Exploring a mixed representation for encoding Temporal Coherence

DOI:
10.1007/978-3-319-46227-1_22

Document Version
Accepted author manuscript

Exploring a mixed representation for encoding Temporal Coherence

Jon Parkinson, Ubai Sandouk, and Ke Chen

School of Computer Science, University of Manchester
Oxford Road, Manchester, M13 9PL, UK
{jon.parkinson,ubai.sandouk,ke.chen}@manchester.ac.uk

Abstract. Guiding representation learning towards temporally stable features improves object identity encoding from video. Existing models have applied temporal coherence uniformly over all features based on the assumption that optimal object identity encoding only requires temporally stable components. We explore the effects of mixing temporally coherent invariant features alongside variable features in a single representation. Applying temporal coherence to different proportions of available features, we introduce a mixed representation autoencoder. Trained on several datasets, model outputs were passed to an object classification task to compare performance. Whilst the inclusion of temporal coherence improved object identity recognition in all cases, the majority of tests favoured a mixed representation.

Keywords: Representation Learning, Temporal Coherence, Object Recognition

1 Introduction

Real world objects likely to appear in video exhibit a natural permanence over time, a property known as temporal coherence. If an object is present in a given frame, whilst it might undergo small changes in position and pose, it is likely the same object will also appear in neighbouring frames [1]. By guiding representation learning from video towards the discovery of temporally coherent structure present in the raw image data, the capture of variance associated with the identity of individual objects is improved [2][3][4].

Existing models applying temporal coherence regularization in unsupervised representation learning have tended to apply the rule uniformly across all available features. This is based on the assumption that as the identity of objects remains temporally coherent, encoding this information in an abstract representation only requires the discovery of input structure exhibiting similar properties.

To investigate whether this approach improves object identity encoding, we explore the effects of discovering a mixture of temporally coherent and variable features, all contained in a single representation. Section 3 introduces a mixed autoencoder, based on a commonly used method for discovering the important variance underlying visual data, the sparse autoencoder [5]. Sparse autoencoders

have previously been adapted to encode temporal coherence [6], with the extra regularization enforced across every feature. We trained a range of models on three video datasets, applying temporal coherence regularisation over different sized subsets of the total available features. We describe our experimental process in Section 4.

Once trained, labelled examples were passed through each of the various models, providing the input to a classifier trained to recognise object identity. Temporally coherent representations have generally been used for encoding object identity, so comparing the performance boost each model provides a supervised classifier is an ideal test bed to evaluate performance. Our results are presented in Section 4.3.

The best classification accuracy from the supervised task came from a representation encoding temporal coherence in every test. Interestingly however, the majority of cases favoured our mixed representation over the all-invariant alternatives. A discussion of our results is provided in Section 5. This section is followed by our final conclusions.

2 Background

There are generally two methods for discovering temporally coherent structure from video. The first discovers variance changing as smoothly [7], also described as slowly [8], as possible over time. This approach is used in Slow Feature Analysis (SFA). The whole video dataset is analysed, with the slowest changing variance extracted on the assumption this encodes important properties of an input, whilst making sure the constant trivial solution is avoided. This approach has had considerable impact, producing behaviour similar to cells present in early areas of the visual cortex [9], and discovering features encoding object position and pose alongside identity [10]. SFA does suffer from drawbacks however. For complex problems, a non-linear expansion of an input is required, the details of which are not known in advance. The two instances of SFA on possibly expanded data can also render the algorithm computationally prohibitive.

The alternative approach manipulates the likelihood that an object present in a single frame is likely to appear in neighbouring frames. Temporally coherent structure is discovered by constraining network activations for each pair of neighbouring frames to be as similar as possible on the assumption of their likely semantic similarity. This method has improved object identity performance on the benchmark COIL100 dataset [11], applying the temporal coherence to the output of a Convolutional Neural Network [12], alongside sparsity in a deep invariant architecture [13] and during pre-training and network output regularization [14]. An architecture relatively similar in nature to ours is that used by Goroshin et al. [6]. Temporal coherence is applied to sparse autoencoders, as part of a convolutional architecture. Unlike our approach, regularization is applied across every feature.

To be the best of our knowledge, there is only one implementation of temporal coherence where regularization is not applied uniformly across all features. The

Temporal Product Network (TPN) learns two sets of features simultaneously, one encoding temporal coherence, the other discovering variable features [15]. This work differs from ours as the two sets of features are seeking different aspects from the input. The invariant features are designed to encode object identity, with the variable set discovering object position. We are learning the two types of features for the single purpose of object identity encoding.

