Discovering Compositional Structure

by
Matthew T. Harrison
B.A., University of Virginia, 1998
Sc.M., Brown University, 2000

Doctoral dissertation
Ph.D. Advisor: Stuart Geman

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy
in the Division of Applied Mathematics at Brown University

Providence, Rhode Island
May 2005

Chapter 3

Learning selectivity

This chapter with minor differences has been circulated and referenced as a technical report in preparation: M. Harrison and S. Geman. *Compositional feature detectors.* August, 2003.

3.1 Introduction

We are interested in statistical algorithms that learn compositional representations of images. Compositional representations are hierarchies of reusable parts. The parts are both more invariant and more selective higher in the hierarchy. Here we use some simple heuristics based on the iterative learning scheme described in Chapter 2 to learn low-level, compositional image features. The resulting hierarchy has increasingly selective parts, but not increasingly invariant parts. Invariance is addressed in Chapter 4. A key component of our learning algorithm is Barlow's principle of detecting and removing "suspicious coincidences" [22].

As mentioned in Chapter 2, computation using compositional systems is not well understood (although the Ph.D. theses of Potter [24] and Huang [14] have made progress in this direction). The main point of this chapter is to see whether or not we can identify the first few compositions that are likely to occur in such a system, even though the details of how they would then be used for image interpretation have not been worked out.

We will focus mainly on binary images of natural scenes (but see Section 3.2.2). We begin with a generative, probabilistic model of such an image that treats small, non-overlapping image patches as a hidden mixture model (Section 3.2). The model itself is actually learned from the image data (Section 3.2.1), which is interesting but probably not crucial for our later results. What is more important is that the parameters of the model are then estimated from the data using an EM-like procedure (Section 3.3). Once the model has been trained, we can use the model to probabilistically interpret a new image. In particular, for each image patch we can compute the posterior probability of all of the hidden states (Section 3.4). This lets us detect suspicious coincidences among collections of neighboring hidden states (Section 3.5). We use compositionality and (translation) invariance to facilitate the detection of suspicious coincidences (Section 3.5.5). Sparse coding is crucial for preventing an explosion of suspicious coincidences (Section 3.5.3).

3.2 The generative model

The model generates binary $3m \times 3n$-pixel images I by independently generating each of the mn non-overlapping 3×3-pixel patches from the same distribution. We will label these mn patches $Y_{k\ell}$, $1 \le k \le m, 1 \le \ell \le n$, preserving the topology of the image and denoting the individual binary pixels as

$$Y_{k\ell}(i,j) := I(3(k-1)+i, 3(\ell-1)+j) \in \{0,1\}, \quad 1 \le i \le 3,\ 1 \le j \le 3.$$

We use 0 to denote black and 1 to denote white. The independence of the $Y_{k\ell}$ implies that

$$P(I) = \prod_{k=1}^{m} \prod_{\ell=1}^{n} P(Y_{k\ell}).$$

A given binary 3×3 patch is generated from a hidden mixture model: first a representative patch (or no patch) is chosen from some small collection of $S+1$ hidden states and then (flip) noise is added. Using $X_{k\ell} \in \{0, 1, \ldots, S\}$ to denote the hidden state associated with the (k,ℓ)-th image patch $Y_{k\ell}$ gives

$$P(I) = \prod_{k=1}^{m} \prod_{\ell=1}^{n} \left[\sum_{s=0}^{S} P(X_{k\ell} = s) P(Y_{k\ell} | X_{k\ell} = s) \right].$$

Since our model does not distinguish between different image locations, we can complete the description of the model by specifying the distributions $P(X)$ and $P(Y|X)$ for a generic patch Y and hidden state X.

Our model has 11 hidden states with probabilities $P(X = s) := p_s$, $0 \le s \le 10$, which sum to 1. The states $1, \ldots, 10$ correspond to 10 different representative 3×3 binary patches denoted $B_1, \ldots, B_{10}$, where $B_s := \{B_s(i,j) : 1 \le i \le 3,\ 1 \le j \le 3\} \in \{0,1\}^{3\times 3}$. The representative patches that we used are shown in Figure 3.4. They were selected from data using heuristics based on suspicious coincidences and sparse coding (see Section 3.2.1). State 0 corresponds to no representative patch.

To generate an observation of a patch Y, first, one of the 11 states are chosen with the corresponding probabilities. If state 0 is chosen, then each of the 9 pixels in the 3×3 observation patch is independent and identically distributed (i.i.d.) with probability of white β and probability of black $1 - \beta$. So

$$P(Y|X=0) := \prod_{i=1}^{3} \prod_{j=1}^{3} \beta^{\mathbb{1}\{Y(i,j)=1\}} (1-\beta)^{\mathbb{1}\{Y(i,j)=0\}} = \left[\beta^{\|Y\|}(1-\beta)^{1-\|Y\|}\right]^9,$$

where $\mathbb{1}\{A\}$ is the indicator of the event A and

$$\|Y\| := \frac{1}{9} \sum_{i=1}^{3} \sum_{j=1}^{3} \big|Y(i,j)\big|.$$

If one of the states $1, \ldots, 10$ is chosen, then the corresponding representative patch B_s is

selected and the 9 pixels in the observation patch are generated by independently flipping the pixels of the representative patch with some small probability α. This gives

$$P(Y|X=s) := \prod_{i=1}^{3}\prod_{j=1}^{3} \alpha^{\mathbb{1}\{Y(i,j)\neq B_s(i,j)\}}(1-\alpha)^{\mathbb{1}\{Y(i,j)=B_s(i,j)\}}$$
$$= \left[\alpha^{\|Y-B_s\|}(1-\alpha)^{1-\|Y-B_s\|}\right]^9$$

for $s = 1, \ldots, 10$, where

$$\|Y - B_s\| := \frac{1}{9}\sum_{i=1}^{3}\sum_{j=1}^{3}\big|Y(i,j) - B_s(i,j)\big|.$$

Except for the specific parameter values (see Section 3.3), this is a complete probabilistic description of the generative model for images.

3.2.1 Learning the model

The 10 representative binary 3×3-patches used in our generative model and shown in Figure 3.4 were discovered from image data using heuristics based on suspicious coincidences and sparse coding. Under the assumption that every pixel in an image is i.i.d. with probability 1/2 of either black or white, then each of the $2^9 = 512$ possible patches is equally likely. We can easily collect the frequency of occurrence of each of these patches in a collection of images. Patches that occur more frequently than 1/512 are suspicious coincidences.

Using every 3×3 patch from the first 100 images from the image data described in Section 3.3, we computed the frequency of each of the 512 patches and found 44 suspicious coincidences, that is, patches with frequencies greater than 1/512. The histogram of frequencies is shown in Figure 3.2 and the suspicious coincidences are shown in Figure 3.3.

