*Cognitive Computation*, doi: 10.1007/s12559-016-9445-1.

# A Hierarchical Predictive Coding Model of Object Recognition in Natural Images

**M. W. Spratling**
King's College London, Department of Informatics, London. UK. michael.spratling@kcl.ac.uk

**Abstract**

**Background** Predictive coding has been proposed as a model of the hierarchical perceptual inference process performed in the cortex. However, results demonstrating that predictive coding is capable of performing the complex inference required to recognise objects in natural images have not previously been presented.

**Methods** This article proposes a hierarchical neural network based on predictive coding for performing visual object recognition.

**Results** This network is applied to the tasks of categorising hand-written digits, identifying faces, and locating cars in images of street scenes. It is shown that image recognition can be performed with tolerance to position, illumination, size, partial occlusion and within-category variation.

**Conclusions** The current results, therefore, provide the first practical demonstration that predictive coding (at least the particular implementation of predictive coding used here; the PC/BC-DIM algorithm) is capable of performing accurate visual object recognition.

**Keywords:** predictive coding; neural networks; object recognition; implicit shape model; deep neural networks; sparse coding

## 1 Introduction

Localising and identifying items in visual scenes is of fundamental importance for many activities carried out by humans and other species. To solve this complex computational task the brain is required to perform perceptual inference in order to find the most likely causes of the visual input. This process of object recognition is believed to be performed by a hierarchy of cortical regions along the ventral occipitotemporal pathway (DiCarlo et al., 2012; Goodale and Milner, 1992; Krüger et al., 2013; Ungerleider and Mishkin, 1982).

Predictive coding (PC) is a highly influential theory of cortical information processing (Clark, 2013; Friston and Kiebel, 2009; Huang and Rao, 2011; Kok and de Lange, 2015; Rao and Ballard, 1999; Spratling, 2014b, ress). PC is specifically suited to performing perceptual inference. Furthermore, PC can be implemented as a hierarchical neural network. PC should thus be suited, both at the functional and neurophysiological levels, to simulating object recognition. However, to-date this has not been demonstrated explicitly. This article presents the first demonstration that PC can perform object recognition in natural images. Specifically, the current results show that a particular implementation of PC (the PC/BC-DIM algorithm[a]) can locate cars in natural images of street scenes, identify individuals from their face, and can categorise numbers in images of hand-written digits.

Object recognition requires the brain to solve an inverse problem: one where the causes (the shapes, surface properties, and arrangements of objects) need to be inferred from the perceived outcome of the image formation process. Inverse problems are typically ill-posed, meaning that they have multiple solutions (or none at all). For example, different sets of objects arranged in different configurations and viewed under different lighting conditions could potentially give rise to the same image. Solving such an ill-posed problem requires additional constraints to be imposed in order to narrow down the number of possible solutions to the single, most likely, one. In other words, constraints are required to infer the most likely causes of the sensory data. Constraints on visual inference might come from many sources, including knowledge learnt from prior experience (such as typical lighting conditions and the shapes and sizes of common objects), the recent past (knowledge about recently perceived causes, and expectations about how these might change or stay the same), and the present (such as information from elsewhere in the image or from another sensory modality).

[a]PC/BC-DIM is a version of PC (Rao and Ballard, 1999) reformulated to make it compatible with Biased Competition (BC) theories of cortical function (Spratling, 2008a,b), and that is implemented using Divisive Input Modulation (DIM; Spratling et al., 2009) as the method for updating error and prediction neuron activations. DIM calculates reconstruction errors using division, which is in contrast to other implementations of PC that calculate reconstruction errors using subtraction (Huang and Rao, 2011; Spratling, 2008a, ress). The divisive method is preferred as it results in non-negative firing-rates and is thus more biologically-plausible (Spratling, 2008a, ress). Furthermore, it has stable dynamics and converges more quickly allowing it to be used to build large-scale models (Spratling, ress; Spratling et al., 2009).

PC proposes a scheme for applying such constraints in order to solve the inverse problem of vision. Specifically, PC suggests that the brain learns, from prior experience, an internal model of the world, or multiple models of specific aspects of the world embedded in different cortical regions. This internal model encodes possible causes of sensory inputs as parameters of a generative model (the weights of prediction neurons). New sensory inputs are then represented in terms of these known causes (by the activation of the prediction neurons). Determining which combination of the many possible causes best fits the current sensory data is achieved through an iterative process of minimising the error between the sensory data and the expected sensory inputs predicted by the causes. This inference process performs "explaining away" (Kersten et al., 2004; Lochmann and Deneve, 2011; Lochmann et al., 2012; Spratling, 2012; Spratling et al., 2009): possible causes compete to explain the sensory evidence, and those causes that are best supported by the evidence, explain away that evidence preventing it from supporting competing causes. This suppression of alternative explanations typically results in a sparse set of predicted causes.

Object recognition requires perceptual representations that are sufficiently selective for shape and appearance properties (to distinguish one individual or one object category from another) as well as being sufficiently tolerant to changes in shape and appearance caused by illumination, viewpoint, partial-occlusion, within category variation, and non-rigid deformations (to allow the same object or object category to be recognised under different viewing conditions) (DiCarlo and Cox, 2007; DiCarlo et al., 2012; Krüger et al., 2013; Pinto et al., 2008; Riesenhuber and Poggio, 1999). It is generally believed that such selectivity and tolerance is built up slowly along the ventral pathway (Gilbert, 1996; Kobatake and Tanaka, 1994; Logothetis, 1998; Mountcastle, 1998; Oram and Perrett, 1994; Rust and Dicarlo, 2010; Wallis and Bülthoff, 1999). Different mechanisms are required to learn more selective representations and to learn more tolerant representations (Riesenhuber and Poggio, 1999; Spratling, 2005). Hence, several existing models of object recognition consist of alternating layers of neurons that perform these two operations in order to form more specialized representations in one layer, and more invariant representations in the next layer (Ciresan et al., 2012; Fukushima, 1980, 1988, 2005; Jarrett et al., 2009; Krizhevsky et al., 2012; LeCun and Bengio, 1995; LeCun et al., 1998, 2010; Mutch and Lowe, 2008; Riesenhuber and Poggio, 1999; Serre et al., 2007; Theriault et al., 2013).

The experiments described in this article were performed using a two-stage hierarchy of PC/BC-DIM networks. The same hierarchical arrangement of PC/BC-DIM networks has previously been used to model word recognition (Spratling, 2016d, except this previous work, in contrast to the current work, used hard-coded weights and inter-stage feedback connections), and to model the learning of receptive fields in cortical areas V1 and V2 (Spratling, 2012, except that previous work used a different learning procedure to that described here). In the proposed model, the synaptic weights for alternate processing-stages are defined differently, in order to form receptive fields (RFs) that are specific to particular image features in one stage, and connections that generalise over these features in the subsequent stage. However, following learning, both stages operate identically. Both stages implement PC/BC-DIM, and hence, perform explaining away. The advantages of using explaining away to perform each of these operations have been demonstrated in two previous publications: Spratling (2016a) has shown that explaining away has advantages for producing neural responses that are selective to image features, while Spratling (2016c) has shown that explaining away has advantages for producing responses that generalise over changes in appearance. Here, it is shown that combining these two applications of PC/BC-DIM into one hierarchical neural network allows PC/BC-DIM to be used for object recognition.

# 2 Methods

The experiments were performed using a two-stage hierarchical neural network model, as illustrated in fig. 1a. The activations of the neurons in both stages were calculated using the PC/BC-DIM algorithm (as described in sect. 2.3). However, because different methods were used to learn the weights of each processing-stage (as described in sect. 2.1), they played different roles in the object recognition process.

## 2.1 Training

The training procedure for the first processing-stage was as follows.

**Image patches were extracted from the grayscale training images.** For those tasks in which the location and scale of the object was fixed (digit and face recognition), each training image was treated as a patch. In contrast, for those tasks in which the location of the object could vary (car recognition), patches were extracted from around keypoints (located using the Harris corner detector). Furthermore, in this case, to help distinguish cars (the "targets") from other objects (the "non-targets") that were also present in the test images, two sets of patches were obtained: those containing parts of the to-be-recognised objects, and those