3 Model Description

This section provides an overview of the models we have tested, with a full description of the network cost function to be minimized during training. For clarity, as features encoded with temporal coherence respond uniformly to small changes in an input, they are referred to as invariant. Their temporally unconstrained counterparts are denoted as variable features. A model with temporal coherence applied across all features will be described as all-invariant, with its opposite, a plain sparse autoencoder known as invariant-free. Versions with a mixture of invariant and variable features are called mixed-representation.

3.1 Overview

To investigate the effects of mixing invariant and variable features together in a single representation, we required a model capable of extracting good quality features from a visual dataset, in an unsupervised manner. The sparse autoencoder [5] is a well studied model fitting these requirements. An input layer is passed to a fully connected hidden layer, which is in turn passed to a fully connected output layer of the same dimensionality as the input. During training, network weights are adjusted via gradient descent to reconstruct the input as closely as possible at the output layer. Reconstruction of each input during training guides the representation to retain as much information present in the input as possible. This has the added benefit of avoiding the trivial constant solution, which is essential for temporal coherence to be applied successfully. Sparsity regularization forces a dictionary of distinct commonly occurring features to form in the hidden layer. Feature learning by this method is known to work well in unsupervised vision problems [16]. Once training is complete, the output layer is removed, with the new representation formed by the activations of all units in the hidden layer.

Whilst sparse autoencoders have previously been adapted to encode temporal coherence, our investigation mixes invariant and variable features in a single representation. Similar to existing methods, temporal coherence is applied by minimizing the difference in feature vectors between consecutive frames for the neurons encoding invariance. Instead of applying temporal coherence across all hidden layer neurons, we only apply the regularization to a variable size subset of the total available units. Remaining neurons are free to discover variable information components. Sparsity is applied across the entire hidden layer, making no distinction between invariant and variable neurons. To compare the

mixed representation with existing architectures, we also trained all-invariant and invariant-free versions of our model. A diagram of the network is shown in Figure 1.

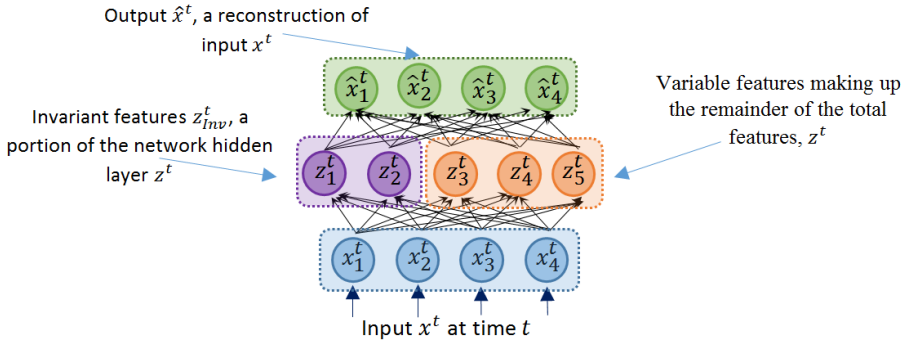

Fig. 1. Diagram of autoencoder architecture. An input x^t of dimensionality M is passed to a hidden layer, with dimensionality N (in this diagram, $M = 4$ and $N = 5$). The hidden layer z^t connects to an output of the same dimension as the input. Temporal coherence regularization is applied across the invariant portion of the hidden layer z^t_{Inv} of dimensionality $P \leq N$. In this case, $P = 2$. If $P = 0$, the network does not encode temporal coherence, when $P = N$ the representation is all-invariant.

3.2 Cost function

An autoencoder is a fully connected network with two components, the encoder and decoder modules. For time-series data, an input at time t is given as $x^t \in \mathbb{R}^M$, out of a total T frames. Inputs are passed to the autoencoder's hidden layer of neurons via the encoder section. Hidden layer neuron activations $z^t \in \mathbb{R}^N$ are calculated using an affine transformation via the encoder weights $W^E \in \mathbb{R}^{N \times M}$ and bias $b^E \in \mathbb{R}^{\mathbb{N}}$, followed by a suitable non-linearity $f()$ to give $z^t = f(W^E x^t + b^E)$. Similarly, the decoder section output activation values $\hat{x}^t \in \mathbb{R}^M$ are calculated using a similar process via the decoder weights $W^D \in \mathbb{R}^{M \times N}$ and bias $b^D \in \mathbb{R}^M$ to produce $\hat{x}^t = f(W^D z^t + b^D)$. For the non-linearity, we chose the commonly used sigmoidal activation function [17].

