closed, Blink, Wink, Eyes turn left and Eyes turn right

- **Brows**: Inner Brow Raiser, Outer Brow Raiser and Brow Lowerer.

UvA-Nemo Smile database contains more than 1000 smile videos (597 spontaneous and 643 deliberate) from 400 subjects. The database does not provide annotations with regards to temporal segments. Hence, we annotated 100 videos in total, 50 displaying deliberate and 50 displaying spontaneous smiles, in terms of temporal segments. Furthermore, we used the same algorithm to track 68 facial landmarks and align the facial images to a fixed size template of 80×80 pixels.

Implementation details. The network (Fig. 2) has in total 10 hidden layers, from which 6 of them are convolutional and 4 of them are fully connected. The 3 convolutional layers use 64, 32 and 16 number of filters, respectively, and are responsible for downsampling the corrupted frames to a size of 10x10x16 by using stride of 2, since we do not use the max pooling operator. On the other hand, the 3 deconvolutional layers are the mirrored version of the convolutional and are responsible for upsampling the images uncorrupted back to its original size of 80x80x1. Finally, for all the convolutional and deconvolutional layers we used the same filter size of 5x5. The first 2 fully connected layers receive as input the output of the convolutional layers as one flat vector and transform it to a low dimensional representation of size 1x1x50 by performing successive non-linear dimensionality reductions ($1024 \rightarrow 512 \rightarrow 50$). Then the other 2 fully connected layers transform this representation to one flat vector, which is then reshaped to a size of 10x10x16 in order to be used as an input to the deconvolutional network. In the case of graph regularized methods (GDAE and GAE) we have tested various approaches to build the graph Laplacian. That is, we used the heat kernel, dot-product kernels, 0-1 weighting etc. and we tested various neighbourhood sizes. The best was a 5-nearest neighbours graph to capture the local geometric structure of the data and a $0-1$ weighting system for defining the weight matrix. Furthermore, for all the regularized autoencoders (SFDAE, GDAE, SFAE and GAE) we set the hyper-parameter λ that regulates the contribution of the regularizer to 100 and for the denoising autoencoders (SFDAE, GDAE and DAE) we used gaussian noise of zero mean and 0.1 variance. Finally, we set the number of epochs to 500, the batch size to 10 and for learning the weights we employed stochastic optimisation with Adam[7] with the default hyperparameters, an initial learning rate of 0.001 with exponential decay of 0.95 every 1.000 iterations.

For the linear component analysis techniques, SFA and LPP, we considered projections to a subspace of dimensionality M = 50 for fair comparisons with the deep architectures. Finally, in order to evaluate to what extend the representations learned by different methods, were capable of accurately disentangling the label information of the AUs we used the cosine similarity as an evaluation metric. Specifically, in order to facilitate this comparison between the representations learned by the tested methods and the ground truth we map the recover latent space by each method to the temporal phases of AUs. To do so, we compute the first order derivative for each obtained hidden representation and select the one that minimizes: $\text{argmin}_i\, \mathbf{z}_i \mathbf{L} \mathbf{z}_i^T$ ($i = 1 \ldots 50$). For the SFA based methods (SFA, SFDAE and SFAE) we simply acquire the first identified latent feature, which corresponds to the most slowly varying one, since SFA introduces an ordering to the derived latent variables sorted by their temporal slowness. In total six variations of autoencoders were tested (SFDAE, GDAE, DAE, SFAE, GAE and AE) and compared with their linear counterparts (SFA and LPP).

Evaluation in MMI. We present the experiments in

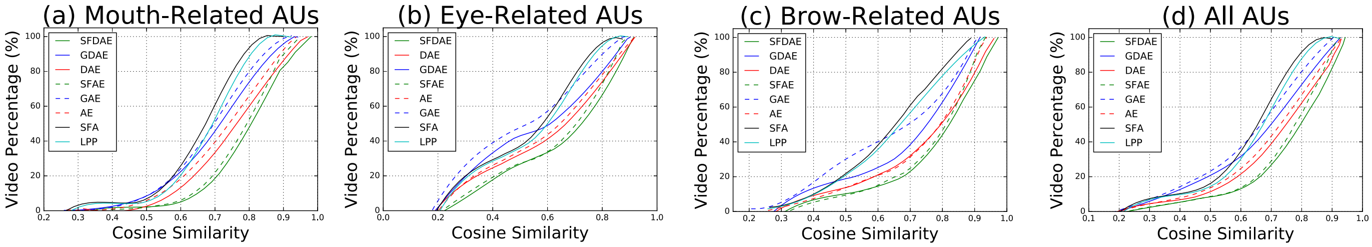

Figure 4: Overall results obtained when comparing various deep auto-encoders against linear component analysis techniques in terms of extracting the temporal dynamics in (a) Mouth-related AUs (b) Eyes-related AUs (c) Brows-related AUs (d) All the AUs in MMI database