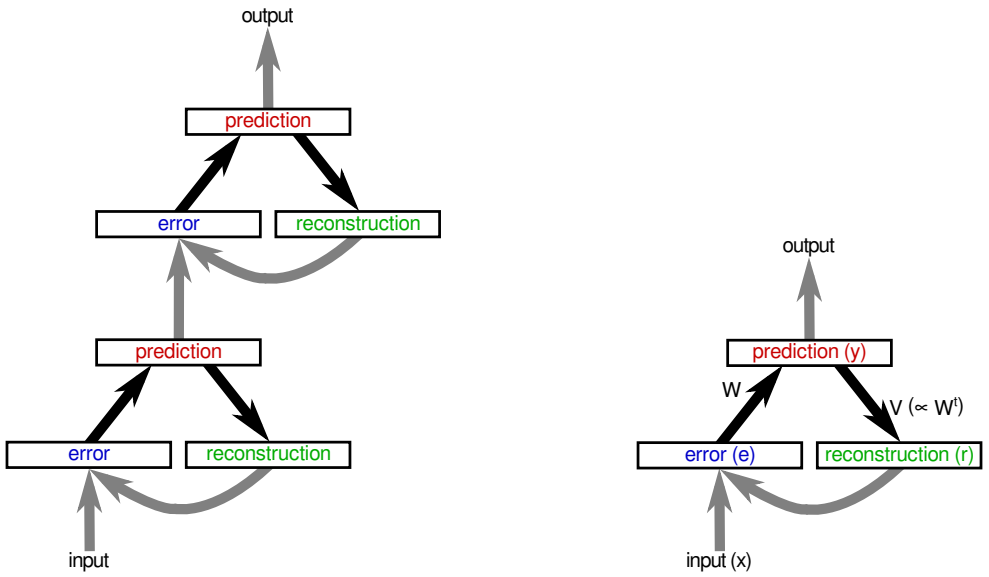



**Figure 1:** (a) The two-stage hierarchical PC/BC-DIM network used in the simulations described in this paper. Rectangles represent populations of neurons and arrows represent connections between those neural populations. The first processing-stage receives visual input. The second processing-stage receives input that is the steady-state prediction neuron responses generated by first processing-stage. (b) In each processing-stage the population of prediction neurons constitute a model of the input environment of that processing-stage. Individual neurons represent distinct causes that can underlie the input (*i.e.*, latent variables). The belief that each cause explains the current input is encoded in the activation level, $\mathbf{y}$, and is used to reconstruct the expected input given the predicted causes. This reconstruction, $\mathbf{r}$, is calculated using a linear generative model (see eq. 1). Each column of the feedback weight matrix $\mathbf{V}$ represents an "elementary component", "basis vector", or "dictionary element", and the reconstruction is thus a linear combination of those components. Each element of the reconstruction is compared to the corresponding element of the actual input, $\mathbf{x}$, in order to calculate the residual error, $\mathbf{e}$, between the predicted input and the actual input (see eq. 2). The errors are subsequently used to update the predictions (via the feedforward weights $\mathbf{W}$, see eq. 3) in order to make them better able to account for the input, and hence, to reduce the error at subsequent iterations. The responses of the neurons in all three populations are updated iteratively to recursively calculate the values of $\mathbf{y}$, $\mathbf{r}$, and $\mathbf{e}$. The weights $\mathbf{V}$ are the transpose of the weights $\mathbf{W}$ (but each set of weights may be normalised differently). Given that the $\mathbf{V}$ weights are proportional to the $\mathbf{W}$ weights there is only one set of free parameters. All other connections (shown using gray arrows) are fixed to have binary values and to provide one-to-one connectivity between corresponding neurons in the pre- and post-synaptic populations.

containing non-target image regions (obtained from images that did not contain the target object). To deal with changes in scale, the training images were rescaled to six different sizes, and patches were extracted from each set of resized training images.

**The image patches were clustered to form a dictionary.** The image patches were clustered using the hierarchical agglomerative clustering algorithm, with zero-mean normalised cross correlation (ZMNCC)[b] between the most different members of each cluster as the measure of similarity. Clustering was terminated once the ZMNCC between all clusters was less than a similarity threshold ($\kappa$). Those clusters with fewer than $\lambda$ members were discarded. The arithmetic mean of the patches forming the remaining clusters were used as the dictionary. For those tasks in which there were multiple classes (digit and face recognition) clustering was performed separately on the image patches extracted from images of each class. Similarly, for those tasks in which there was only one class of object to be recognised (cars) clustering was performed separately for target and non-target image patches. To deal with changes in scale, separate clustering of patches taken from each size of image was used.

The PC/BC-DIM algorithm can be used to allow the first processing-stage to find matches between the dic-

[b] Also known as the sample Pearson correlation coefficient.

tionary elements and an input image. The prediction neuron responses will represent the closeness of the match between the dictionary element and the image. If the dictionary elements are thought of as templates for object parts, then PC/BC-DIM can be considered as a method of template matching, but one that has considerable advantages over traditional template matching methods (Spratling, 2016a). Specifically, by using PC/BC-DIM the match between a template and the image takes into account the evidence provided by the image and the full range of alternative explanations represented by the other templates. In other words, PC/BC-DIM performs explaining away. The result is that the prediction neuron responses (representing the match between templates and image locations) are very sparse. Those locations that match a template can therefore be readily identified and there is greater tolerance to changes in appearance due to changes in viewpoint (Spratling, 2016a).

Image features are better distinguished using relative intensity (or contrast) rather than absolute intensity. Hence, template matching was performed with the first processing-stage after the input image had been pre-processed as follows. The grayscale input image $I$ was convolved with a 2D circular-symmetric Gaussian mask $g$ with standard deviation equal to $\sigma$ pixels, such that: $\bar{I} = I * g$. $\bar{I}$ is an estimate of the local mean intensity across the image. To avoid a poor estimate of $\bar{I}$ near the edges of the image, it was first padded on all sides by $4\sigma$ pixels with intensity values that were mirror reflections of the image pixel values near the edges of $I$. $\bar{I}$ was then cropped to be the same size as the original input image. The relative intensity can be approximated as $\mathbf{X} = I - \bar{I}$. For biological-plausibility the PC/BC-DIM algorithm requires inputs to be non-negative (weights and neural activations are also non-negative). To produce non-negative input to the PC/BC-DIM algorithm, the positive and rectified negative values of $\mathbf{X}$ (representing, respectively, increases and decreases in local contrast, or ON and OFF channels) were both used to form the input to the first processing-stage. The weights of each prediction neuron in the first processing-stage were defined by processing each dictionary element in an identical way to the input image. These weights were normalised so that the weights forming the RF of each prediction neuron summed to one.

The training procedure for the second processing-stage was as follows.

**First-stage prediction neuron responses were calculated for all the images in the training set.** The weights of the first processing-stage were defined as described in the preceding paragraph. An image from the training set (after being pre-processed as described in the preceding paragraph) was presented as input to the first processing-stage, and the PC/BC-DIM algorithm (as described in sect. 2.3) was executed. This was repeated for every image in the training set, and the first-stage prediction neuron responses to each training image were recorded.

**The second-stage weights were defined based on the responses of the first-stage prediction neurons.** A separate second-stage prediction neuron was defined to represent each object that was to be recognised. For those tasks in which the class or identity of the object was to be determined (digit and face recognition), a prediction neuron for each class or individual was defined. For tasks in which the location and scale of the object could vary (car recognition) prediction neurons were defined for each location and scale. The weights of these second-stage prediction neurons were set to be proportional to the sum of the responses of the first-stage prediction neurons to all training images containing the to-be-recognised object.

By having weights that connect a second-stage prediction neuron to all the prediction neurons in the first stage that represent (parts of) members of the to-be-recognised object category (at a specific scale or location), the second-stage prediction neuron will respond when those image features are identified by the first processing-stage. The strength of response will depend not only on how many and how strongly the first processing-stage templates match the image, but will also depend on the weights of other second-stage prediction neurons. Specifically, the second processing-stage performs explaining away, meaning that if an image feature is consistent with more than one of the objects represented by second-stage prediction neurons, then the PC/BC-DIM algorithm will activate the neuron corresponding to the most likely object and suppress the image feature's support for alternative objects. The result is that the prediction neuron responses (representing the match between the image and a to-be-recognised objects) are very sparse. The true matches can therefore be readily identified and the generalisation over changes in appearance is more selective for those objects that have the most evidence (Spratling, 2016c).

For the task in which the location of the object could vary (*i.e.*, car recognition) second-stage prediction neurons were defined to signal the presence of the object at each location. If the task had required the recognition of objects seen from different directions, or at different orientations, then it would have been necessary to define different second-stage prediction neurons to represent these different views of the same object. Such model neurons can be seen to be analogous to view-tuned cells observed in inferior temporal cortex (Logothetis and Pauls, 1995; Logothetis et al., 1995). It would be possible to add a third processing stage to integrate information from such view-tuned neurons in order to signal the presence of the object irrespective of location or orientation. However, it is unlikely that such neurons, invariant to viewpoint, could be defined directly from the outputs of the first processing stage (*i.e.*, by skipping the view-tuned neurons). This is because first-stage to view invariant

connections would have to be very abundant, and this would allow the view invariant neurons to respond to combinations of image features that might appear in an image but not form the to-be-recognised object. In other words, attempting to increase tolerance to too quickly will lead to to a loss of selectivity. Hence, building PC/BC--DIM models that can recognise objects with greater tolerance to changes in appearance is likely to require the building of deeper hierarchical models (Anselmi et al., 2014; Poggio et al., 2015).

## 2.2 Recognition

Following the training of both stages, described above, the hierarchical PC/BC-DIM model can be used to recognise objects in novel, test, images. The test image is pre-processed into ON and OFF channels as described in sect. 2.1. These are input to the first processing-stage, and the PC/BC-DIM algorithm (as described in sect. 2.3) is executed. The first-stage prediction neuron responses are then provided as inputs to the second processing stage and the PC/BC-DIM algorithm (as described in sect. 2.3) is executed for the second-stage. The second-stage prediction neuron responses are then used to identify the to-be-recognised objects. For those tasks in which the location and scale of the object was fixed and for which each image contained exactly one object (digit and face recognition), the maximum response was taken to indicate the class of the image. For those tasks in which the location of the object could vary and in which the number of objects in each image could vary (car recognition), the presence of an object was indicated by prediction neurons responses that were peaks in the spatial neighbourhood and which exceeded a global threshold.

## 2.3 The PC/BC-DIM Algorithm

The main mathematical operation required to implement the PC/BC-DIM algorithm is the calculation of sums of products. The algorithm can therefore be equally simply implemented using matrix multiplication or convolution.