The first term in the model cost function is the reconstruction term. By constraining the difference between network input and output to be as small as possible, information loss is avoided. As mentioned previously, this ensures the trivial constant solution is avoided. The reconstruction term L_{Rec} is given as follows, with θ representing the encoder and decoder weights and biases:

$$L_{Rec}(x^t, \theta) = \sum_{t=1}^{T} \|x^t - \hat{x}^t\|^2 \tag{1}$$

We have imposed sparsity in the autoencoder using Kullback-Leibler (KL) Divergence across the hidden layer activations. KL-Divergence is an approximation to L_1-regularization known to perform well on vision problems [18]. During training, the average activation of each hidden layer neuron is calculated over all training examples as $\hat{\rho}_n = \frac{1}{T}\sum_t^T z_n^t$. The sigmoidal activation function constrains active and inactive neurons to generally have values very close to one and zero respectively. This characteristic enables the KL-Divergence term to enforce sparsity by constraining the average activation of each hidden layer neuron to be as close to a desired value ρ. The sparsity term L_{Spar} is calculated as:

$$L_{Spar}(x^t, \theta) = \sum_{n=1}^{N} KL(\rho||\hat{\rho}_n) = \sum_{n=1}^{N} \rho \log \frac{\rho}{\hat{\rho}_n} + (1-\rho)\log \frac{1-\rho}{1-\hat{\rho}_n} \tag{2}$$

The invariant term L_{Inv} regularises hidden layer neurons with temporal coherence. The term is applied to the invariant subset of the total hidden layer neurons $z_{Inv} \in \mathbb{R}^P$, where $P \leq N$ is an adjustable parameter setting the number of invariant features. When $P = 0$ the network is an invariance-free sparse autoencoder with no additional temporal cost. When $P = N$, temporal coherence is encoded in every hidden layer neuron, similar to the architecture described in [6]. With values inbetween, a mixed representation of invariant and variable features forms. Remaining neurons in the hidden layer are not regularized with temporal coherence, and make up the variable features. It is important to note that the invariant and variable neurons combine to make a single representation. Temporal coherence is enforced in the invariant neurons by minimizing the difference in hidden layer activations for every pair of adjacent frames in the training set:

$$L_{Inv}(x^t, \theta) = \sum_{t=1}^{T-1} \|z_{Inv}^t - z_{Inv}^{t+1}\|^2 \tag{3}$$

Putting these components together, the complete cost function to be minimized during training is given as:

$$L(x^t, \theta) = \sum_{t=1}^{T} \|x^t - \hat{x}^t\|^2 + \alpha \sum_{t=1}^{T-1} \|z_{Inv}^t - z_{Inv}^{t+1}\|^2 + \beta \sum_{n=1}^{N} KL(\rho||\hat{\rho}_n) \tag{4}$$

where the α and β values are hyperparameters used to control the influence of the temporal coherence and sparsity terms respectively. Similar to when $P = 0$, when the α parameter is set to zero, the network is a sparse autoencoder, with no temporal coherence regularization.

4 Evaluation of our models

4.1 Datasets

To evaluate the representation learning capacity of our different models, we required datasets consisting of distinct objects moving through a visual field. To keep things simple, we restricted our tests to videos of single objects moving across a uniform blank background. Whilst the unsupervised architecture does not require labelled data, it was necessary that datasets contained object identity labels to allow the training of a supervised classifier acting as a performance metric.

Toy-data Shapes Dataset Initial proof-of-concept work was carried out on a toy dataset. The video consists of a sequence of five simple objects moving through a 12×12 pixel visual field (see figure 2). For each sequence, a shape is positioned at an edge of the visual field, moving in a perpendicular direction, without changing direction or speed until it hits the opposite side. At this point, the object moves one pixel in a perpendicular direction to its previous motion before making the journey back across the visual field. This process is repeated until an object has traversed the visual field in each of the four cardinal directions. Once an object has exhausted its motion, the next shape in the dataset appears and the sequence starts again until all five shapes have appeared. Object speed remains constant, and there is no rotation. Three sequences from this dataset are shown in figure 3.