The list of suspicious patches includes many of the features that we would expect, such as the constant patches and the horizontal and vertical edges. Unfortunately, as indicated in Figure 3.3, the collection of suspicious patches is highly redundant. If a certain patch is suspicious, then many of its slight variations (e.g., single pixel flips) will also be suspicious. We cannot keep the entire list of suspicious patches and still maintain a sparse representation of image patches. We need to prune the list.

Ideally, we would like to select a single representative patch for each "feature" and let some sort of noise model take care of the rest. There are many ways to proceed, but the first and simplest thing worked, so that is all that we tried. Each patch has 9 neighboring patches created by flipping the color of a single pixel. For each of the 512 patches, we selected those that were both a suspicious coincidence (frequency > 1/512) and whose frequency was greater than the maximum of its 9 neighbors' frequencies. There were 10 patches that satisfied these criteria. They are shown in Figure 3.4 and were used as the representative patches in the generative model. They are the two constant patches and each of the eight possible horizontal and vertical edges.

3.2.2 Possible extensions of the model

All that we need is a generative model with some hidden states that lets us compute the posterior probabilities of the hidden states given an image (see Section 3.4). These hidden states are conceptualized as features. The collection of them should be small and sparse to facilitate looking for suspicious coincidences among their joint probabilities. With this in mind, several extensions of the model are apparent.

Extending the model to patch sizes other than 3×3 is trivial. Some care may need to be taken in selecting the representative patches for the hidden states. The method described in Section 3.2.1 found 48 different 4×4 patches, composed of the two constant patches, all the horizontal and vertical edges/lines, some diagonal edges/lines and some center-surround patches. These are shown in Figure 3.5. Increasing the patch size much more than this will likely lead to an explosion of representative patches and the sparsening procedure will need to be modified.

Extending the model to images with a few more intensity levels is also straightforward. The noise model will need to be modified slightly. The local maxima sparsening procedure still works on ternary 3×3-patches, finding 40 representative patches composed of the three constant patches, vertical and horizontal lines and edges and a few diagonal edges. These are shown in Figure 3.6. Again, increasing the number of intensity levels quickly leads to an explosion of representative patches using this simple sparsening procedure.

Extending the model to gray-scale or color images will require several major changes. It seems likely that the low-frequency component or the mean intensity level of an image patch should be modeled separately from the high-frequency components. The high-frequency components could be treated like our representative patches with a more sophisticated noise model. The low-frequency component might need to be quantized into a few representative intensity levels, again with a more sophisticated noise model.

The projection pursuit [15, 9] methods of finding collections of filters from natural images seem like an attractive option for discovering the representative hidden states in the model. These methods are exemplified by sparse components analysis [21] and independent components analysis [3, 16]. By locally searching for filters with the highest possible kurtosis (or something like it), these methods are in some ways simultaneously looking for suspicious coincidences and sparsity. Closely related work includes products of experts (with sparse experts) [12, 28] and additive random fields / maximum entropy models [4, 23, 29, 30, 31]. Other related work that could be adapted to the situation here includes [11, 26].

There would still be several striking drawbacks of such a hidden mixture model for image patches. The rigidity of the placement of the non-overlapping patches and the complete lack of any invariance will quickly overwhelm any learning algorithm because a given feature can occur in so many different ways (although, see [25]). The model cannot account for the great variety of instantiations of a feature. It will have to learn them each separately. Unfortunately, modifying the model to accommodate these deficiencies prevents us from easily learning the parameters of the model (see Section 3.3) and computing posterior distributions (see Section 3.4), both of which seem crucial for discovering suspicious coincidences.

3.3 Estimating the model parameters

The generative model has 12 parameters: $p_1, \ldots, p_{10}, \alpha, \beta$. ($p_0$ is fixed by the probability constraint.) These are easily learned from image data using the expectation-maximization (EM) algorithm (see [5] for details). Given a collection of T binary 3×3 image patches $Y^1, \ldots, Y^T$, where $Y^t := \{Y^t(i,j) : 1 \leq i \leq 3,\ 1 \leq j \leq 3\} \in \{0,1\}^{3\times 3}$, the EM update equations are

$$p_s^{\text{new}} = \frac{1}{T} \sum_{t=1}^{T} w_{ts}, \quad s = 0, \ldots, 10,$$

$$\beta^{\text{new}} = \frac{1}{T} \sum_{t=1}^{T} \frac{w_{t0} \|Y^t\|}{p_0^{\text{new}}} \quad \text{and} \quad \alpha^{\text{new}} = \frac{1}{T} \sum_{t=1}^{T} \frac{\sum_{s=1}^{10} w_{ts} \|Y^t - B_s\|}{\sum_{s=1}^{10} p_s^{\text{new}}}.$$

The weights w_{ts} are just the posterior probabilities of the states given the image patch (2) and can be computed from the noise model $P(Y|X)$ using Bayes' rule and the old parameters.

$$w_{ts} = P(X = s | Y = Y^t) = \frac{p_s^{\text{old}} P(Y = Y^t | X = s; \beta^{\text{old}}, \alpha^{\text{old}})}{\sum_{u=0}^{10} p_u^{\text{old}} P(Y = Y^t | X = u; \beta^{\text{old}}, \alpha^{\text{old}})} \tag{1}$$

for $t = 1, \ldots, T, s = 0, \ldots, 10$. All of these computations are relatively straightforward. Derivations of the EM update equations can be found at the end of this section.

The model was trained on all non-overlapping 3×3 patches using the first 1000 natural images from a large collection courtesy of Hans van Hateren and described in [27]. The gray-scale images were first reduced in size to 126×192 pixels (the JPEG thumbnails of van Hateren) and then converted to binary by thresholding each image at its median intensity value. Sample images with enlargements are shown in Figure 3.1.

To avoid having to store all the images in memory simultaneously (relevant for much larger data sets), we ran a single iteration of the EM update equations on each image (2688 images patches per image). This gives a reasonable estimate of the parameters but it weights the final image. To get a better estimate we then repeated this process using a running average update, for example,

$$\beta^{\text{new}} = \frac{1}{t} \beta^{\text{new}} + \frac{t-1}{t} \beta^{\text{old}},$$

where the β^{new} on the right comes from the original EM update equation and t is the current image number. The initial parameters for the whole process were $\alpha = .1$, $\beta = .5$ and $p_1, \ldots, p_{10}$ set to the empirical probabilities of their respective patches (see Section 3.2.1). The fitted model parameters are shown in the next table.