The matrix-multiplication version of PC/BC-DIM is illustrated in fig. 1b and was implemented using the following equations:

$$\mathbf{r} = \mathbf{V}\mathbf{y} \tag{1}$$

$$\mathbf{e} = \mathbf{x} \oslash [\mathbf{r}]_{\epsilon_2} \tag{2}$$

$$\mathbf{y} \leftarrow [\mathbf{y}]_{\epsilon_1} \odot \mathbf{W}\mathbf{e} \tag{3}$$

Where $\mathbf{x}$ is a ($m$ by 1) vector of input activations, $\mathbf{e}$ is a ($m$ by 1) vector of error neuron activations; $\mathbf{r}$ is a ($m$ by 1) vector of reconstruction neuron activations; $\mathbf{y}$ is a ($n$ by 1) vector of prediction neuron activations; $\mathbf{W}$ is a ($n$ by $m$) matrix of feedforward synaptic weight values, defined by the training process described in sect. 2.1; $\mathbf{V}$ is a ($m$ by $n$) matrix of feedback synaptic weight values; $[v]_\epsilon = \mathtt{max}(\epsilon, v)$; $\epsilon_1$ and $\epsilon_2$ are parameters; $\oslash$ and $\odot$ indicate element-wise division and multiplication respectively; and $\leftarrow$ means that the left-hand side of the equation is assigned the value of the right-hand side. The matrix $\mathbf{V}$ is equal to the transpose of the $\mathbf{W}$ but each column of $\mathbf{V}$ is normalised to have a maximum value of one. Hence, the feedforward and feedback weights are simply rescaled versions of each other.

The convolutional version of PC/BC-DIM was implemented using the following equations:

$$\mathbf{R}_i = \sum_{j=1}^{p} (\mathbf{v}_{ji} \star \mathbf{Y}_j) \tag{4}$$

$$\mathbf{E}_i = \mathbf{X}_i \oslash [R_i]_{\epsilon_2} \tag{5}$$

$$\mathbf{Y}_j \leftarrow [\mathbf{Y}_j]_{\epsilon_1} \odot \sum_{i=1}^{k} (\mathbf{w}_{ji} \star \mathbf{E}_i) \tag{6}$$

Where $\mathbf{X}_i$ is a 2-dimensional array representing channel $i$ of the input; $\mathbf{R}_i$ is a 2-dimensional array representing the network's reconstruction of $\mathbf{X}_i$; $\mathbf{E}_i$ is a 2-dimensional array representing the error between $\mathbf{X}_i$ and $\mathbf{R}_i$; $\mathbf{Y}_j$ is a 2-dimensional array that represent the prediction neuron responses for a particular class, $j$, of prediction neuron; $\mathbf{w}_{ji}$ is a 2-dimensional kernel representing the feedforward synaptic weights from a particular channel, $i$, of the input to a particular class, $j$, of prediction neuron, defined by the training process described in sect. 2.1; $\mathbf{v}_{ji}$ is a 2-dimensional kernel representing the feedback synaptic weights from a particular class, $j$, of prediction neuron to a particular channel, $i$ of the input; and $\star$ represents cross-correlation. The weights $\mathbf{v}_{ij}$ are equal to the weights

$\mathbf{w}_{ij}$ but are rotated by $180^o$ and are normalised so that for each $j$ the maximum weight value, across all $i$, is equal to one. Hence, the feedforward weights, between a pair of error-detecting and prediction neurons, and the feedback weights, between the corresponding pair of reconstruction and prediction neurons, are simply re-scaled versions of each other.

The matrix-multiplication and convolutional version of PC/BC-DIM are interchangeable, and which particular method was used depended on which was most convenient for the particular task. For example, the convolutional version was used when prediction neurons with identical RFs were required to be replicated at every pixel location in an image. To simplify the description of the proposed method, the rest of the text will refer only to the matix-multiplication version of PC/BC-DIM.

For all the experiments described in this paper $\epsilon_1$ and $\epsilon_2$ were given the values $\epsilon_1 = \frac{\epsilon_2}{max(\tilde{\mathbf{V}})}$ (where $\tilde{\mathbf{V}}$ is a vector containing the sum of each row of $\mathbf{V}$, *i.e.*, the sums of feedback weights targeting each reconstruction neuron) and $\epsilon_2 = 1 \times 10^{-2}$. Parameter $\epsilon_1$ prevents prediction neurons becoming permanently non-responsive. It also sets each prediction neuron's baseline activity rate and controls the rate at which its activity increases when a new stimulus appears at the input to the network. Parameter $\epsilon_2$ prevents division-by-zero errors and determines the minimum strength that an input is required to have in order to effect prediction neuron response. As in all previous work with PC/BC-DIM, these parameters have been given small values compared to typical values of $\mathbf{y}$ and $\mathbf{x}$, and hence, have negligible effects on the steady-state activity of the network. To determine this steady-state activity, the values of $\mathbf{y}$ were all set to zero, and eqs. 1 to 3 were then iteratively updated with the new values of $\mathbf{y}$ calculated by eq. 3 substituted into eqs. 1 and 3 to recursively calculate the neural activations. This process was terminated after 50 iterations. After 50 iterations, values of $\mathbf{y}$ less than 0.001 were set to zero. To perform simulations with a hierarchical model the steady-state responses for the first processing-stage were determined. The first-stage prediction neuron responses were then provided as input to the second processing-stage, and equations eqs. 1 to 3 applied to the second processing-stage to determine its response[c].

The values of $\mathbf{y}$ represent predictions of the causes underlying the inputs to the network. The values of $\mathbf{r}$ represent the expected inputs given the predicted causes. The values of $\mathbf{e}$ represent the discrepancy (or residual error) between the reconstruction, $\mathbf{r}$, and the actual input, $\mathbf{x}$. The full range of possible causes that the network can represent are defined by the weights, $\mathbf{W}$ (and $\mathbf{V}$). Each row of $\mathbf{W}$ (which correspond to the weights targeting an individual prediction neuron, *i.e.*, its RF) can be thought of as a "dictionary element", or "basis vector" or "elementary component" or "preferred stimulus", and $\mathbf{W}$ as a whole can be thought of as a "dictionary" or "codebook" of possible representations, or a model of the external environment. The activation dynamics, described by eqs. 1, 2 and 3, perform gradient descent on the reconstruction error in order to find prediction neuron activations that accurately reconstruct the input (Achler, 2014; Spratling, 2012; Spratling et al., 2009). Specifically, the equations operate to find values for $\mathbf{y}$ that minimise the Kullback-Leibler (KL) divergence between the input ($\mathbf{x}$) and the reconstruction of the input ($\mathbf{r}$) (Solbakken and Junge, 2011; Spratling et al., 2009). The activation dynamics thus result in the PC/BC-DIM algorithm selecting a subset of active prediction neurons whose RFs (which correspond to dictionary elements) best explain the underlying causes of the sensory input. The strength of activation reflects the strength with which each dictionary element is required to be present in order to accurately reconstruct the input. This strength of response also reflects the probability with which that dictionary element (the preferred stimulus of the active prediction neuron) is believed to be present, taking into account the evidence provided by the input signal and the full range of alternative explanations encoded in the RFs of the whole population of prediction neurons.

Compared to some earlier implementations of the PC/BC-DIM model, the algorithm described here differs in the following respects:

1. the calculation of the reconstruction error (in eq. 2) is performed using $\mathtt{max}(\epsilon_2, \mathbf{r})$ rather than $\epsilon_2 + \mathbf{r}$.

2. the calculation of the prediction neuron responses (in eq. 3) uses $\mathtt{max}(\epsilon_1, \mathbf{y})$ rather than $\epsilon_1 + \mathbf{y}$.

[c] Determining, sequentially, the steady-state responses for each processing stage was necessary in order to make the proposed model tractable given the available computational resources (a Core i7-4790K desktop PC with 16GB RAM). A more biologically-plausible model would iterate eqs. 1 to 3 for both processing stages simultaneously, with the prediction neuron response calculated for the first-stage at each iteration provided as input the second processing stage before the next iteration. In such an implementation, it would also be possible to explore the effects of inter-stage feedback connections from the second to the first processing stage. In the current, more tractable implementation, such connections would have no effect as the first stage has finished processing by the time the second stage starts. However, psychophysical experiments showing that image classification can be determined very rapidly in humans and monkeys (DiCarlo et al., 2012; Fabre-Thorpe et al., 2001; Hochstein and Ahissar, 2002; Keysers et al., 2001; Oliva and Torralba, 2006; VanRullen and Thorpe, 2001) suggest that cortical feedback connections (which would be modelled by inter-stage feedback) have little influence on object recognition (in unambiguous cases). The lack of inter-stage feedback connections in the current model also allows more direct comparison to other neural model of object recognition that contain only feedforward connections (*e.g.*, Bengio, 2009; Bengio et al., 2013; Ciresan et al., 2012; Fukushima, 1980, 1988, 2005; Hamidi and Borji, 2010; Hinton and Salakhutdinov, 2006; Hinton et al., 2006; Jarrett et al., 2009; Krizhevsky et al., 2012; LeCun and Bengio, 1995; LeCun et al., 1998, 2010; Mutch and Lowe, 2008; Riesenhuber and Poggio, 1999; Serre et al., 2007; Spratling, 2016c; Theriault et al., 2013; Thorpe et al., 2004; Vincent et al., 2010; Wallis and Rolls, 1997).

3. the value of $\epsilon_1$ is a function of the sum of the feedback weights targeting the reconstruction neurons rather than a fixed value (such as $1 \times 10^{-5}$).

These changes help PC/BC-DIM to scale-up to very large networks of neurons. Specifically, for a very large population of prediction neurons, adding $\epsilon_1$ to each prediction neuron response (even when $\epsilon_1$ is very small) will cause the responses of the reconstruction neurons to be elevated, and the error neurons responses to be suppressed, which will in turn effect the prediction neuron responses. The second change above reduces this effect of $\epsilon_1$ on the neural responses. The first and third changes allow $\epsilon_1$ to be given the largest value possible (which speeds-up convergence to the steady-state) while preventing $\epsilon_1$ from effecting the responses.