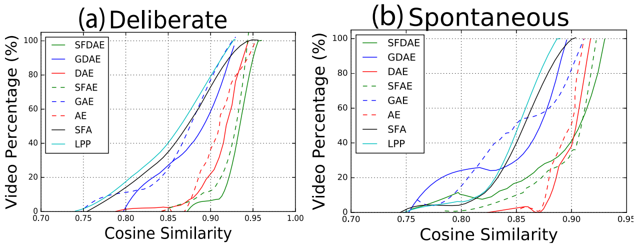

Figure 5: Overall results obtained when comparing various deep auto-encoders against linear component analysis techniques in terms of extracting the temporal dynamics in (a) Deliberate and (b) Spontaneous Smiles in UVS Smile database

MMI according to the region of the face the performed FAU is related to (i.e., mouth, eyes, and brows related-AUs respectively), as well as overall results from all the AUs. In the first set of experiments we measured the performance of the tested methods along the whole performed AU with respect to the percentage of the videos.[1] The evolution of the performance for each of the tested methods is plotted in Fig. 4.

From that figure it is immediately obvious that all the deep autoencoders greatly outperformed the linear ones in terms of extracting features that can better capture the dynamics of the AUs. The second inference of that figure regards the effectiveness of the corruption process. Specifically, as can be observed all the denoising autoencoders achieved better performance compared with their respective non-noisy autoencoders. For instance inspect the difference in the performance by comparing the results between the AE (red dashed line) and the DAE (red solid line) in Fig. 4(a). The final inference is that the proposed SFDAE (green solid line) outperforms all other methods in the task of unsupervised discovery of latent representations that can effectively disentangle the label information. The explanation of this is attributed to the fact that the slowly varying constraint in addition to the denoising effect provides smoothness to the representations and also maintains the temporal coherence of the input frames.

The second set of experiments, evaluates further the performance of the tested methods by providing the average cosine similarity for each temporal phase separately and the results are reported in Table 1. Specifically, this table summarizes the average cosine similarity for the onset, apex and offset temporal phases along with the overall performance for the whole performed AU for each region of the face separately.

Evaluation in UVS. A similar experimental setup was used in UNS database to evaluate the performance of the proposed methods (SFDAE and SFAE) against its competitors in deliberate and spontaneous smiles. Fig. 5 plots the cosine similarity curves versus the percentage of videos for both deliberate and spontaneous smiles, while Table 2 provides the average cosine similarity for each temporal phase separately. The experiments in UVS smile databse demonstrate once more the usefulness of the denoising process in the representations and that the proposed methods outperform the tested methods in both deliberate and spontaneous smiles. Specifically, overall the SFDAE achieved on average the better performance in deliberate smiles while SFAE achieved on average the better performance in spontaneous smiles (Table 2 last column).

4.2. Experiments in Supervised Learning

For evaluating the performance of our supervised model we employed the DISFA[12] database consisting of 27 videos that depict subjects to perform spontaneous facial activities. Every video contains 4845 frames annotated in terms of which of the following 12 actions units (AU1, AU2, AU4, AU5, AU6, AU9, AU12, AU15, AU17, AU20, AU25, AU26) is activated and in terms of its respective intensity level within a scale of 0 to 5, with 0 corresponding to the absence of an AU and 5 to the maximum intensity

[1] For example a point (60%, 0.8) indicates that 60% of the videos have cosine similarity less than 0.8 or 40% of the videos have cosine similarity greater than 0.8.