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
0.3572	0.0127	0.0101	0.0100	0.0092	0.0114	0.0107	0.0130
p_9	p_{10}	p_0	α	β			
0.0101	0.3423	0.2133	0.0403	0.5046			

The relative entropy between the truth (empirical probabilities of all possible 512 patches)

and our fitted model for image patches is 0.1935 bits/patch (or 0.0215 bits/ pixel). This is a simple way of quantifying the fit of the model and can be interpreted as the penalty that we would expect to pay using our model to compress image patches instead of the ideal model. The entropy of the empirical distribution is 4.6505 bits/patch (or 0.5167 bits/pixel), so this penalty only increases code lengths by about 4%. For comparison, the entropy of the model is 5.5033 bits/patch (or 0.6115 bits/pixel).

Proof of the EM update equations. For estimating the parameters of a hidden mixture model, the EM equations for the weights w_{ks} (1) are the posterior probabilities of the hidden states given the observations and the previous parameter values. The new (next step) estimates of the mixing probabilities p_s^{new} are the average of the posterior probabilities (the weights) [5]. The parameters of the noise model $(\alpha^{\text{new}}, \beta^{\text{new}})$ are maximizers of

$$\sum_{t=1}^{T}\sum_{s=0}^{10} w_{ts} \log P(Y^t|\text{state } s; \alpha, \beta)$$
$$= \sum_{t=1}^{T} w_{t0} \log \Big[\beta^{\|Y^t\|}(1-\beta)^{1-\|Y^t\|}\Big]^9 + \sum_{t=1}^{T}\sum_{s=1}^{10} w_{ts} \log \Big[\alpha^{\|Y^t - B^s\|}(1-\alpha)^{1-\|Y^t-B^s\|}\Big]^9,$$

over $0 \le \alpha, \beta \le 1$, where the substitutions came from the noise model $P(Y|X)$ in Section 3.2. Differentiating this expression gives

$$\frac{\partial}{\partial\beta}\Big[\cdots\Big] = \frac{9}{\beta}\sum_{t=1}^{T} w_{t0}\|Y^t\| - \frac{9}{1-\beta}\sum_{t=1}^{T} w_{t0}(1-\|Y^t\|),$$
$$\frac{\partial}{\partial\alpha}\Big[\cdots\Big] = \frac{9}{\alpha}\sum_{t=1}^{T}\sum_{s=1}^{10} w_{ts}\|Y^t - B^s\| - \frac{9}{1-\alpha}\sum_{t=1}^{T}\sum_{s=1}^{10} w_{ts}(1-\|Y^t - B^s\|).$$

Solving for zeros to find the maximizers gives the new parameter values

$$\beta^{\text{new}} = \frac{\sum_{t=1}^{T} w_{t0}\|Y^t\|}{\sum_{t=1}^{T} w_{t0}} \quad \text{and} \quad \alpha^{\text{new}} = \frac{\sum_{t=1}^{T}\sum_{s=1}^{10} w_{ts}\|Y^t - B^s\|}{\sum_{t=1}^{T}\sum_{s=1}^{10} w_{ts}},$$

which can be rewritten as in the text. □

3.4 Computing the posterior

Once we have specified all the parameters of the model, we can compute the posterior probability of each of the 11 hidden states given an image patch Y using Bayes' rule and the noise model $P(Y|X)$:

$$P(X = s|Y) = \frac{p_s P(Y|X=s)}{\sum_{u=0}^{10} p_u P(Y|X=u)}, \quad s = 0, \ldots, 10. \tag{2}$$

This is the same computation for the weights used by EM when training the model (1).

For an image I we can independently compute the 11 posterior probabilities given each of the mn non-overlapping patches $Y_{k\ell}$. This is an exact computation because of the independence assumption in the model. Preserving the topology of the image grid allows us to arrange the posterior probabilities into a new $m \times n$ grid with 11 values at each point, or equivalently, 11 different $m \times n$ grids (perhaps visualized as stacked on top of one another).

We will denote the posterior probability of state s from the (k, ℓ)-th patch as

$$Q_{sk\ell}(I) := P(X_{k\ell} = s|I) = P(X_{k\ell} = s|Y_{k\ell}),$$

$k = 1, \ldots, m$, $\ell = 1, \ldots, n$, $s = 0, \ldots, 10$. This is just (2) with extra notation to indicate where the patch is located in the image. Since $Q_{sk\ell}$ is a probability, it has a value between 0 and 1. For $s = 1, \ldots, 10$, $Q_{sk\ell}$ can be viewed as a non-linear filter based on representative patch B_s applied to the (k, ℓ)-th image patch. We do not need to explicitly compute $Q_{0k\ell}$ because it is fixed by the probability constraint $\sum_{s=0}^{10} Q_{sk\ell} = 1$.

In summary, given an image we compute the values of 10 different non-linear filters centered at each non-overlapping 3×3 image patch location. These filter values $Q_{sk\ell}$ are the posterior probabilities of the states given the image patches using our generative model with the fitted parameters. We can use Q to look for certain deficiencies in the model, namely suspicious coincidences.

3.5 Detecting suspicious coincidences

Our model describes a probability distribution P on images as well as hidden states X. The world also has a true probability distribution $\mathbb{P}$ for images but not for hidden states, since these are an invention of the model and do not necessarily correspond to reality. We can, however, combine these distributions to create a "true" distribution for hidden states

$$\mathbb{P}(X_{k\ell} = s) := \mathbb{E}\left[P(X_{k\ell} = s|I)\right],$$

where $\mathbb{E}$ denotes expectation over images I with distribution $\mathbb{P}$. We also have the identity

$$P(X_{k\ell} = s) = E\left[P(X_{k\ell} = s|I)\right],$$

where E denotes expectation over images I with distribution P. Using $\mathbb{P}$ to talk about the hidden states of the model is an abuse of notation, but it nicely captures the intuition. If $\mathbb{P}$ and P give the same distribution on images, that is, if $\mathbb{E}$ and E are the same expectation, then $\mathbb{P}$ and P give the same distribution on hidden states. By computing the distribution on hidden states in both cases, we can evaluate our model for images.

Of course, $\mathbb{P}$ is unknown, but we can approximate it by using the empirical distribution $\hat{\mathbb{P}}$ of a large collection of images $I^1, \ldots, I^T$. For example

$$\hat{\mathbb{P}}(X_{k\ell} = s) := \hat{\mathbb{E}}\left[P(X_{k\ell} = s|I)\right] := \frac{1}{T}\sum_{t=1}^{T} P(X_{k\ell} = s|I^t) := \frac{1}{T}\sum_{t=1}^{T} Q_{sk\ell}(I^t).$$

Each of these expressions is just a different way of writing the same thing. The $\hat{\mathbb{P}}$ notation

on the left is useful for thinking about suspicious coincidences and other probabilistic considerations. The Q notation on the right shows exactly what would be computed by the algorithm.