In addition, in some earlier implementations of the PC/BC-DIM model, the reconstruction has been used purely as a means to calculate the errors, and hence, eqs. 1 and 2 have been combined into a single equation. Here, the underlying mathematical model is identical to that used in previous work, but the interpretation has changed in order to consider the reconstruction to be represented by a separate neural population. This change, therefore, has no effect on the current results. However, other recent results have shown that a separate neural population encoding the reconstruction can perform a useful computational role (Muhammad and Spratling, 2015; Spratling, 2016b,d).

### 2.4 Code

Open-source software, written in MATLAB, which performs all the experiments described in this article is available for download from: `http://www.corinet.org/mike/Code/pcbc_image_recognition.zip`.

## 3 Results and Discussion

### 3.1 Handwritten Digit Recognition and Comparison with Deep Learning

To test the ability of the proposed method to categorise images with tolerance to within-class variation it was applied to the MNIST hand-written digits dataset[d]. This datset consists of 28-by-28 pixel grayscale images of isolated digits. The training set contains 60000 images and the test set contains 10000 images. For this task the following parameters were used: the similarity threshold for the clustering performed on the image patches was set equal to $\kappa = 0.85$; the threshold on the number of patches in each cluster was set equal to $\lambda = 0$; and the standard deviation of the Gaussian used to pre-process both the images and RFs of the first processing-stage was set equal to $\sigma = 4$ pixels. After pre-processing, each individual input image was rescaled to fill the range $[0, 1]$. The training procedure for the first processing stage (see sect. 2.1) produced a dictionary containing 35956 elements. Examples of these dictionary elements are shown in fig. 2a.

This dictionary was used to define the weights for 35956 prediction neurons in the first processing stage (see sect. 2.1). As there were 10 classes, the second processing stage contained 10 prediction neurons. The responses of the first and second stage prediction neurons to two test images are shown in fig. 2c and d. When tested on all images from the test set, it was found that 2.19% of these images were mis-classified. Examples of incorrectly classified test images are shown in fig. 2b. The classification error of the proposed method is compared to those of a variety of other algorithms in tab. 1. It can be seen that while the results of the proposed method are good, they fall far short of the current state-of-the-art.

Most of these state-of-the-art algorithms are deep hierarchical neural networks. Deep architectures can be subdivided into two main types: (1) stacked generative models, such as deep belief networks (Hinton and Salakhutdinov, 2006; Hinton et al., 2006), and stacked autoencoders (Bengio, 2009; Bengio et al., 2013; Vincent et al., 2010); and (2) discriminative models with alternating layers of feature detection and pooling, such as convolutional neural networks (CNN; Ciresan et al., 2012; Jarrett et al., 2009; Krizhevsky et al., 2012; LeCun and Bengio, 1995; LeCun et al., 1998, 2010), HMAX (Hamidi and Borji, 2010; Mutch and Lowe, 2008; Riesenhuber and Poggio, 1999; Serre et al., 2007; Theriault et al., 2013), and Neocognitron (Fukushima, 1980, 1988, 2005).

In common with architectures of the first type, the proposed algorithm also employs a hierarchy of generative models. However, the generative models are implemented using a different algorithm: PC/BC-DIM. Furthermore, PC/BC-DIM employs the generative model during inference: the generative model is used to make predictions of the expected sensory inputs, and through the iterative activation dynamics described by eqs. 1 to 3, determine the prediction neuron activations that minimise the discrepancy between the predicted and actual inputs. In contrast, autoencoders and restricted Boltzmann machines (RBM; Hinton, 2002; Teh et al., 2003) which are the building blocks of previous architectures of the first type, only employ the generative model during learning. Once the

[d] `http://yann.lecun.com/exdb/mnist/`

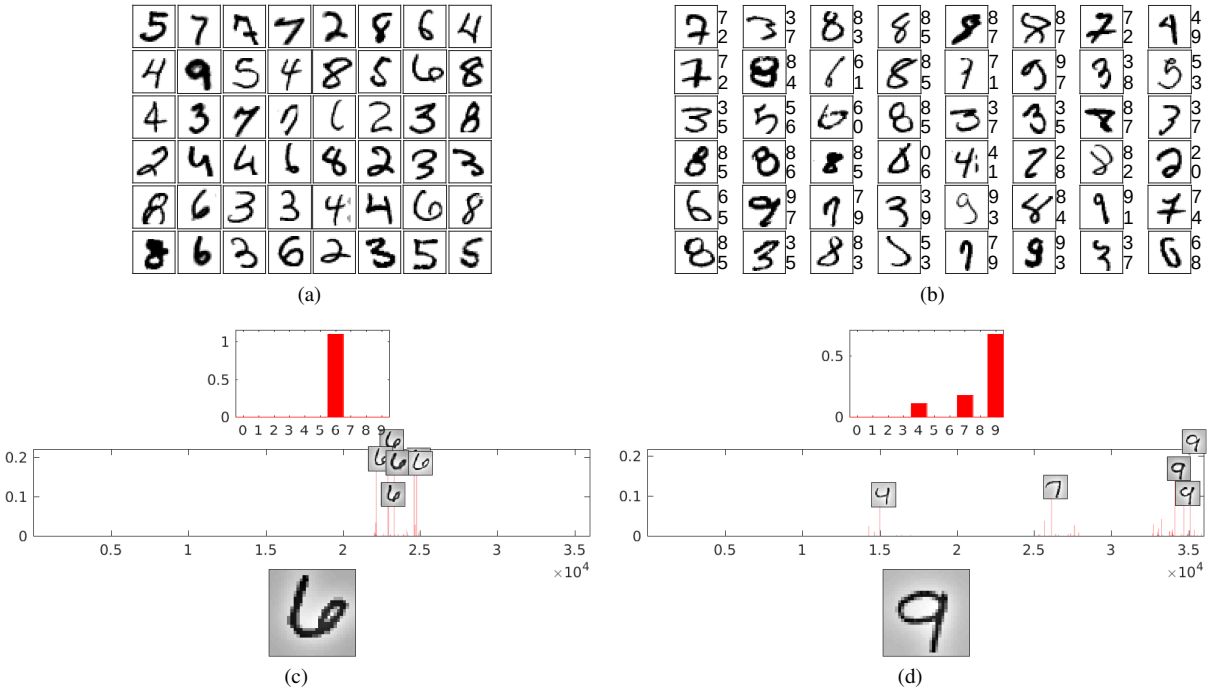



**Figure 2:** Results for the MNIST dataset. (a) Exemplars from the dictionary learnt from image patches. (b) Exemplars of mis-classified images from the test set. There are two numbers to the right of each image. The lower number is the class predicted by the PC/BC-DIM network. The top number is the true class of the image. (c) and (d) show the responses of the prediction neurons to two images from the test set. Responses are shown as histograms where the x-axis is neuron number, and the y-axis is activation level (in arbitrary units). The bottom panel is the input to the PC/BC-DIM network. The middle panel shows the response of the prediction neurons in the first processing stage. The RFs of the most active prediction neurons are indicated by the images superimposed on the histogram. The top panel shows the response of the prediction neurons in the second processing stage.

weights have been set to allow these models to reconstruct the input, new inputs are processed using the feedforward weights only.

In common with architectures of the second type, the proposed algorithm has alternate processing stages that specialise in creating more discriminate representations in one layer, and more invariant representations in the next layer. This is achieved by defining the weights differently, but by applying the same algorithm to determine the neural activations during inference. In contrast, existing architectures of the second type use completely different mathematical operations to perform these two functions. For example, more specialised representations are often created by applying a linear filtering operation, while more tolerant representations are usually formed by finding the maximum response within a sub-population of pre-synaptic neurons. The proposed model is thus simpler, in that it only requires one type of processing stage.

Another difference between the proposed architecture and deep architectures of both type 1 and 2, is that in the proposed model classification is performed by the last processing stage of the PC/BC-DIM hierarchy. In contrast, most existing deep architectures are used only as a method of feature extraction (Bengio et al., 2013) to provide input to a distinct classification algorithm, such as a support vector machine (SVM) or a logistic regression classifier. The proposed model is thus simpler, in that it integrates feature extraction and classification within a single homogeneous framework, rather than using different methods for each.

However, as illustrated by the results in tab. 1 deep architectures have an advantage in terms of classification accuracy. There are many reasons for this. Firstly, it is known that the deeper the architecture, the better the performance (He et al., 2016). The proposed architecture is very shallow compared to most deep architectures. Creating deeper PC/BC-DIM hierarchies by stacking more processing-stages, might thus allow better performance, and potentially create a better model of the ventral pathway. However, doing so will require more sophisticated methods of defining the weights in those processing stages. The current model uses an unsupervised learning method. In contrast, much of the success deep architectures derives from using supervised learning. Using more training data is also known to generally improve performance. One way to generate additional training data is to generate

| Method | MNIST |
|---|---|
| hierarchical PC/BC-DIM | 2.19 |
| SVM (Yu et al., 2009) | 12.0 |
| MO-SFL (Gong et al., 2015) | 6.55 |
| ICA+ELM (Zhang et al., 2014) | 5.6 |
| spiking NN + unsupervised learning (Diehl and Cook, 2015) | 5.0 |
| spiking S2M + Event-driven CD (Neftci et al., 2016) | 4.4 |
| PC/BC-DIM no pre-processing, classification via linear readout (Spratling, 2014a) | 4.1 |
| Nearest Neighbour | 2.77 |
| spiking DBN (O'Connor et al., 2013) | 2.52 |
| PC/BC-DIM no pre-processing, classification via sub-dictionary error (Spratling, 2014a) | 2.19 |
| task-driven PSD (Lv et al., 2016) | 1.98 |
| DBN+SVM (Yu et al., 2009) | 1.9 |
| CNN (LeNet-1) (LeCun et al., 1995) | 1.7 |
| Sprase Coding (Sprechmann and Sapiro, 2010) | 1.26 |
| DBN (Hinton et al., 2006) | 1.25 |
| Stacked RBM (Larochelle et al., 2009) | 1.2 |
| Deep Sparse rectifier Neural Network (Glorot et al., 2011) | 1.16 |
| CNN (LeNet-4) (LeCun et al., 1995) | 1.1 |
| SDL-G (Mairal et al., 2008) | 1.05 |
| Deep Boltzmann Machine (Salakhutdinov and Hinton, 2012) | 0.95 |
| CNN (LeNet-5) (LeCun et al., 1995) | 0.9 |
| sparse-HMAX+SVM (MTC) (Cardoso and Wichert, 2013) | 0.71 |
| locally shift invariant sparse hierarchical features (Ranzato et al., 2007) | 0.64 |
| Task-driven dictionary learning (Mairal et al., 2012) | 0.54 |
| CNN (PSD) (Jarrett et al., 2009) | 0.53 |
| Multi-column deep neural network (Ciresan et al., 2010) | 0.35 |
| MCDNN (Ciresan et al., 2012) | 0.23 |