Cosine Similarity													
	Onset			Apex			Offset			Overall			
Method	Mouth	Eyes	Brows	Mouth	Eyes	Brows	Mouth	Eyes	Brows	Mouth	Eyes	Brows	All AUs
SFDAE	**0.9113**	0.9271	**0.9332**	0.9631	**0.9770**	0.9773	**0.8514**	**0.8372**	**0.8030**	**0.7888**	**0.6572**	**0.7697**	**0.7555**
GDAE	0.8623	0.8496	0.8351	**0.9743**	0.9650	**0.9880**	0.8050	0.7873	0.7354	0.7028	0.5616	0.6933	0.6602
DAE	0.8916	**0.9278**	0.9120	0.9415	0.9555	0.9696	0.8288	0.8287	0.7619	0.7542	0.6097	0.7387	0.7183
SFAE	0.8857	0.8989	0.9052	0.9342	0.9477	0.9479	0.8259	0.8121	0.7789	0.7744	0.6465	0.7586	0.7424
GAE	0.8364	0.8241	0.8101	0.9450	0.9360	0.9584	0.7809	0.7637	0.7133	0.6867	0.5029	0.6389	0.6323
AE	0.8648	0.9006	0.8846	0.9131	0.9268	0.9405	0.8040	0.8038	0.7637	0.7268	0.5960	0.7320	0.6966
SFA	0.8670	0.9003	0.8776	0.9531	0.9703	0.9611	0.8135	0.7935	0.7438	0.6608	0.5364	0.6339	0.6283
LPP	0.8851	0.9165	0.8910	0.9704	0.9710	0.9693	0.8235	0.8176	0.7517	0.6814	0.5503	0.6438	0.6461

Table 1: Average performance of various deep auto-encoders against linear component analysis techniques in terms of disentangling the label information from Actions Units related to mouth, eyes and brows, evaluated on all AU temporal phases in MMI database.

	Onset		Apex		Offset		Overall	
Method	Deliberate	Spontaneous	Deliberate	Spontaneous	Deliberate	Spontaneous	Deliberate	Spontaneous
SFDAE	**0.8753**	0.8588	**0.8992**	0.8874	**0.8394**	0.8202	**0.8308**	0.7806
GDAE	0.8176	0.8242	0.8899	0.8781	0.8317	0.8260	0.7776	0.7577
DAE	0.8721	**0.8721**	0.8895	0.8806	0.8187	0.8336	0.8065	0.7903
SFAE	0.8686	0.8552	0.8909	**0.8885**	0.8215	**0.8505**	0.8210	**0.7967**
GAE	0.8267	0.8668	0.8879	0.8860	0.8262	0.8193	0.7652	0.7851
AE	0.8721	0.8177	0.8795	0.8814	0.8181	0.8357	0.8060	0.7680
SFA	0.7787	0.7755	0.8847	0.8698	0.7374	0.7379	0.7631	0.7518
LPP	0.7753	0.7724	0.8791	0.8576	0.7379	0.7341	0.7534	0.7463

Table 2: Average performance of various deep auto-encoders against linear component analysis techniques in terms of disentangling the label information from deliberate and spontaneous smiles, evaluated on all AU temporal phases in UVS smile database.

level. Finally, this database provides 66 landmarks points where we used them in order to align and scale the frames to a fixed size template of 80x80 pixels.

Implementation details. The network used for this task (Fig. 3) has in total 8 layers, from which 6 of them are convolutional and 2 are fully connected. The 3 convolutional and the 3 deconvolutional layers use the same configuration described in sec. 4.1, while each one of the 2 fully connected have 1024 hidden units and use ReLU activations. These layers form the supervised part and are responsible for transforming the hidden code $f(\tilde{\mathbf{X}})$ into a low dimensional representation $\mathbf{P}$ of size 1x1x12, where 12 is the number of AUs that each frame is annotated to, in order minimize the cosine distance between this representation and the given labels $\mathbf{Y}$. Finally, in order to avoid overfitting we used dropout which was set to 0.5 during training and 1 during testing. The rest of the settings are the same with the ones described in the previous section.

Evaluation in Disfa. For the purpose of this experiment we used 60% of the videos for training and 40% for testing. Therefore in total we used 78.489 frames during the training phase and for 52.326 frames for testing. Similarly to the unsupervised task, in order to evaluate how accurately the learnt representations were able to predict the dynamics of the AUs we used the cosine similarity as evaluation metric. Finally for this experiment we evaluate the performance only of the deep auto encoders since SFA and LPP are unsupervised methodologies.

Table 3 reports the obtained results for each AU separately in both training and testing phases. The results show that in most of the AUs the proposed SFDAE outperform all others in both training and testing evaluation with an average score of 0.5792 for all the AUs, while the second best score was 0.5351 achieved by the DAE. In addition, as we can observe we did not experience overfitting during the learning process since the results indicate that our models learned to generalize quite well the supervised task.