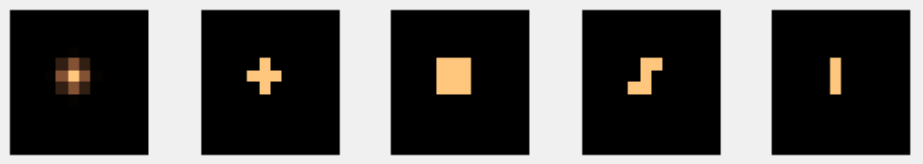

Fig. 2. The five simple objects used in the toy-data shapes dataset

COIL20-variant Dataset COIL20 is a well known benchmark dataset for visual problems [11]. The dataset is comprised of 20 different labelled objects, in sequences of 72 images each. Each sequence shows an object rotating on its axis through 360°, in 5° increments. Whilst not strictly a video, the images can be shown in sequence to give a good approximation. As we wanted objects to be able to move around the visual field, COIL20 in its existing form was not appropriate, so we manipulated the dataset for our purposes. COIL20 was reduced from its original 128×128 pixel size down to 24×24 images. For every shape, each 72 image sequence was placed in turn over a blank 48×48 pixel background, with

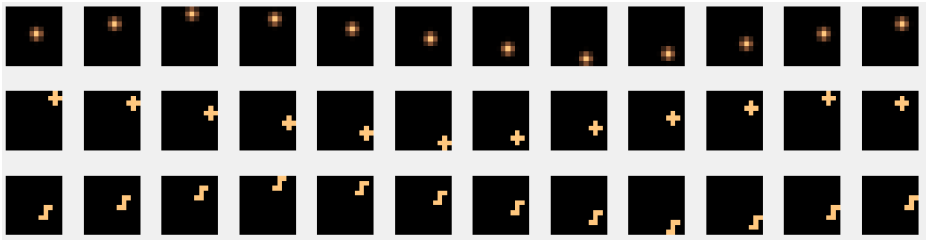

Fig. 3. Three sequences from the toy-data shapes dataset

the positioning and movement for each object directed by a random generator. There is an 85% chance an object carries on moving in the same direction as the previous frame, with reduced probability direction might change by either $\pm 22.5°, 45°$ or $67.5°$. Once an image reached an edge of the visual field, it was sent back in the direction it arrived. This process was repeated 20 times for each object producing 1440 examples for each different shape. Three sequences from our COIL20-variant dataset are shown in figure 4.

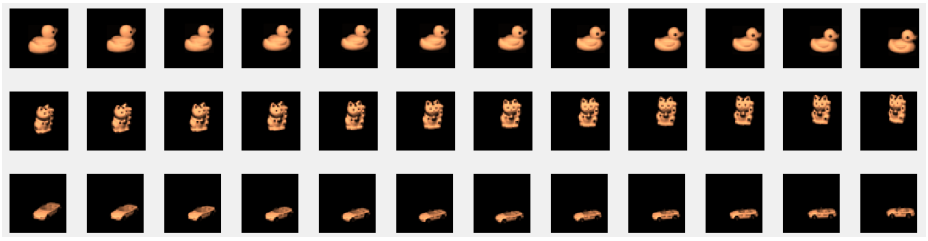

Fig. 4. Three 12 frame example sequences from the COIL20-variant dataset

Fish Dataset For our final test, we wanted a dataset with increased image, motion and rotation complexity. In their 2011 paper describing object recognition and pose detection using SFA, Franzius, Wilbert and Wiskott used a dataset consisting of 3D models of fish. The fish rotate and move with variable speed and direction [10]. After getting in touch with the authors, they kindly provided us the code with which to create their fish dataset. The parameters used to control the motion of the artificial fish are user provided. We kept to the same settings described in the SFA paper, with a couple of exceptions. Firstly, we removed the chance a fish might randomly switch to a different type of fish between frames. This was done to ensure the dataset contained the same number of examples for each fish. Video sequences were created with 2000 frames of each type of fish. Secondly, to reduce the time taken to train our networks, we only used the first 15 out of the 25 available types of fish in our dataset. Images were resized from

155 x 155 pixels to 48 x 48 pixels. Three example sequences of frames from this dataset can be seen in figure 5.

Fig. 5. Three 8 frame sequences from the fish dataset.