**Table 1:** Percentage classification error of various methods on the MNIST hand-written digits dataset.

images that are affine deformations of the original training images. This can result in a significant improvement in performance. For example, Ciresan et al. (2010) report an error rate of 0.35% on MNIST with deformation, and 1.47% without[e]. Expanding the dataset in this way could also be used to potentially improve the performance of the proposed PC/BC-DIM architecture. State-of-the-art performance on many classification tasks has been generated using an ensemble of deep architectures (Ciresan et al., 2012): where multiple, different, deep networks are used to independently classify the input, and the final classification is a combination of these individual classifications. If classification accuracy, rather than biological-plausibility, were the main motivation then using the current architecture as the building block for an ensemble might also be considered.

## 3.2 Face Recognition and Comparison with Sparse Coding

To test the ability of the proposed method to perform sub-ordinate level categorization (*i.e.*, identification) with tolerance to illumination it was applied to the cropped and aligned version of the Extended Yale Face Database B[f] (Georghiades et al., 2001; Lee et al., 2005). This dataset consists of 168-by-192 pixel grayscale images of faces taken from a fixed viewpoint in front of the face under varying lighting conditions. There are approximately 64 images for each of 38 individuals. Following the method used in previous work with this dataset (Jiang et al., 2011, 2013; Wright et al., 2009; Zhang et al., 2011; Zhang and Li, 2010), half the images for each class were used for training and the other half for testing.

In previous work, classification has been performed using images down-sampled to 21-by-24 pixels (or fewer). This has been necessary as previous methods have used pre-processing steps (such as the calculation of Eigenfaces and Laplacian-faces) that are too memory intensive to be performed on larger images (Wright et al., 2009). To allow a direct comparison with this previous work results are presented for the proposed method using images

[e] http://people.idsia.ch/~ciresan/results.htm
[f] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

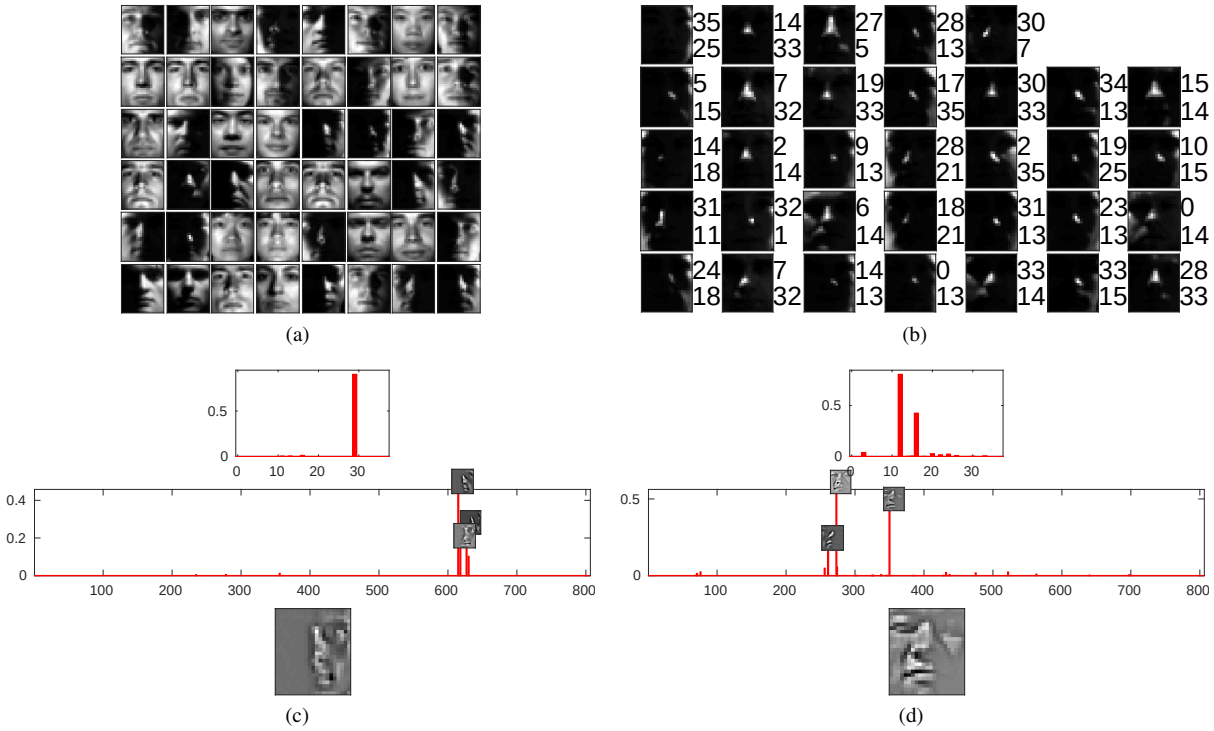



**Figure 3:** Results for the Extended Yale Face Database B, when using 21-by-24 pixel images. (a) Exemplars from the dictionary learnt from image patches. (b) All of mis-classified images from the test set. There are two numbers to the right of each image. The lower number is the class predicted by the PC/BC-DIM network. The top number is the true class of the image. (c) and (d) show the responses of the prediction neurons to two images from the test set. The bottom panel is the input to the PC/BC-DIM network. The middle panel shows the response of the prediction neurons in the first processing stage. The RFs of the most active prediction neurons are indicated by the images superimposed on the histogram. The top panel shows the response of the prediction neurons in the second processing stage.

that have also been resized by a scale factor $\delta = \frac{1}{8}$ to 21-by-24. However, as the proposed method can work successfully with larger images, results are also presented for images at the original size (*i.e.*, for $\delta = 1$).

For this task the following parameters were used: the similarity threshold for the clustering performed on the image patches was set equal to $\kappa = 0.9$; the threshold on the number of patches in each cluster was set equal to $\lambda = 0$; and the standard deviation of the Gaussian used to pre-process both the images and the RFs of the first processing-stage was set equal to $\sigma = 2.5\sqrt{\delta}$ pixels. After pre-processing, each individual input image was rescaled to fill the range $[0, 1]$. For the 21-by-24 pixel images, the training procedure for the first processing stage (see sect. 2.1) produced a dictionary containing 806 elements. Examples of these dictionary elements are shown in fig. 3a. This dictionary was used to define the weights for 806 prediction neurons in the first processing stage (see sect. 2.1). As there were 38 individuals, the second processing stage contained 38 prediction neurons. The responses of the first and second stage prediction neurons to two test images are shown in fig. 3c and d. The incorrectly identified test images, for the 21-by-24 pixel version of this task, are shown in fig. 3b. It can be seen that all the mis-classified images were taken under very poor lighting conditions.

The classification error of the proposed method is compared to those of a variety of other algorithms in tab. 2. It can be seen that the performance of the proposed method is competitive with the current state-of-the-art for this task. The current state-of-the-art algorithms are based on sparse coding. These algorithms represent the image using a sparse set of elements selected from an overcomplete dictionary. They then perform classification by analysing the reconstruction errors produced by dictionary elements associated with different classes (Spratling, 2014a; Sprechmann and Sapiro, 2010; Wright et al., 2009; Zhang and Li, 2010). In common with these algorithms, PC/BC-DIM also represents the input images using a sparse code (examples can be seen in the lower histograms in fig. 3c and d, where it can be seen that only a very small subset of the first stage prediction neurons are active). However, in contrast to most existing sparse dictionary-based classifiers, the proposed method makes the classification using the sparse code (the prediction neuron responses) rather than the reconstruction error (the error neuron responses). This latter method is more biologically-plausible, but less accurate (Spratling, 2014a). It

| Method | YALE (21x24) | YALE (168x192) |
|---|---|---|
| hierarchical PC/BC-DIM | 2.7 | 0.5 |
| Nearest Neighbour (Wright et al., 2009) | 9.3 | |
| D-KSVD (Zhang and Li, 2010) | 4.4 | |
| LC-KSVD2 (Jiang et al., 2011, 2013) | 3.3 | |
| Laplacianfaces+SVM (Wright et al., 2009) | 2.3 | |
| SRC (Wright et al., 2009) | 1.9 | |

**Table 2:** Percentage classification error of various methods on the Extended Yale Face Database B.

has been found that the performance of sparse dictionary-based classifiers is improved by the supervised learning of more discriminative dictionaries (Chiang et al., 2013; Jiang et al., 2013; Mairal et al., 2012; Sprechmann and Sapiro, 2010; Yang et al., 2011; Zhang et al., 2013). Such learning might potentially also improve the performance of the proposed algorithm.