Finally for better inspection of the performance of the tested methods we provide in Fig. 6 a qualitative experiment of disentangling the label information from a subject performing AU25 (Lips Part) in a video sequence of 4845 frames. Specifically, this figure plots the low dimensional representations for each of the tested methods and after training them to extract the label information (blue parts) we tested them by evaluating their ability to predict the rest of the AU activations (red parts). As can be observed the representation learnt by the proposed method (Fig. 6 (a)) were capable of better predicting the dynamics of the AU25 since it provides smoother and more accurately features which in turn matches better with the true label information (green curve).

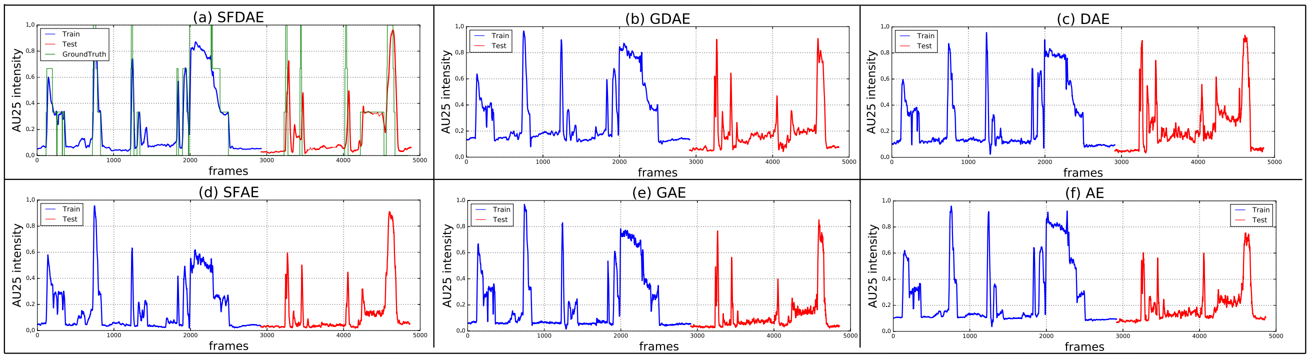

Figure 6: Disentangling the temporal changes from AU25 by applying the SFDAE, GDAE, DAE, SFAE, GAE and AE on a video sequence from the DISFA database. The blue curves correspond to the representations obtained during training while the red curves correspond to the predicted representations obtained during testing. The green curve indicate the annotated ground truth from the AU25.

	Cosine Similarity											
	SFDAE		GDAE		DAE		SFAE		GAE		AE	
Action Unit	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
AU1	**0.7320**	**0.5441**	0.5048	0.4705	0.7010	0.4996	0.7147	0.5076	0.7162	0.4837	0.6862	0.4888
AU2	0.6132	**0.4627**	0.4559	0.4409	0.5479	0.4142	**0.6419**	0.4192	0.6298	0.3977	0.5994	0.4182
AU4	**0.9034**	0.7191	0.6610	0.5797	0.8859	**0.7192**	0.8492	0.5736	0.8660	0.5710	0.8770	0.5190
AU5	0.4112	0.2903	0.2873	0.2525	**0.5122**	0.2946	0.4536	0.3122	0.4707	**0.3160**	0.5099	0.3024
AU6	**0.9144**	**0.5783**	0.7304	0.5003	0.9023	0.5139	0.8959	0.5302	0.8988	0.5389	0.8914	0.5346
AU9	0.5422	**0.4440**	0.4897	0.4053	0.5418	0.3878	0.6255	0.4223	0.6292	0.4251	**0.6476**	0.4073
AU12	0.9310	**0.6673**	0.8621	0.5099	0.9336	0.5772	0.9356	0.5263	0.9257	0.5371	**0.9347**	0.5224
AU15	**0.6716**	**0.4576**	0.5465	0.3813	0.6701	0.3591	0.6540	0.3922	0.6634	0.3993	0.6512	0.3775
AU17	0.8824	**0.6743**	0.6738	0.5117	**0.9096**	0.6377	0.7424	0.5687	0.7217	0.5206	0.7515	0.5173
AU20	**0.6113**	0.4158	0.5387	0.3894	0.5859	0.4051	0.5304	**0.4159**	0.5570	0.3821	0.5377	0.2744
AU25	0.9280	**0.9073**	0.8839	0.7790	0.9048	0.8772	0.9643	0.8531	**0.9661**	0.8482	0.9648	0.7947
AU26	**0.9064**	**0.7896**	0.7468	0.6061	0.9013	0.7367	0.8771	0.6779	0.8713	0.6258	0.8753	0.6167
Average	**0.7538**	**0.5792**	0.7150	0.4855	0.7497	0.5351	0.7403	0.5166	0.7429	0.5037	0.7440	0.4811

Table 3: Average performance of different supervised auto-encoders in terms of disentangling the label information in all 12 AUs in DISFA database.