4.2 Experimental Setting

Unsupervised Architecture Our experiments proceeded as follows. For each dataset, the number of hidden layer neurons in the unsupervised architecture was fixed by training networks with an increasing number of neurons. We set the number of hidden layer neurons to be the network size at which the reconstruction term ceased reducing. The number of hidden layer units trained on the shapes dataset was fixed at 120 neurons, COIL20-variant at 400, and the fish dataset at 600. For each experiment, a range of invariant to variable feature splits for the mixed representations were picked. For the shapes dataset, five different models were chosen, with the number of invariant neurons increasing from 0 to 120 in increments of 30. All remaining neurons were trained without the invariant term applied. For the COIL20-variant and fish dataset, increments were set at 100 neurons, giving 5 and 7 different models respectively.

Due to the relatively small number of examples available in the shapes dataset, 2200 in total, the unsupervised architecture was trained using all available examples. As the COIL20-variant and fish dataset contain significantly more examples, we reduced the training set to half the available examples. To preserve the temporal integrity of the datasets, examples were split up into runs of 50 frames, each run containing image sequences of a single object. For each test, half the available runs for each object were picked at random for the training set, with remaining examples providing a validation set.

Unsupervised training of each model was carried by minimizing the cost function by gradient descent. Two measures were applied to determine training stopping conditions. At increments of 100 epochs, training was halted and the overall cost function and its individual terms were evaluated on the validation set.

During each pause, a classifier was also trained to predict object identity using a labelled training set, recording classification accuracy, also on the validation set. Whilst we used a supervised measure for helping determine when to halt training, we did not apply any supervised fine-tuning to guide network weights.

Suitable values for the α and β cost function hyperparameters were chosen by performing a grid search and observing the evolution of the invariant and sparsity terms in the cost function as training progressed.

Supervised Classifier Temporal coherence is generally used as a method for encoding input variance associated with object identity. For this reason, we passed labelled outputs from each of our various models to a supervised classifier trained to recognise the individual shapes present in the videos. Although we are using a supervised metric to test the performance of our unsupervised architecture, the labelled data has no effect on the representations learned by the various autoencoders.

For the supervised part of our experiments, we also wanted to observe whether the quantity of labelled data made available to the object identity classification task influenced which of the mixed representations produced the best results. For the shapes dataset, we trained a classifier to recognise object identity using one in every two, five and twenty labelled examples from the total available data. For the COIL20-variant and fish datasets, six different classifiers were trained, picking either every second, third, fifth, tenth, twentieth or fiftieth examples. All remaining examples provided the testing set.

4.3 Results

The classification accuracy results from each dataset are presented in figures 6, 7 and 8. In each figure, the groups of bars indicate results conducted with the same number of classifier training examples together, with the number indicated on the x-axis. The bars making up each group indicate an increasing number of invariant neurons as a percentage of total neurons available. The results displayed in each figure depict the highest classification accuracy recorded for that particular setting, over any of the cost function hyperparameter values tested. Each test was run four times. The results displayed in each graph represent average classification accuracy on a testing set, with error bars corresponding to one standard deviation.

The most obvious thing standing out from each figure is the reduction in classification accuracy as the amount of labelled data is reduced. As the quantity of training data is reduced, a classifier has less information with which to make its predictions, which correspondingly suffer. More significantly, it is a representation encoding temporal coherence is some shape or form that produces the highest classification accuracy every time. Across every test, regularizing for temporal coherence has improved the capture of information related to object identity.

From the point of view of our experiments, the most important result is the performance observed from the mixed representations. For each dataset, when

the quantity of labelled data passed to the classifier is at its largest, a mixed representation always produced the highest classification accuracy. In the case of our shapes and COIL20-variant datasets, optimal performance swings to the invariant only representation when the amount of classifier data is at its lowest. A split representation produces the best result every time for the fish dataset. We have attributed this anomaly to the higher complexity of the images, motion and rotation of objects in this dataset. The split representations producing the best results are those where the proportion of invariant to variable features is the smallest we tested. As mentioned, an invariant-free representation failed to produce the best results from a classifier in any of our tests.

Table 1 gives the results gained when the classifiers are trained using half the available data. Classification accuracy is recorded for the all-invariant and invariant free models, and the best performance from a mixed representation, with the proportion of invariant features indicated. As a final comparison, classification accuracy is also provided from when a classifier is trained using the raw image data. In each of these cases, a mixed representation produced the best performance.