### 3.3 Car Recognition and Comparison with Generalised Hough Transform

To test the ability of the proposed method to localise and recognise objects in natural images with tolerance to position, illumination, size, partial occlusion, and within-category shape variation it was applied to the UIUC cars dataset (Agarwal et al., 2004; Agarwal and Roth, 2002)[g]. This dataset consists of greyscale images of outdoor scenes. The training set consists of 550 car images and 500 images that do not contain cars. There are two sub-tasks: recognising side views of cars at a single scale (the location and number of cars varies between test images), and recognising side views of cars across multiple scales (the size, location and number of cars varies between test images). For the single-scale task the test set contains 170 images containing 200 side views of cars. The multi-scale task has a test set of 108 images containing 139 cars.

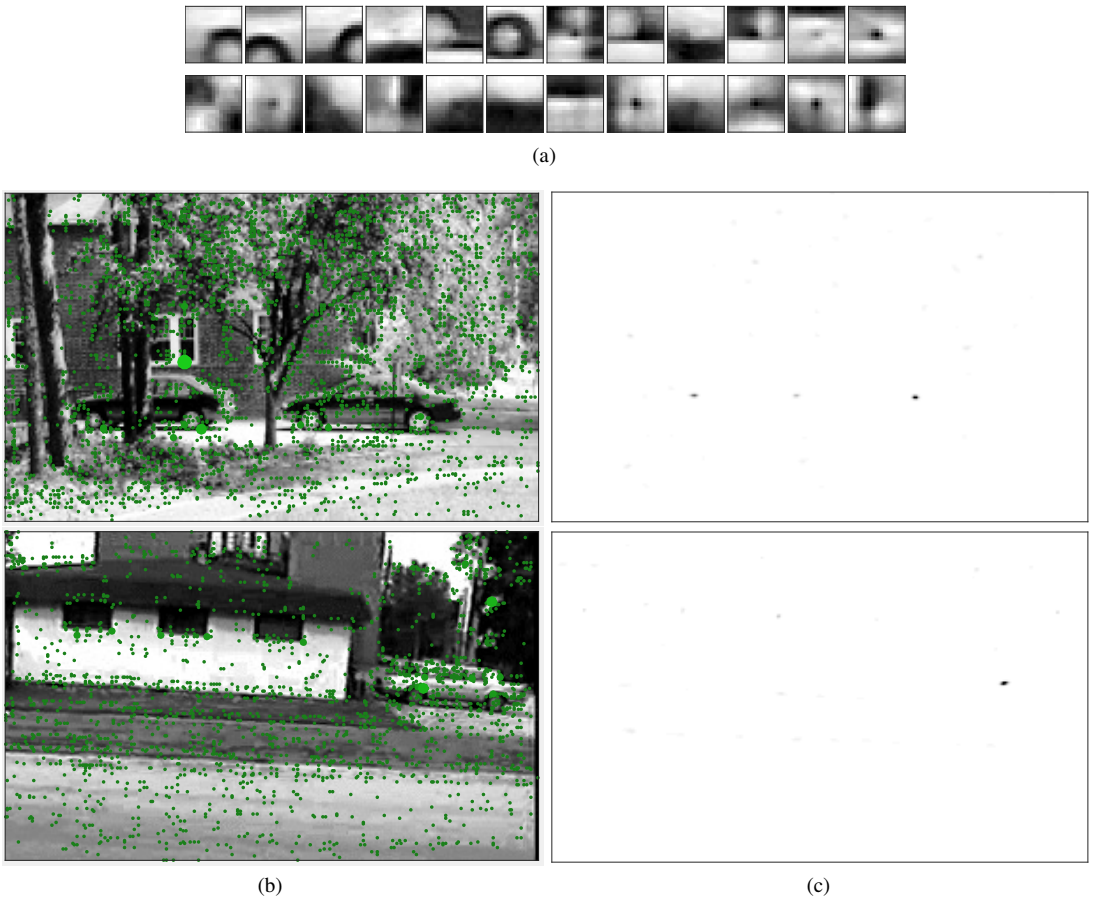
The same training set, and the same parameter values, were used for both sub-tasks. Specifically, the similarity threshold for the clustering performed on the image patches was set equal to $\kappa = 0.4$, the threshold on the number of patches in each cluster was set equal to $\lambda = 12$, and the standard deviation of the Gaussian used to pre-process both the images and the RFs of the first processing-stage was set equal to $\sigma = 3.5$ pixels. Training of the dictionary used to define the weights of the first processing-stage was performed on 15-by-15 pixel patches extracted from the training images around keypoints located using the Harris corner detector. For the single-scale task the patches taken from the car images were clustered into 273 dictionary elements. The non-car image patches were clustered into 140 dictionary elements. Examples, of these first-stage dictionary elements are shown in fig. 4a. These dictionary elements were used to define the RFs of the prediction neurons in the first PC/BC-DIM processing stage, resulting in 413 prediction neurons at each pixel location in the input image. For the multi-scale task training was performed on the 1050 car and non-car training images resized to six different scales. The dictionary consisted of 2465 elements representing non-car parts and 3601 elements representing car parts, resulting in 6066 first-stage prediction neurons at each pixel location.

Figure 4b shows two example test images for the single-scale task on which have been superimposed dots to show locations where there is a strong response from the sub-population of first processing-stage prediction neurons that represent car parts. The size of the dot is proportional to the magnitude of the response of the prediction neuron. For prediction neurons whose RFs were defined using the same dictionary element, non-maximum suppression was performed over those prediction neuron responses, so that all response other than the local maximum were set to zero.

For the single-scale task, the number of second-stage prediction neurons was equal to the number of pixels in the input image. Each second-stage prediction neuron had the same weights (but at spatially sifted positions), equal to the summed response of all the first-stage prediction neurons to all the car images in the training set. However, to improve tolerance to position, these weights were smoothed across space by convolving them with a two-dimensional circular symmetric Gaussian function with a standard deviation of two pixels. Figure 4c shows the responses of all the second-stage prediction neurons for the two images shown in fig. 4b. For the multi-scale task the second processing-stage consisted of six sub-populations of prediction neurons (one for each scale), each sub-population contained one prediction neuron for each pixel in the test image. In this case the weights were smoothed across space and scale using a three-dimensional Gaussian function.

To determine the location of cars predicted by the proposed method, the spatial distribution of prediction

[g] https://cogcomp.cs.illinois.edu/Data/Car/



(a) (b) (c)

**Figure 4:** (a) A small sample of the dictionary elements represented by the first-stage prediction neurons. The top row shows RFs of prediction neurons trained on patches taken from the car images. The second row shows RFs of prediction neurons trained on patches taken from the non-car images. (b) Two example test images from the UIUC single-scale cars dataset (Agarwal et al., 2004; Agarwal and Roth, 2002). The green dots show the locations where dictionary elements representing car parts have been matched to the image: the size of the dot is proportional to the strength of the response of the corresponding first-stage prediction neuron. (c) The response of all the second-stage prediction neurons to the corresponding example test image shown in (b). The response is indicated by the grayscale, with white corresponding to no response and black corresponding to a high response. It can be seen that the strongest responses correspond to the centres of the cars.

neuron responses (as illustrated in fig. 4c) was analysed to find the coordinates of spatially-contiguous regions of strong activity. Such a region was defined as a contiguous neighbourhood in which each neuron had an activity of more than 0.001, and which was completely surrounded by neurons with a response of 0.001 or less. The coordinates represented by such a region were then determined using population vector decoding (Georgopoulos et al., 1986). This simply calculates the average of the coordinates represented by the neurons in the region, weighted by each neuron's response. For the multi-scale task, the coordinates of regions of high activity were determined in the same way, but in a three-dimensional space (position and scale). The total sum of the response in each region was also recorded.

To quantitatively assess the performance of the proposed algorithm the procedures advocated in Agarwal and Roth (2002) were followed. Specifically, for each region with a total response exceeding a threshold, the location (and scale) represented by that region were determined (as described in the preceding paragraph) and these values were compared to the true location (and scale) of each car provided in the ground-truth data. The comparison was performed using the java code supplied with UIUC cars data set. If the predicted parameter values were sufficiently close to the ground-truth, this was counted as a true-positive. If multiple regions of high activity corresponded to the same ground-truth parameters, only one match was counted as a true-positive, and the rest were counted as false-positives. All other regions of high activity that failed to match the ground-truth data were also counted as false-positives. Ground-truth parameters for which there was no corresponding values found by

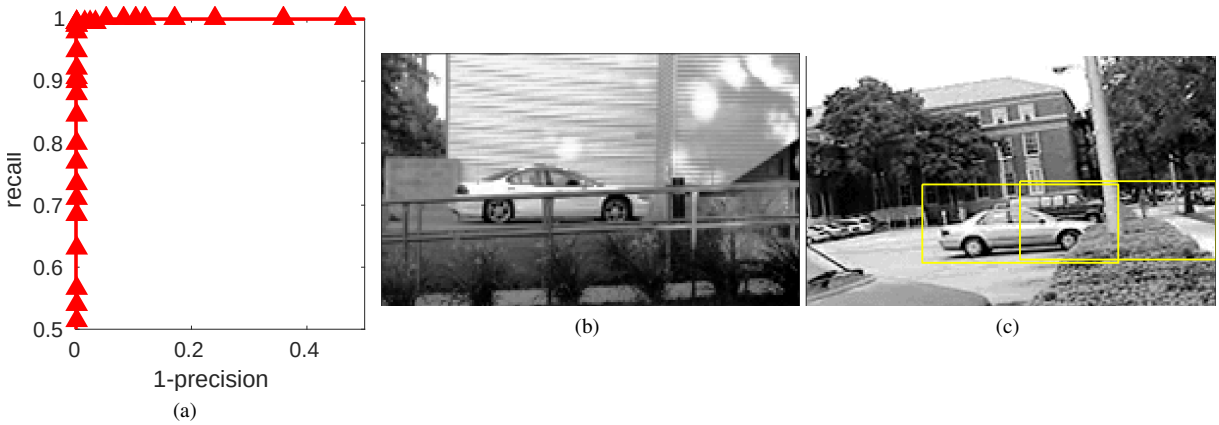



**Figure 5:** Results of applying the proposed method to the single-scale UIUC cars dataset. (a) recall versus 1-precision. At the threshold for equal error rate there were two images in which there were errors. (b) The only false negative. (c) The only false positive. The bounding boxes, shown in yellow, indicate locations in which cars were detected by the proposed algorithm.