Our model does not distinguish among patch or hidden state locations so it makes sense to talk about $P(X)$ for a generic hidden state X. If we want to condition on an image, however, $P(X|I)$ is ambiguous and this creates problems for defining $\mathbb{P}(X)$ in the same way that $\mathbb{P}(X_{k\ell})$ was defined. One way to make sense of this is to let X be a patch chosen randomly and uniformly from the possible patch locations in the image, which is $P(X = s|Y = Y_{k\ell})$ averaged over each patch $Y_{k\ell}$ in the image I.

$$P(X = s|I) := \frac{1}{mn} \sum_{k=1}^{n} \sum_{\ell=1}^{m} P(X = s|Y = Y_{k\ell}).$$

This now lets us define

$$\mathbb{P}(X = s) := \mathbb{E}\left[P(X = s|I)\right],$$

which is approximated by the empirical distribution

$$\hat{\mathbb{P}}(X = s) := \hat{\mathbb{E}}\left[P(X = s|I)\right] := \frac{1}{T} \sum_{t=1}^{T} P(X = s|I^t) := \frac{1}{T} \sum_{t=1}^{T} \frac{1}{mn} \sum_{k=1}^{m} \sum_{\ell=1}^{n} Q_{sk\ell}(I^t). \tag{3}$$

The main reasons for dealing with a generic X is to make $\hat{\mathbb{P}}$ a better approximation of $\mathbb{P}$ for a given number of images because of the increased amount of averaging and to reduce the number of statistics that we have to measure. The drawback is that we lose the ability to detect statistics for specific locations in the image plane. If the true distribution for images is translation invariant, then we will have lost nothing. In this way our algorithm makes explicit use of an invariance bias.

As previously mentioned, one way to compare our distribution for images $P(I)$ to the true distribution $\mathbb{P}(I)$ is to verify that $P(X = s) = \mathbb{P}(X = s)$ for each s. We cannot do this, but we can verify that

$$p_s := P(X = s) \approx \hat{\mathbb{P}}(X = s) := \frac{1}{T} \sum_{t=1}^{T} \frac{1}{mn} \sum_{k=1}^{m} \sum_{\ell=1}^{n} Q_{sk\ell}(I^t), \tag{4}$$

because both sides are either known or easily computable from a collection of images. If this approximation is clearly violated, then there is something wrong with our model. In fact, this particular approximation will be quite good because the EM learning algorithm that we used is designed to enforce this constraint. We will need a different statistic to identify the problems with our model.

One of the striking deficiencies in our model is the assumed independence among patches. Neighboring patches in an image will likely have many of the same statistical properties. For example, the state corresponding to the all black patch is much more likely (than independence would predict) to have all black neighbors because of the presence of large contiguous regions in images. Similarly, the state corresponding to a horizontal edge is much more likely to have left / right neighbors which are also horizontal edges because images have long

continuous edges. We can detect these discrepancies using Q. This amounts to searching for suspicious coincidences.

The model asserts that $P(X_{k\ell} = s, X_{k'\ell'} = s') = P(X_{k\ell} = s)P(X_{k'\ell'} = s')$ as long as $(k, \ell) \neq (k', \ell')$. Does $\mathbb{P}$ have the same independence?

$$\begin{aligned}\mathbb{P}(X_{k\ell} = s, X_{k'\ell'} = s') &:= \mathbb{E}\left[P(X_{k\ell} = s, X_{k'\ell'} = s'|I)\right] \\ &= \mathbb{E}\left[P(X_{k\ell} = s|I)P(X_{k'\ell'} = s'|I)\right] \stackrel{?}{=} \mathbb{P}(X_{k\ell} = s)\mathbb{P}(X_{k'\ell'} = s').\end{aligned}$$

We can test this using $\hat{\mathbb{P}}$ and Q, but we would first like to incorporate the location invariance of the model and the presumed translation invariance of $\mathbb{P}$ by using a generic hidden states X and X' instead of hidden states $X_{k\ell}$ and $X_{k'\ell'}$ with specific locations in the image. The relative coordinates of X and X' will still need to be preserved; the joint statistics of neighboring patches might be quite different from those of distant patches. We will use the notation $(X, X')_{k_0\ell_0}$ to denote a generic pair of hidden states X and X' with X' offset $(k_0, \ell_0) \neq (0, 0)$ from X. The location of this pair is chosen randomly and uniformly from all possible locations so that both hidden states fit in the image. For example, $(X, X')_{10}$ means that X is uniformly selected from $\{X_{k\ell} : 1 \leq k \leq m-1, 1 \leq \ell \leq m\}$ and that X' is the immediate right neighbor of X. We thus have

$$\begin{aligned}&P((X, X')_{k_0\ell_0} = (s, s')|I) \\ &:= \frac{1}{(m-k_0)(n-\ell_0)} \sum_{k=1}^{m-k_0} \sum_{\ell=1}^{n-\ell_0} P(X_{k\ell} = s, X_{(k+k_0)(\ell+\ell_0)} = s'|I) \\ &= \frac{1}{(m-k_0)(n-\ell_0)} \sum_{k=1}^{m-k_0} \sum_{\ell=1}^{n-\ell_0} P(X_{k\ell} = s|Y_{k\ell})P(X_{(k+k_0)(\ell+\ell_0)} = s'|Y_{(k+k_0)(\ell+\ell_0)}).\end{aligned}$$

We can now define

$$\mathbb{P}((X, X')_{k_0\ell_0} = (s, s')) := \mathbb{E}\left[P((X, X')_{k_0\ell_0} = (s, s')|I)\right],$$

which is approximated by

$$\begin{aligned}\hat{\mathbb{P}}((X, X')_{k_0\ell_0} = (s, s')) &:= \hat{\mathbb{E}}\left[P((X, X')_{k_0\ell_0} = (s, s')|I)\right] \\ &:= \frac{1}{T} \sum_{t=1}^{T} P((X, X')_{k_0\ell_0} = (s, s')|I^t) \\ &:= \frac{1}{T} \sum_{t=1}^{T} \frac{1}{(m-k_0)(n-\ell_0)} \sum_{k=1}^{m-k_0} \sum_{\ell=1}^{n-\ell_0} Q_{sk\ell}(I^t)Q_{s'(k+k_0)(\ell+\ell_0)}(I^t). \qquad (5)\end{aligned}$$

Does

$$\mathbb{P}((X, X')_{k_0\ell_0} = (s, s')) \stackrel{?}{=} \mathbb{P}(X = s)\mathbb{P}(X' = s')$$

as the model predicts, or is there some additional dependence? We can test this by verifying that

$$\hat{\mathbb{P}}((X, X')_{k_0\ell_0} = (s, s')) \approx \hat{\mathbb{P}}(X = s)\hat{\mathbb{P}}(X' = s').$$

Both sides are easily computable using (3) and (5). We are specifically interested in situations where

$$\hat{\mathbb{P}}((X, X')_{k_0\ell_0} = (s, s')) \gg \hat{\mathbb{P}}(X = s)\hat{\mathbb{P}}(X' = s'), \tag{6}$$

a suspicious coincidence.