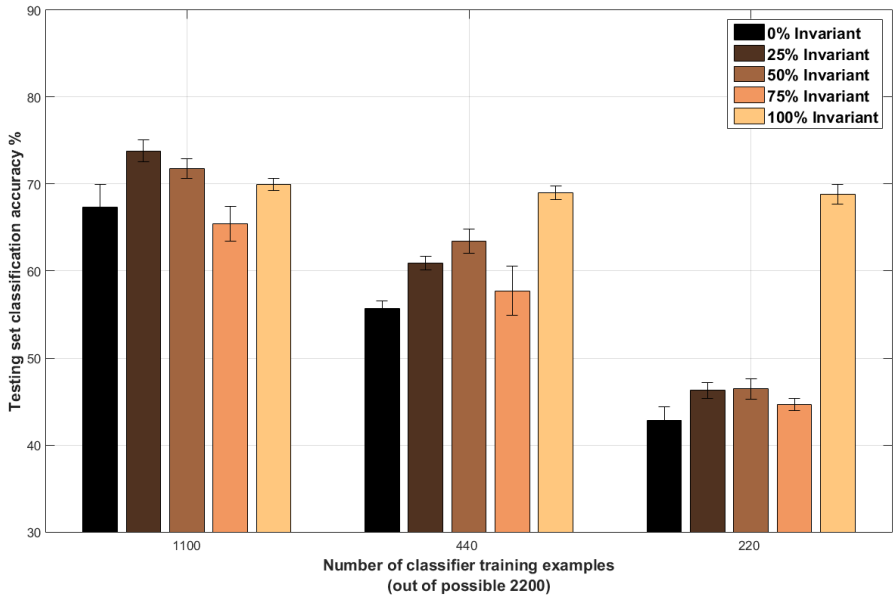

Fig. 6. Classification accuracies for the shapes dataset. Groups of bars display average classification accuracy, along with one standard deviation, from classifiers trained with a set number of examples, the number of which is given on the x-axis. Each bar in a group refers to a variant of the unsupervised architecture. Proportion of total features that are invariant for each bar is provided in the legend. The same notation is used in figures 7 and 8

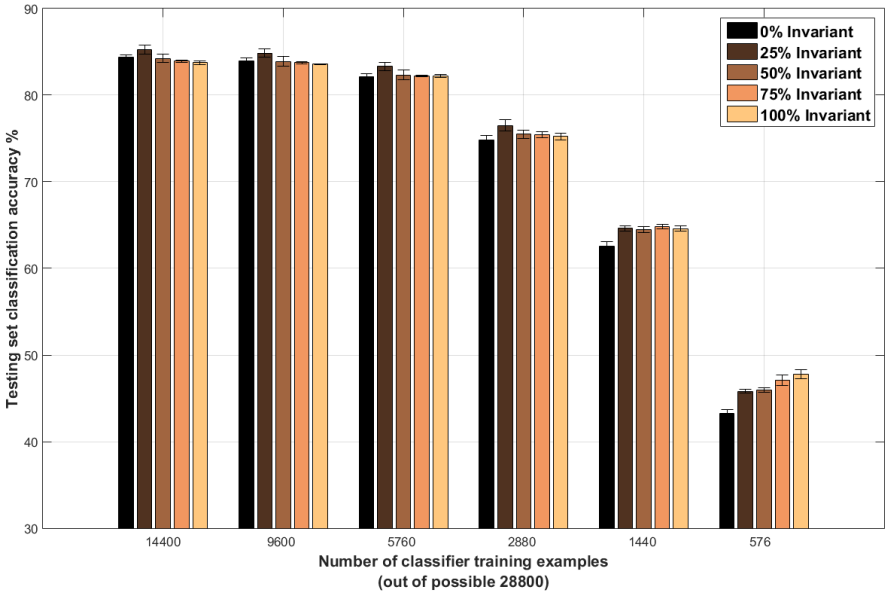

Fig. 7. Classification accuracies for the COIL20-Variant dataset.