5. Conclusions

In this paper, the SFDAE and SFAE have been proposed in order to learn robust and stable representations capable of describing the evolution of the temporal events in unsupervised and capable of predicting the label information in a supervised manner. We showed that by combing the slowly varying properties with the non-linear capabilities of denoising autoencoders we can extract abstract features which can represent faithfully the facial behavioural dynamics.

6. Acknowledgements

The work of Lazaros Zafeiriou and Maja Pantic was partially funded by EPSRC Project EP/N007743/1 (FACER2VM). Stefanos Zafeiriou was partially funded by the FiDiPro program of Tekes (project number: 1849/31/2015). This work was done while Lazaros Zafeiriou was working as Research Associate at Imperial College.

References

[1] H. Dibeklioglu, A. A. Salah, and T. Gevers. Uva-nemo smile database. http://www.uva-nemo.org/. 5

[2] H. Dibeklioglu, A. A. Salah, and T. Gevers. Are you really smiling at me? spontaneous versus posed enjoyment smiles. In *Proceedings of the 12th European conference on Computer Vision*, volume 3, pages 525–538, Firenze, Italy, Oct. 2012. 1

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 3

[4] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 3

[5] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 2

[6] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length, and helmholtz free energy. *Advances in neural information processing systems*, pages 3–3, 1994. 2

[7] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 5

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[9] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area v2. In *Advances in neural information processing systems*, pages 873–880, 2008. 2

[10] W. Y. Liao Y and L. Y. Graph regularised auto-encoders for image representation. *Proceedings of IEEE Transaction of Image Processing*, 2016. 3

[11] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013. 3

[12] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013. 1, 6

[13] M. A. Nicolaou, V. Pavlovic, and M. Pantic. Dynamic probabilistic cca for analysis of affective behavior and fusion of continuous annotations. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1299–1311, 2014. 1

[14] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *ICME*, pages 317–321, Amsterdam, The Netherlands, July 2005. 5

[15] C. Poultney, S. Chopra, Y. L. Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2006. 2

[16] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot. Higher order contractive auto-encoder. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 645–660. Springer, 2011. 2

[17] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 833–840, 2011. 2

[18] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert. Recognition of 3d facial expression dynamics. *Image and Vision Computing*, 30(10):762–773, 2012. 1

[19] D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, pages 92–101. Springer, 2010. 3

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 3

[22] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012. 4

[23] M. Valstar and M. Pantic. Fully automatic facial action unit detection and temporal analysis. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 149–149. IEEE, 2006. 1

[24] M. F. Valstar and M. Pantic. Mmi facial expression database. http://www.mmifacedb.com/. 5

[25] M. F. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(1):28–43, 2012. 1

[26] P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 2

[27] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 2

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010. 2

[29] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, April. 2002. 2

[30] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 532–539. IEEE, 2013. 5

[31] L. Zafeiriou, E. Antonakos, S. Zafeiriou, and M. Pantic. Joint unsupervised face alignment and behaviour analysis. In *European Conference on Computer Vision*, pages 167–183. Springer, 2014. 1

[32] L. Zafeiriou, M. A. Nicolaou, S. Zafeiriou, S. Nikitidis, and M. Pantic. Learning slow features for behaviour analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2847, 2013. 1

[33] L. Zafeiriou, M. A. Nicolaou, S. Zafeiriou, S. Nikitidis, and M. Pantic. Probabilistic slow features for behavior analysis. *IEEE transactions on neural networks and learning systems*, 27(5):1034–1048, 2016. 1

[34] L. Zafeiriou, S. Nikitidis, S. Zafeiriou, and M. Pantic. Slow features nonnegative matrix factorization for temporal data decomposition. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1430–1434. IEEE, 2014. 1

[35] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, Jan. 2009. 1

[36] F. Zhou and F. Torre. Canonical time warping for alignment of human behavior. In *Proceedings of Advances in neural information processing systems*, pages 2286–2294, Whistler, BC, Canada, Dec. 2009. 1