3.5.1 Second-order suspicious coincidences

The EM algorithm takes care of first-order suspicious coincidences (departures from the model), in the sense that it makes the (first-order marginal) probabilities of the hidden states match their empirical estimates from Q. That is, (4) is a valid approximation.

We can now look for second-order suspicious coincidences – when the joint probability of two states in different locations is higher than predicted by independence – by finding state pairs (s, s') and offsets (k_0, ℓ_0) where (6) holds.

An image has $(m-1)(n-1)-1$ different allowable offsets (k_0, ℓ_0) (we do not need to consider negative offsets because these are included by switching s and s') and our model has $S+1$ different hidden states. This gives about mnS^2 different binary associations for consideration in (6). Since mn can be quite large, this is a significant memory burden. We expect the independence assumption to be most violated by neighboring patches, so we can restrict ourselves to the cases where the offset corresponds to neighboring locations in an image. For example, we can only consider the 2 offsets horizontal $(1, 0)$ and vertical $(0, 1)$. This gives about $2S^2$ different associations to remember, which is much more manageable.

We also restrict the states to $1, \ldots, S = 10$, and do not consider state 0. States $1, \ldots, S$ represent the presence of a specific feature in the image patch, like a horizontal edge. State 0 represents the absence of any features. Ignoring state 0 maintains some consistency between the setup in this chapter and a more general framework that we are developing in which there will be no state 0. It is also more in the spirit of compositionality, where multiple *present* features are composed into a new high-level feature. In all then, we will only consider $2S^2 = 200$ different possible suspicious coincidences using (6).

For a collection of images $I^1, \ldots, I^T$, and each pair of states (s, s') we compute $\hat{\mathbb{P}}(X = s)$, $\hat{\mathbb{P}}(X' = s')$, $\hat{\mathbb{P}}((X, X')_{10} = (s, s'))$ and $\hat{\mathbb{P}}((X, X')_{01} = (s, s'))$ using Q as indicated in (3) and (5). A suspicious coincidence is registered when

$$\hat{\mathbb{P}}((X, X')_{10} = (s, s')) > \hat{\mathbb{P}}(X = s)\hat{\mathbb{P}}(X' = s')$$

and similarly for offset $(0, 1)$.

3.5.2 A minimum description length (MDL) criterion

This method identifies 89 different second-order suspicious coincidences. We use a minimum description length (MDL) criterion to rank them. Each state pair (s, s') and each offset (k_0, ℓ_0) (we only consider two of them) is assigned a measure of suspiciousness

$$r_{k_0\ell_0}(s, s') := \hat{\mathbb{P}}((X, X')_{k_0\ell_0} = (s, s')) \log \frac{\hat{\mathbb{P}}((X, X')_{k_0\ell_0} = (s, s'))}{\hat{\mathbb{P}}(X = s)\hat{\mathbb{P}}(X' = s')}.$$

This is a reasonable measure because $r_{k_0\ell_0}(s, s') > 0$ exactly when we have a suspicious coincidence. It increases as the joint probability becomes proportionally larger than the product of the probabilities, giving a higher rank to larger departures from independence. It also increases as the joint probability increases, giving a higher rank to feature combinations that occur more frequently. From an information theory point of view, we can loosely interpret r as the number of bits (using $\log_2$) that we would save on average by coding with a probability distribution that accounted for this suspicious coincidence as compared with one that did not (our model). MDL-like criteria are quite common for learning and evaluating models [2].

Each of the 89 second-order suspicious coincidences that are discovered are shown in Figure 3.7. They are ranked from highest to lowest using r. The constant patches are first, followed by the extended edges, then some other edge or line elements, then some high frequency elements and then some corner or junction configurations. Many of these combinations intuitively make sense when thinking about how the independence assumption might be violated in natural images.

Ideally, we would use this information to create a new model, probably by adding another hierarchical layer of hidden states, that incorporates these dependencies. From the compositionality perspective, these dependencies arise because the low-level features occasionally occur as parts of a higher-level feature. The new layer capturing these suspicious coincidences would thus represent these higher-level features. Although we do not actually build this new model, we can still think about the new higher-level representation. What are the suspicious coincidences among elements of this new level? Is the representation appropriate for detecting suspicious coincidences? Can the process be iterated and where does it break down? These are some of the questions that we try to address in the remainder of this chapter.

3.5.3 Using sparse coding

Figure 3.7 has many similar elements. If a given local region of an image "excites" one of these elements, that is, gives it a high posterior probability, then the same region is likely to "excite" another, similar element. The representation is not sparse. As mentioned earlier this will cause a combinatorial explosion of suspicious coincidences in higher levels. We need to prune the representation to make it sparser.

The local maxima procedure described in Section 3.2.1 for finding the representative 3×3 patches also works here. We only keep those elements in Figure 3.7 whose MDL measure of suspiciousness, r, is higher than any of its neighboring elements. We use an ad hoc method for determining neighbors. Compositions of the same shape (we only have two shapes: horizontal neighbors or vertical neighbors) are neighbors if their per pixel Hamming distance is less than 1/3. That is,

$$\frac{\|B_{s_1} - B_{s_2}\| + \|B_{s'_1} - B_{s'_2}\|}{2} < \frac{1}{3},$$

where (s_1, s'_1) are the state pairs for one element and (s_2, s'_2) are the states for another. 1/3 was chosen because it is the minimum per pixel Hamming distance between any two different

representative patches B_s. For compositions of a different shape we do the same thing where they overlap (at states s_1 and s_2) and add a penalty of 1/2 for the patches that do not overlap. The criterion is

$$\frac{\|B_{s_1} - B_{s_2}\| + 1/2}{2} < \frac{1}{3}.$$

1/2 was chosen because it is the mean per pixel Hamming distance between any two different representative patches B_s. The patches are always aligned by the upper left corner for comparison.

The 16 elements of the sparsened representation are shown in Figure 3.8. Each of the 4 constant intensity patches and the 8 horizontal and vertical edges remain. The remaining 4 patches are 2-pixel width lines. Presumably, these 16 would be the "representative compositions" in the next layer of the model and they would capture some of the dependencies detected by the suspicious coincidences. These are the only 2-patch compositions that are considered in later iterations of this procedure.