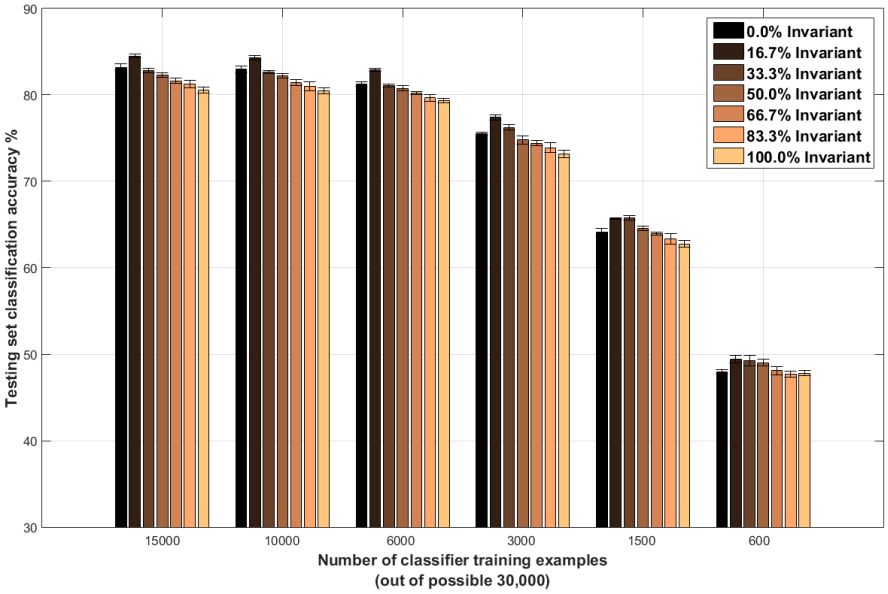

Fig. 8. Classification accuracies for the Fish dataset.

Table 1. Average object classification accuracy from each model when 50% of data is passed to the classifier. For the mixed representation, values are given for the model producing the highest classification accuracy, with the percentage of invariant neurons that produced this result given. Classification accuracy from each dataset was also recorded for the situation where the classifier is trained on the raw image data, with no representation learning applied.

Dataset	**Raw Data**	**Best Mixed (% invariant)**	**Sparse Only**	**All Invariant**
Shapes	30.16%	**73.82%** (25%)	69.38%	66.77%
COIL20-variant	34.06%	**85.27%** (25%)	84.38%	83.76%
Fish	57.29%	**84.51%** (16.67%)	83.21%	80.59%

5 Discussions

The goal of regularizing features with temporal coherence is to learn stable variance common to each pair of frames in video. Image-specific details are discarded to facilitate the discovery of temporally coherent structure on the assumption this encodes details related to object identity. Compared to an invariant-free representation, the quantity of patterns collected for each object is reduced on the basis that their classification quality is increased. A mixed-representation seeks to combine image-specific details, distinguishing adjacent frames apart, alongside temporally coherent information common over adjacent pairs of frames. A larger quantity of patterns is produced for each object, whilst still encoding the invariant structure related to object identity missing in an invariant-free representation. A greater percentage of the input structure is captured, losing some of the generalization present in an all-invariant representation.

When the quantity of labelled data is scarce, a classifier's performance will improve when the total pool of possible patterns for each object is reduced. Similarly, if every feature in a labelled example contains information common to semantically similar neighbours, a classifier will have a greater chance of learning how to differentiate between objects. Conversely, when each example contains image-specific details not applicable to other examples of the same object, this will confuse a classifier when there is not enough labelled data available to build a full picture of each object. When data is scarce, the generalized features learned by an all-invariant representation improves a classifier's chances of learning the details required to distinguish between objects.

When the quantity of labelled data is increased, classification performance is observed across every model tested, but the benefits of having an all-invariant representation start to dampen. Encoding the structure common to neighbouring examples of distinct objects, alongside a greater amount of image-specific detail, enables a classifier to build a more complete picture of each object from a mixed representation. Information discarded by an all-invariant representation, confusing to a classifier when data is scarce, starts to become useful. When this point is reached, object identity encoding benefits from having as much image-specific data as possible, so long as a portion of features remain invariant. This

can clearly be seen from our tests, as a mixed representation predominantly composed of variable features performs optimally over every test where labelled data is abundant.

Optimal performance was observed when a classifier was passed a representation encoding temporal coherence in some manner in all of our tests. Consistent with previous work, extracting temporally stable aspects from video improves the encoding of information related to object identity [12]. When labelled data is plentiful, and a classifier is able to build a more complete picture of the structure underlying each distinct object, applying a mixed representation improves object identity encoding.

5.1 Future Work

The findings of this work lead to quite a wide range of possibilities. Firstly, we have applied our mixed-representation to a relatively simple single layered sparse autoencoder. Temporal coherence has been successfully applied to a wide range of models, including architecturally more complex deep networks. It would be interesting to investigate whether or not applying a mixed-representation to these existing models will boost object identity performance further.

Another route of study would be to consider more complex datasets. The videos in our study were all artificially generated, and only contain a single object at a time moving across a uniform background. Tests could be conducted to observe how a mixed representation copes with more complex images, including more than one object, and natural images.