| Method | UIUC-single | UIUC-multi |
|---|---|---|
| hierarchical PC/BC-DIM | 0.5 | 2.9 |
| ISM (Leibe et al., 2008) | 9 | - |
| ISM+MDL verification (Leibe et al., 2008) | 2.5 | 5 |
| Hough Forest (Gall and Lempitsky, 2009; Gall et al., 2011) | 1.5 | 2.4 |
| Discriminative HT (Okada, 2009) | 1.5 | - |
| ESS (Lampert et al., 2008) | 1.5 | 1.4 |
| keypoint patch matching+PC/BC-DIM voting (Spratling, 2016c) | 1 | 3.6 |
| chains model (Karlinsky et al., 2010) | 0.5 | - |
| sliding window HMAX+verification (Mutch and Lowe, 2006) | 0.06 | 9.4 |
| IHRF (Lin et al., 2014) | 0 | 1.3 |
| PRISM (Lehmann et al., 2011) | - | 2.2 |

**Table 3:** Percentage EER of various methods on the UIUC single-scale and multi-scale cars dataset.

the proposed method were counted as false-negatives. The total number of true-positives ($TP$), the number of false-positives ($FP$), and the number of false-negatives ($FN$) were recorded over all test images, and were used to calculate recall ($\frac{TP}{TP+FN}$) and precision ($\frac{TP}{TP+FP}$). By varying the threshold applied to select regions of high activity, precision-recall curves were plotted to show how detection accuracy varied with threshold. To summarise performance, the f-score ($= \frac{2.recall.precision}{recall+precision} = \frac{2TP}{2TP+FP+FN}$) which measures the trade-off between precision and recall, was calculated at the threshold that gave the highest value. In addition, to allow comparison with previously published results, the equal error rate (EER) was also found. This is the percentage error when the threshold is set such that the number of false-positives equals the number of false-negatives.

The precision recall curve obtained on the UIUC single-scale cars dataset is shown in fig. 5. The f-score was 0.9975 and the EER was 0.5%. Figure 5b and c show the only two images in the test set on which the proposed method makes a mistake at the threshold for equal error rate. The results obtained on the UIUC multi-scale cars dataset are shown in fig. 6. In this case, the f-score was 0.9718 and the EER was 2.9%. These results are compared to those of other published methods in tab. 3. It can be seen that the proposed method is competitive with the state-of-the art, and particularly, that it outperforms the method described in Spratling (2016c). That method is similar to the one proposed here, except that the first processing-stage described here was replaced by a process that found keypoints in the image, and matched (using the ZMNCC as the similarity metric) the image patches around these keypoints to elements in the dictionary. Hence, the method proposed here is simpler, in that both stages are implemented using PC/BC-DIM, rather than being implemented in completely different ways.

The algorithm described in Spratling (2016c) was inspired by the implicit shape model (ISM; Leibe et al., 2008), which employs the generalised Hough transform (Ballard, 1981; Duda and Hart, 1972; Hough, 1962) to allow dictionary elements that match features in the image to cast votes for the possible location and scale of


**Figure 6:** Results of applying the proposed method to the multi-scale UIUC cars dataset. (a) recall versus 1-precision. At the threshold for equal error rate there were seven images in which there were errors. These images are shown in (b)-(h) with bounding boxes, in yellow, indicating locations in which cars were detected by the proposed algorithm. (b)-(d) Show the three images in which there were false negatives. (e) Shows the only image in which there was both a false negative and a false positive. Note that while both cars appear to have been recognised, one has not be located with sufficient accuracy. (f)-(h) Show the three images in which there were false positives. Note that the last image has been flagged as containing a false-positive as the left-most car is not included as a true-positive in the ground-truth data.

the to-be-recognised object. Once all the votes have been cast, ISM uses a Minimum Description Length (MDL) criteria to reject false peaks caused by votes that come from image elements which have also voted for other peaks that are more likely to be the true ones. The second processing-stage in the proposed model can also be thought of as implementing the voting process of the generalised Hough transform, but using explaining away (rather than MDL) to suppress false peaks (Spratling, 2016c). In a previous section the function of the second processing-stage was described as being analogous to the function of the pooling stages in deep neural networks. There is therefore also an analogy between the Hough transform and pooling. Both attempt to allow recognition with tolerance to location, but the Hough transform is both less constrained and less arbitrary than the pooling used in deep networks.

# 4 Conclusions

The current work provides an initial proof-of-concept demonstration that predictive coding can perform object recognition in natural images. Hence, it provides concrete support for previous speculation about the possible role of predictive coding in perceptual inference. Object recognition is a complex task that requires being able to

distinguish one individual or class of object from other individuals or classes while being able tolerate changes in the appearance of the to-be-recognised object from one image to another. The results presented here show that PC/BC-DIM can recognise individuals and classes, and that it can do so with tolerance to position, illumination, size, partial occlusion, and within-category shape variation. The experiments used here have not addressed tolerance to non-rigid shape deformations, or rotations.

As discussed in sect. 3, the proposed model has strong similarity to existing methods like deep neural networks, ISM, and sparse dictionary-based classification. These previous methods tend to make use of different mechanisms to perform different sub-tasks. For example, deep networks use different mechanisms for feature detection, pooling, and classification, while ISM uses different mechanisms for detecting image features and counting votes. In contrast, the proposed model uses the same mechanism (PC/BC-DIM) to perform each of these sub-tasks.

Improving the performance of the proposed method on the tasks used here, or extending it to more complex object recognition tasks that require tolerance to a greater range of image transformations of the recognition of a wider range of objects, or developing it into a model of ventral stream processing, is likely to require the building of deeper and more complex networks. Defining appropriate weights for such networks is the key to their success. In the current article the weights have been set in a rather ad-hoc and non-biologically plausible way. This is sufficient for a proof-of-concept demonstration, but would need to be addressed in future work.

# References


Achler, T. (2014). Symbolic neural networks for cognitive capacities. *Biologically Inspired Cognitive Architectures*, 9(0):71–81.

Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–90.

Agarwal, S. and Roth, D. (2002). Learning a sparse representation for object detection. In *Proceedings of the European Conference on Computer Vision*, volume IV, pages 113–30.

Anselmi, F., Leibo, J., Rosasco, L., Mutch, J., Tacchetti, A., and Poggio, T. (2014). Unsupervised learning of invariant representations with low sample complexity: the magic of sensory cortex or a new framework for machine learning? CBMM Memo 001, Center for Brains Minds and Machines, Massachusetts Institute of Technology.

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–22.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–828.

Cardoso, A. and Wichert, A. (2013). Handwritten digit recognition using biologically inspired features. *Neurocomputing*, 99:575–80.

Chiang, C.-K., Liu, C.-H., Duan, C.-H., and Lai, S.-H. (2013). Learning component-level sparse representation for image and video categorization. *IEEE Transactions on Image Processing*, 22(12):4775–87.

Ciresan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–20.

Ciresan, D. C., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(03):181–204.

DiCarlo, J. J. and Cox, D. D. (2007). Untangling invariant object recognition. *Trends in Cognitive Sciences*, 11(8):333–41.

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415–34.

Diehl, P. and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9:99.

Duda, R. O. and Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–5.

Fabre-Thorpe, M., Delorme, A., Marlot, C., and Thorpe, S. (2001). A limit to the speed of processing in ultra-rapid visual categorization of novel natural scenes. *Journal of Cognitive Neuroscience*, 13:171–80.

Friston, K. and Kiebel, S. (2009). Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 364:1211–21.

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Fukushima, K. (1988). Neocognitron: a hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119–30.

Fukushima, K. (2005). Restoring partly occluded patterns: a neural network model. *Neural Networks*, 18(1):33–43.

Gall, J. and Lempitsky, V. (2009). Class-specific Hough forests for object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Gall, J., Yao, A., Razavi, N., Van Gool, L., and Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2188–202.

Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–60.

Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, 233:1416–9.

Gilbert, C. D. (1996). Plasticity in visual perception and physiology. *Current Opinion in Neurobiology*, 6(2):269–74.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*.

Gong, M., Liu, J., Li, H., Cai, Q., and Su, L. (2015). A multiobjective sparse feature learning model for deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 26(12):3263–3277.

Goodale, M. A. and Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, 15:20–5.

Hamidi, M. and Borji, A. (2010). Invariance analysis of modified C2 features: case studyhandwritten digit recognition. *Machine Vision and Applications*, 21(6):969–79.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Hinton, G. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–7.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1711–1800.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–54.

Hochstein, S. and Ahissar, M. (2002). View from the top: hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804.

Hough, P. V. C. (1962). Method and means for recognizing complex patterns. U.S. Patent 3 069 654.

Huang, Y. and Rao, R. P. N. (2011). Predictive coding. *WIREs Cognitive Science*, 2:580–93.

Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *Proceedings of the International Conference on Computer Vision*, pages 2146–53. IEEE.

Jiang, Z., Lin, Z., and Davis, L. S. (2011). Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Jiang, Z., Lin, Z., and Davis, L. S. (2013). Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–64.