The distance function that we use for determining neighboring patches was the first one that we tried. Later investigations showed that the behavior of our algorithm is incredibly sensitive to the parameters 1/3 and 1/2. Changing these constants even slightly can cause the sparsening procedure to drastically over or under prune. A more principled and hopefully more robust approach to pruning would use statistical information to determine neighbors. Two similar compositions in the same image location will be highly correlated and could thus be identified as neighbors. The local maxima procedure or some other clustering algorithm could then be used on this statistical distance. We leave this idea for future implementations. The notion of using statistical dependencies to measure redundancies and then remove them to obtain a sparser representation is nearly as old as the notion of sparseness itself. See Földiák [8] for an early computational example and Hyvärinen et al. [17, 18] for more recent developments.

3.5.4 Higher order suspicious coincidences

Up to this point we have only discussed binary associations, but we can also consider higher order suspicious coincidences

$$P(A_1, \ldots, A_N) \gg P(A_1) \cdots P(A_N).$$

In principle, nothing really changes except the notation becomes burdensome. We will use $(X^1, X^2, \ldots, X^N)$ to represent an Nth-order generic composition of patches. To denote the relative coordinates of the X^ν, we subscript the collection with a list of $N-1$ ordered pairs, each denoting the offset from the position of X^1,

$$(X^1, X^2, \ldots, X^N)_{k^2\ell^2, k^3\ell^3, \ldots, k^N\ell^N},$$

so that X^ν is offset (k^ν, ℓ^ν) from X^1. This is consistent with our previous notation $(X, X')_{k_0\ell_0}$, but now we would prefer to write $(X^1, X^2)_{k^2\ell^2}$. We can define $P(\cdot|I)$ and then $\mathbb{P}(\cdot)$ as before,

approximating the latter with

$$\begin{aligned}&\hat{\mathbb{P}}((X^1,\ldots,X^N)_{k^2\ell^2,\ldots,k^N\ell^N}=(s^1,\ldots,s^N))\\&\quad:=\frac{1}{T}\sum_{t=1}^{T}\frac{1}{(m-k_0)(n-\ell_0)}\sum_{k=1}^{m-k_0}\sum_{\ell=1}^{n-\ell_0}\prod_{\nu=1}^{N}Q_{s^\nu(k+k^\nu)(\ell+\ell^\nu)}(I^t),\end{aligned}\tag{7}$$

where we take $(k^1,\ell^1):=(0,0)$ and we define

$$k_0:=\max_{\nu\leq N}k^\nu,\quad \ell_0:=\max_{\nu\leq N}\ell^\nu.$$

As long as $(k^\nu,\ell^\nu)\neq(k^\mu,\ell^\mu)$ for $1\leq\nu\neq\mu\leq N$, the model predicts that this distribution should (approximately) factor. An Nth-order suspicious coincidence is detected when

$$\hat{\mathbb{P}}((X^1,\ldots,X^N)_{k^2\ell^2,\ldots,k^N\ell^N}=(s^1,\ldots,s^N))\gg\prod_{\nu=1}^{N}\hat{\mathbb{P}}(X=s^\nu).\tag{8}$$

Unfortunately, this approach does not extend very far. The number of Nth-ordered pairs increases exponentially, not to mention the steadily increasing number of possible spatial arrangements. Even if we restrict ourselves to connected components, there are $6S^3=6\,000$ combinations for 3rd-order associations, $19S^4=190\,000$ for 4th-order and $55S^5=5\,500\,000$ for 5th-order, which is approaching the limits of feasible computation. We partially surmount this problem by making explicit use of the compositionality bias.

3.5.5 Using compositionality

Compositionality asserts that the structure found in natural images can be built up hierarchically with reusable parts. If the features that we just detected with suspicious coincidences are, say, level 2 in this hierarchy, then we should be able to build level 3 features by looking for suspicious coincidences among level 2 features. These level 3 features will be high-order suspicious coincidences back in the original data, but they will be low-order suspicious coincidences in the level 2 data. Iterating this process a few times will allow us to detect very high-order structure in the original data, much higher order than would ever be feasible using the exhaustive search techniques of the previous section. If natural images are truly compositional, then we may not be sacrificing much for this incredible gain in efficiency. The high-level features that we find will not only be suspicious coincidences in the pixel statistics, but also suspicious coincidences among reusable parts at many hierarchical levels.

In Section 3.5.1 we described how to find second-order suspicious coincidences. After ranking (Section 3.5.2) and pruning (Section 3.5.3), we were left with a sparse collection of 16 features shown in Figure 3.8. These, along with the 10 original representative patches shown in Figure 3.4 (Section 3.2.1), are the reusable parts that will be composed into features in the next level of the hierarchy.

We will use notation similar to that in Section 3.5.4, adding parentheses to indicate the

hierarchical relationship among the components. For example,

$$(X^1, (X^2, X^3)_{k^3\ell^3})_{k^2\ell^2}$$

denotes three generic hidden states with specific relative offsets. X^1 can be in any allowable location. Once the position of X^1 is fixed, the $(X^2, X^3)_{k^3\ell^3}$ composition unit is offset (k^2, ℓ^2) from the position of X^1. The position of this unit is its first member's position, in this case, X^2. So X^2 is offset (k^2, ℓ^2) from X^1 and X^3 is offset (k^3, ℓ^3) from X^2, which means X^3 is offset $(k^2 + k^3, \ell^2 + \ell^3)$ from X^1. We can rewrite this as a 3rd-order association

$$(X^1, (X^2, X^3)_{k^3\ell^3})_{k^2\ell^2} \Longleftrightarrow (X^1, X^2, X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)}, \tag{9}$$

but then we loose the fact that (X^2, X^3) are composed into a 2nd-order feature. The order of binding does not matter, but of course that can modify the relative offset:

$$(X^1, (X^2, X^3)_{k^3\ell^3})_{k^2\ell^2} \Longleftrightarrow ((X^2, X^3)_{k^3\ell^3}, X^1)_{(-k^2)(-\ell^2)}.$$

When a hierarchical association like this is rewritten in the form of a simple higher-order association, as in (9), we always require that the induced offsets are valid, in the sense that none are identically $(0, 0)$ and none are the same. This ensures that each of the hidden states occupies a different, non-overlapping position in the image. The original generative model then asserts that they are all independent (and identically distributed).