Finally, whilst we have been capturing a mixture of invariant and variable features, this study has only been concerned with encoding object identity. As discussed in Section 2, there has been a small body of work applying temporal coherence to discover the what and the where of objects [10][15]. As position and motion of objects generally change over quicker time scales than object identity, it is possible that our mixed representation also contains features associated with these properties. This is the primary direction we are going to be looking for our own future studies with this work.

6 Conclusions

We trained a range of sparse autoencoders, encoding temporal coherence over different portions of the available hidden layer units. We can summarize our findings as follows:

- Optimal performance across all tests involved an architecture encoding temporal coherence across a portion of the available features.
- For the majority of tests, the greatest classification performance was achieved when the classifier input received a representation mixing temporally coherent features alongside 'variable' counterparts.
- When a large amount of labelled data was available, a mixed representation always produced the best encoding.

We believe our work demonstrates there are situations where the previously accepted method of applying temporal coherence uniformly across all features is non-optimal. By discovering a mixed representation, consisting of both invariant and variable neurons, object identity encoding can be improved.

Acknowledgements

We'd like to pass on our gratitude to Laurenz Wiskott and Niko Wilbert for providing us with access to their fish dataset, and to Jon Shapiro for the invaluable contribution he has made to this project. This project has been funded by the EPSRC.

References

1. Becker, S.: Learning to categorize objects using temporal coherence. In C.L. Giles, S.J. Hanson & J.D. Cowan (Eds), Advances in Neural Information Processing Systems, 5, 361-368 (1993)
2. Hinton, G.E.: Connectionist learning lrocedures, Artificial Intelligence, 40, 185-234 (1989)
3. Foldiak, P.: Learning Invariance from Transformation Sequences, Neural Computation, 3, 194-200 (1991)
4. Mitchison, G.: Removing Time Variation with the Anti-Hebbian Differential Synapse, Neural Computation, 3, 312-320 (1991)
5. Olshausen, B. and Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature, 381, 607-609 (1996)
6. Goroshin, R., Bruna, J., Tompson, J., Eigen, D. and LeCun, Y.: Unsupervised Learning of Spatiotemporally Coherent Metrics, In Proceedings of the IEEE International Conference on Computer Vision, 4086-4093 (2015)
7. Stone, J. and Bray, A.: A Learning Rule for Extracting Spatio-temporal Invariances. Network: Computation in Neural Systems, 6(3), 429-436 (1995)
8. Wiskott, L and Sijnowski, J.: Slow Feature Analysis: Unsupervised Learning of Invariances. Neural Computation, 14(4) 715-770 (2002)
9. Wiskott, L. and Berkes, P.: Is slowness a learning principle of the visual cortex?, Zoology. 106, 373-382 (2003)
10. Franzius, M., Wilbert, N. and Wiskott, L.: Invariant Object Recognition and Pose Estimation with Slow Feature Analysis, Neural Computation 23, 2289-2323, (2011)
11. Nene, S.A., Nayar, S.K. and Murase, H.: Columbia Object Image Library (COIL-20), technical report, CUCS-005-96 (1996)
12. Mobahi, H., Collobert, R. and Weston, J.: Deep Learning from Temporal Coherence in Video. International Conference of Machine Learning, 2009, 737-744 (2009)
13. Zou, W.J., Zhu, S., Ng, A. and Yu, K.: Deep Learning of Invariant Features via Simulated Fixations in Video. Advances in Neural Information Processing Systems, 3212-3220 (2012)
14. Weston, J., Ratle, F., Mobahi, H. and Collobert, R.: Deep Learning via Semi-Supervised Embedding, Neural Networks: Tricks of the Trade, In Lecture Notes in Computer Science, 7700, 639-655, Springer, Berlin (2012)
15. Gregor, K. and LeCun Y.: Emergence of Complex-Like Cells in a Temporal Product Network with Local Receptive Fields. Technical Report, arXiv: 1006.0448 (2010)

16. Bengio, Y.: Deep learning of representations: Looking forward. In Statistical language and speech processing, 1-37. Springer Berlin Heidelberg, 2013.
17. Duch, W. and Jankowski, N.: Survey of neural transfer functions, Computing Surveys, 2, 163-213 (1999)
18. Bradley, D.M. and Bagnell, J.A.: Differentiable Sparse Coding, Advances in Neural Information Processing Systems, 21, 113-120, (2009)