Karlinsky, L., Dinerstein, M., Daniel, H., and Ullman, S. (2010). The chains model for detecting parts by their context. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Kersten, D., Mamassian, P., and Yuille, A. (2004). Object perception as Bayesian inference. *Annual Review of Psychology*, 55(1):271–304.

Keysers, C., Xiao, D. K., Földiák, P., and Perrett, D. I. (2001). The speed of sight. *Journal of Cognitive Neuroscience*, 13(1):90–101.

Kobatake, E. and Tanaka, K. (1994). Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. *Journal of Neurophysiology*, 71(3):856–67.

Kok, P. and de Lange, P. F. (2015). Predictive coding in sensory cortex. In Forstmann, U. B. and Wagenmakers, E.-J., editors, *An Introduction to Model-Based Cognitive Neuroscience*, pages 221–44. Springer, New York, NY.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–105. Curran Associates, Inc.

Krüger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., Rodrguez-Sánchez, A. J., and Wiskott, L. (2013). Deep hierarchies in the primate visual cortex: what can we learn for computer vision? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1847–71.

Lampert, C., Blaschko, M., and Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. (2009). Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, 1:1–40.

LeCun, Y. and Bengio, Y. (1995). Convolutional networks for images, speech, and time-series. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–324.

LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cartes, C., Dencker, J., Drucker, H., Guyon, I., Müller, U., Säckinger, E., Simard, P., and Vapnik, V. (1995). Comparison of learning algorithms for handwritten digit recognition. In Fogelman, F. and Gallinari, P., editors, *Proceedings of the International Conference on Artificial Neural Networks*, pages 53–60. EC2 Cie Publishers, Paris, France.

LeCun, Y., Kavukvuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *Procedings of the International Symposium on Circuits and Systems (ISCAS10)*. IEEE.

Lee, K. C., Ho, J., and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–98.

Lehmann, A., Leibe, B., and Gool, L. V. (2011). Fast PRISM: Branch and bound Hough transform for object class detection. *International Journal of Computer Vision*, 94(2):175–197.

Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–89.

Lin, Y., Lu, N., Lou, X., Zou, F., Yao, Y., and Du, Z. (2014). Invariant Hough random ferns for object detection and tracking. *Mathematical Problems in Engineering*, 2014(513283):20.

Lochmann, T. and Deneve, S. (2011). Neural processing as causal inference. *Current Opinion in Neurobiology*, 21(5):774–81.

Lochmann, T., Ernst, U. A., and Denève, S. (2012). Perceptual inference predicts contextual modulations of sensory responses. *The Journal of Neuroscience*, 32(12):4179–95.

Logothetis, N. (1998). Object vision and visual awareness. *Current Opinion in Neurobiology*, 8(4):536–44.

Logothetis, N. K. and Pauls, J. (1995). Psychophysical and physiological evidence for viewer-centred object representations in the primate. *Cerebral Cortex*, 3:270–88.

Logothetis, N. K., Pauls, J., and Poggio, T. (1995). Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5:552–63.

Lv, L., Zhao, D., and Deng, Q. (2016). A semi-supervised predictive sparse decomposition based on task-driven dictionary learning. *Cognitive Computation*, pages 1–10.

Mairal, J., Bach, F., and Ponce, J. (2012). Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):791–804.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2008). Supervised dictionary learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 21, pages 1033–40. Curran Associates, Inc.

Mountcastle, V. B. (1998). *Perceptual Neuroscience: The Cerebral Cortex*. Harvard University Press, Cambridge, MA.

Muhammad, W. and Spratling, M. W. (2015). A neural model of binocular saccade planning and vergence control. *Adaptive Behavior*, 23(5):265–82.

Mutch, J. and Lowe, D. G. (2006). Multiclass object recognition with sparse, localized features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 11–18, New York, NY.

Mutch, J. and Lowe, D. G. (2008). Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, 80(1):45–57.

Neftci, E. O., Pedroni, B. U., Joshi, S., Al-Shedivat, M., and Cauwenberghs, G. (2016). Stochastic synapses enable efficient brain-inspired learning machines. *Frontiers in Neuroscience*, 10:241.

O'Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience*, 7:178.

Okada, R. (2009). Discriminative generalized Hough transform for object dectection. In *Proceedings of the International Conference on Computer Vision*, pages 2000–2005.

Oliva, A. and Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. In

Martinez-Conde, S., Macknik, S. L., Martinez, L. M., Alonso, J.-M., and Tse, P. U., editors, *Progress in Brain Research: Visual Perception*, volume 155, pages 23–36. Elsevier.

Oram, M. W. and Perrett, D. I. (1994). Modelling visual recognition from neurobiological constraints. *Neural Networks*, 7(6–7):945–72.

Pinto, N., Cox, D. D., and DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1).

Poggio, T., Anselmi, F., and Rosasco, L. (2015). I-theory on depth vs width: hierarchical function composition. CBMM Memo 041, Center for Brains Minds and Machines, Massachusetts Institute of Technology.

Ranzato, M. A., Huang, F. J., Boureau, Y., and LeCun, Y. (2007). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE Press.

Rao, R. P. N. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–25.

Rust, N. C. and Dicarlo, J. J. (2010). Selectivity and tolerance ('invariance') both increase as visual information propagates from cortical area V4 to IT. *The Journal of Neuroscience*, 30:12978–95.

Salakhutdinov, R. and Hinton, G. (2012). An efficient learning procedure for deep boltzmann machines. *Neural Computation*, 24(8):1967–2006.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–26.

Solbakken, L. L. and Junge, S. (2011). Online parts-based feature discovery using competitive activation neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1466–73.

Spratling, M. W. (2005). Learning viewpoint invariant perceptual representations from cluttered images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):753–61.

Spratling, M. W. (2008a). Predictive coding as a model of biased competition in visual selective attention. *Vision Research*, 48(12):1391–408.

Spratling, M. W. (2008b). Reconciling predictive coding and biased competition models of cortical function. *Frontiers in Computational Neuroscience*, 2(4):1–8.

Spratling, M. W. (2012). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. *Neural Computation*, 24(1):60–103.

Spratling, M. W. (2014a). Classification using sparse representations: a biologically plausible approach. *Biological Cybernetics*, 108(1):61–73.

Spratling, M. W. (2014b). Predictive coding. In Jaeger, D. and Jung, R., editors, *Encyclopedia of Computational Neuroscience*, pages 1–5. Springer, New York, NY.

Spratling, M. W. (2016a). Accurate and tolerant image patch matching using explaining away. *submitted*.

Spratling, M. W. (2016b). A neural implementation of Bayesian inference based on predictive coding. *Connection Science*, 28(4):346–83.

Spratling, M. W. (2016c). A neural implementation of the hough transform and the advantages of explaining away. *Image and Vision Computing*, 52:15–24.

Spratling, M. W. (2016d). Predictive coding as a model of cognition. *Cognitive Processing*, 17(3):279–305.

Spratling, M. W. (in press). A review of predictive coding algorithms. *Brain and Cognition*, in press.

Spratling, M. W., De Meyer, K., and Kompass, R. (2009). Unsupervised learning of overlapping image components using divisive input modulation. *Computational Intelligence and Neuroscience*, 2009(381457):1–19.

Sprechmann, P. and Sapiro, G. (2010). Dictionary learning and sparse coding for unsupervised clustering. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 2042–5.

Teh, Y. W., Welling, M., Osindero, S., and Hinton, G. E. (2003). Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–60.

Theriault, C., Thome, N., and Cord, M. (2013). Extended coding and pooling in the HMAX model. *IEEE Transactions on Image Processing*, 22(2):764–77.

Thorpe, S. J., Guyonneau, R., Guilbaud, N., Allegraud, J. M., and VanRullen, R. (2004). Spikenet: Real-time visual processing with one spike per neuron. *Neurocomputing*, 58-60:857–64.

Ungerleider, L. G. and Mishkin, M. (1982). Two cortical visual systems. In Ingle, D. J., Goodale, M. A., and Mansfield, R. J. W., editors, *Analysis of Visual Behavior*, pages 549–86. MIT Press, Cambridge, MA.

VanRullen, R. and Thorpe, S. J. (2001). Is it a bird? is it a plane? ultra-rapid visual categorisation of natural and artifactual objects. *Perception*, 30:655–68.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. (2010). Stacked denoising autoencoders: Leaning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning*

*Research*, 11:3371–408.

Wallis, G. and Bülthoff, H. (1999). Learning to recognize objects. *Trends in Cognitive Sciences*, 3(1):22–31.

Wallis, G. and Rolls, E. T. (1997). Invariant face and object recognition in the visual system. *Progress in Neurobiology*, 51(2):167–94.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–27.

Yang, M., Zhang, L., Feng, X., and Zhang, D. (2011). Fisher discrimination dictionary learning for sparse representation. In *Proceedings of the International Conference on Computer Vision*, pages 543–50.

Yu, K., Zhang, T., and Gong, Y. (2009). Nonlinear learning using local coordinate coding. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22, pages 2223–31. Curran Associates, Inc.

Zhang, H., Zhang, Y., and Huang, T. S. (2013). Simultansous discriminative projection and dictionary learning for sparse represntation based classification. *Pattern Recognition*, 46:346–54.

Zhang, L., Yang, M., and Feng, X. (2011). Sparse representation or collaborative representation: Which helps face recognition? In *Proceedings of the International Conference on Computer Vision*, pages 471–8.

Zhang, Q. and Li, B. (2010). Discriminative k-svd for dictionary learning in face recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698.

Zhang, S., He, B., Nian, R., Wang, J., Han, B., Lendasse, A., and Yuan, G. (2014). Fast image recognition based on independent component analysis and extreme learning machine. *Cognitive Computation*, 6(3):405–422.