A compositional association like this will be a suspicious coincidence if its highest binding is suspicious:

$$\mathbb{P}\left((X^1, (X^2, X^3)_{k^3\ell^3})_{k^2\ell^2} = (s^1, (s^2, s^3))\right) \gg \mathbb{P}(X^1 = s)\mathbb{P}((X^2, X^3)_{k^3\ell^3} = (s^2, s^3)).$$

Rewriting the left side as a simple higher-order suspicious coincidence (9) and using the empirical distribution gives

$$\begin{aligned}&\hat{\mathbb{P}}((X^1, X^2, X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)} = (s^1, s^2, s^3)) \\ &\quad \gg \hat{\mathbb{P}}(X^1 = s^1)\hat{\mathbb{P}}((X^2, X^3)_{k^3\ell^3} = (s^2, s^3)).\end{aligned} \tag{10}$$

Both sides are easily computable using (3), (5) and their generalization (7). In fact, both terms on the right will have already been computed during the search for 2nd-order suspicious coincidences.

Even though the left side is a 3rd-order suspicious coincidence, we do not search all possible 6 000 such connected components. We demand that $(X^2, X^3)_{k^3\ell^3}$ is one of the 16 allowable 2nd-order associations found previously (Figure 3.8). As usual, we continue to require that $(X^1, X^2, X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)}$ forms a connected component (diagonals are not allowed). This gives 10 possibilities for X^1, 16 for $(X^2, X^3)_{k^3\ell^3}$ and 6 for (k^2, ℓ^2) for a total of 960 new associations to consider. While only a moderate reduction in the size of the search space, iterating this idea leads to enormous gains at higher levels.

We still use an MDL ranking (Section 3.5.2)

$$r((s^1,(s^2,s^3)_{k^3\ell^3})_{k^2\ell^2}) := \hat{\mathbb{P}}((X^1,X^2,X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)} = (s^1,s^2,s^3)) \times \log \frac{\hat{\mathbb{P}}((X^1,X^2,X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)} = (s^1,s^2,s^3))}{\hat{\mathbb{P}}(X^1 = s^1)\hat{\mathbb{P}}((X^2,X^3)_{k^3\ell^3} = (s^2,s^3))},$$

finding 282 configurations with $r > 0$. Some of these configurations are identical because there can be multiple compositions that lead to the same pixel configuration. We immediately prune any exact (pixel level) repeats, leaving only a single representative composition for each (the one with the highest r). This leaves 258 compositions, shown in Figure 3.9 and ranked by decreasing r.

There are many similar elements and we use the same local maxima pruning procedure to get a sparser representation (see Section 3.5.3). We compute the distance between two Nth-order patch configurations ($9N$ pixels), by first aligning the patches into the upper left corner of the same box (so a patch must be touching on the top and on the left), adding the per-pixel Hamming distance of the patches that align, adding 1/2 for each misaligned patch (not double counting) and dividing by N to get a per-pixel measure. If this distance is less than 1/3, the configurations are neighbors. For example, if two N-th order configurations with states $(s_1^1,\ldots,s_1^N)$ and $(s_2^1,\ldots,s_2^N)$ have K spatially overlapping patches, indexed by $\nu^1,\ldots,\nu^K$, then they will be neighbors if

$$\frac{\sum_{j=1}^K \|B_{s_1^{\nu_j}} - B_{s_2^{\nu_j}}\| + (1/2)(N-K)}{N} < \frac{1}{3}. \tag{11}$$

As discussed in Section 3.5.3, this just happens to work, is quite sensitive to the parameters 1/2 and 1/3 and can probably be accomplished with many added benefits using some sort of statistical distance.

The sparsened list has 28 3rd-order configurations, shown in Figure 3.10. Each of these is a suspicious coincidence between a single patch and a 2nd-order suspicious coincidence, all in a fixed configuration. Each is also a 3rd-order suspicious coincidence among single patches as defined by (8). In fact any compositional suspicious coincidence is also a (stronger) suspicious coincidence at all lower levels. This can be seen by multiplying the likelihood ratios. For example,

$$\begin{aligned}
&\frac{\hat{\mathbb{P}}((X^1,X^2,X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)} = (s^1,s^2,s^3))}{\hat{\mathbb{P}}(X^1 = s^1)\hat{\mathbb{P}}(X^2 = s^2)\hat{\mathbb{P}}(X^3 = s^3)} \\
&\quad > \frac{\hat{\mathbb{P}}((X^1,X^2,X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)} = (s^1,s^2,s^3))}{\hat{\mathbb{P}}(X^1 = s^1)\hat{\mathbb{P}}(X^2 = s^2)\hat{\mathbb{P}}(X^3 = s^3)} \frac{\hat{\mathbb{P}}(X^2 = s^2)\hat{\mathbb{P}}(X^3 = s^3)}{\hat{\mathbb{P}}((X^2,X^3)_{k^3\ell^3})} \\
&\quad = \frac{\hat{\mathbb{P}}((X^1,X^2,X^3)_{k^2\ell^2,(k^2+k^3)(\ell^2+\ell^3)} = (s^1,s^2,s^3))}{\hat{\mathbb{P}}(X^1 = s^1)\hat{\mathbb{P}}((X^2,X^3)_{k^3\ell^3})} > 1,
\end{aligned}$$

where the first inequality comes from the fact that $(X^2,X^3)_{k^3\ell^3}$ is a suspicious coincidence and the second from the same fact about $(X^1,(X^2,X^3)_{k^3\ell^3})_{k^2\ell^2}$. This same idea iterates to higher orders, as does the computational procedure.

There are important reasons that we rank a coincidence using the highest level of composition and not the product of all the lowest level elements. A large suspicious coincidence can have a very large likelihood ratio when measured at the lowest level, to the degree that almost anything small can be composed with it and the likelihood ratio will still be much larger than one – a suspicious coincidence. While the new configuration is definitely unusual structure, the only interesting structure comes from the large piece. Nothing is added by composing it with the smaller piece. Moreover, in future work we expect the dependencies discovered at each new level to be incorporated into an updated model. In this new model, the dependencies are accounted for and are no longer suspicious coincidences when compared to the lowest level. The only interesting dependencies are to be found in new compositions which are naturally compared to the marginal statistics of the current level, not the all the way back to the lowest level.

3.6 Results

We can easily iterate the procedure described thus far. We look for suspicious coincidences between single patches and 3-patch configurations and also between two 2-patch configurations to discover 4th-order suspicious coincidences. We remove identical (pixel level) elements, rank with an MDL measure that only considers the highest level of composition and sparsen using (11). This gives 26 level 4 configurations shown in Figure 3.11. Now we compose levels 1 and 4 and levels 2 and 3 to search level 5. The sparsened level 5 representation has 26 elements shown in Figure 3.12. We continue in this manner until level 8 (Figures 3.13–3.15), after which we begin to have computational difficulties (mostly because of the inefficiency of our Matlab implementation when generating a list of all possible compositions that need to be considered). The number of elements in each level is shown in the next table.

order	1	2	3	4	5	6	7	8
# elements	10	16	28	26	26	65	81	158

The higher levels begin to look quite noisy. Indeed, an 8th-order suspicious coincidence is composed of 72 pixels and thus represents unusual statistical structure in a 72-dimensional space. Estimating such structure should require a lot of data. Increasing the number of images for training improves things somewhat. We have experimented with up to 4000 images. Because the representation is less noisy, the local maxima sparsening procedure produces fewer and less noisy elements. This allows the method to continue into the 10th or 11th stage of iteration before things become noisy and the computation breaks down.

We have been able to find 32-order suspicious coincidences by skipping stages and by considering only a small subset of the possible compositions at each stage. At each new level, we only consider compositions created by combining together two configurations from the previous level. This creates suspicious coincidences with orders that are successive powers of 2. We also use a different sparsening procedure, keeping only the best configuration (according to the MDL ranking r) that uses each composition found in the previous level. This prevents the number of elements from increasing. This highly constrained search finds a few 32-patch horizontal and vertical perfect edges. These compositions are 96×3 pixel structures, which is a significant edge in a 126×192 pixel image. It seems unlikely that any

of the training images contain one of these configurations exactly (although we have not checked), but the model allows for noisy perturbations. These detected structures probably correspond to horizon lines, tree trunks, roads, buildings and other such large image features.

3.6.1 Other data sets

We have experimented briefly with several other image data sets, including the same images used here but at a much higher resolution (1023×1536). The results are all qualitatively similar. Some of the differences between natural images and text images are worth mentioning.

We used the same 10 representative patches, but trained the model parameters and searched for suspicious coincidences with (single author) handwritten text images. The resulting compositions were quite similar to the ones we found here with three notable exceptions. Line-like compositions were more prevalent than edge-like compositions, large all-black compositions were absent and diagonal lines had replaced vertical lines. This last difference is important. It represents the slant of the handwriting and demonstrates that the algorithm is not forced to find vertical and horizontal edges exclusively, even though it is clearly biased in that direction because the of grid nature of the pixel representation and because of the vertical and horizontal (but no diagonal) neighborhood constraint.

Another interesting data set was fixed font (Times New Roman 12pt) text, generated by turning a PDF manuscript into a binary GIF image. Again we used the same 10 representative patches. We had hoped that the algorithm might discover actual letters, but this was not the case. The suspicious coincidences that we discovered mostly represented features characteristic of the typesetting, such as the typical inverted T shape at the bottom of many letters. We also found very long horizontal lines and even long parallel pairs of horizontal lines representing the bottom of a line of text or in the parallel case, two lines of text. These sorts of features quickly came to dominate the representation.

3.7 Related Work

Chapter 2 contains several references on the general theme of sequential model building. Section 3.2.2 of this chapter contains some pointers to related work on unsupervised feature selection. There is also a large body of empirical work describing various properties of natural image statistics, including the strong signal for lines and edges that our model naturally detected, for example, [10, 6, 1]. Here we will focus mostly on work related to growing hierarchies of features with increasing selectivity. Note that almost any hierarchical neural network model will loosely fit into this category (for example [20]), but we do not review that literature here.

The general principle behind the feature induction algorithm in [23] is similar to the idea here of growing more complicated features from simpler features that have already been discovered and incorporated into the model. Other comparisons with this work are discussed in Chapter 2.

In Section 3.2.2 we noted that our initial generative model bears some resemblance to sparse coding models, like independent components analysis (ICA). Several groups have

experimented with various methods of adding more hidden layers on top of a sparse coding basis in order to extract higher-order dependencies, much like our goal here. Although these methods typically only operate on a single patch, whereas here we are operating on spatially adjacent patches, this difference is perhaps not as big as it seems. Sparse coding bases represent much larger patches (64×64-pixel patches are not uncommon) with the result that many basis elements are confined to a small spatial extent within the large patch. These spatially localized basis functions are analogous to our original 10 ideal patches. Extracting higher-order dependencies within the large patch is then analogous to composing our small patches into larger ones.

Hyvärinen and Hoyer (2001) [17, 18] show that the dependencies among ICA units can be used to define a topographic ordering on the units. This topography can then be used in various ways to (locally) pool the outputs of the ICA units and create a new layer of units with interesting properties. Depending on the (nonlinear) pooling function, these higher-order units can have various properties, including certain types of invariance. Further adding another sparse coding layer on top of these units creates basis functions that often look like longer lines and edges [13], reminiscent of the results here. Presumably this procedure could be iterated to create increasingly complex units. Note that the nonlinear pooling operation is important because simply iterating ICA on the same patch is not effective: all the transformations are linear and can be collapsed into a single linear transformation.

Karklin and Lewicki (2003) [19] model higher-order dependencies among ICA units in a somewhat different fashion. They add a second hidden layer which is essentially another ICA basis, not for the outputs of the first layer, but for the variance (technically, for a dispersion parameter) in the outputs of the first layer. This introduces the nonlinearity between successive applications of ICA in a much more general fashion. It also allows the second layer to capture very coarse information about the image patch. Many of the higher-order basis functions can be interpreted as longer lines and edges, but there are also many units that appear to be capturing certain textural properties. Again, this procedure could presumably be iterated.

Fleuret and D. Geman (2001) [7] use decision trees arranged in a coarse-to-fine hierarchy for face detection in cluttered backgrounds. Interestingly, although the work is motivated by computational concerns and the learning is supervised, their feature selection procedure is quite similar to ours. In particular, they combine low-level features into higher level features exactly when the low-level features are strongly correlated (i.e., when they are a suspicious coincidence) on objects of interest (in this case, faces). They provide a much more rigorous mathematical framework than we do for investigating such hierarchies of suspicious coincidences and they also use the resulting representation for a difficult object detection task. In future work we hope to further investigate connections between the two approaches.

3.8 Discussion

We developed some heuristics based on detecting sparse collections of suspicious coincidences which allow us to discover simple hierarchical features in binary natural images. The results are not surprising: there is a strong signal for detecting lines and edges. The method has several problems which prevent us from using it recursively to explore the possibility that

larger and more interesting structure could be discovered.

One of the striking deficiencies in the method described here is the lack of invariance. Although we have used some location invariance to speed learning and reduce dimensionality, we are still constrained to the 3×3 grid. This means that every feature will have at least 9 different copies that need to be learned. Furthermore, the model only operates at a single scale and there is no invariance to illumination or rotation or deformation. These types of invariance are likely to be important in any realistic composition system. The lack of them will necessitate an explosion of features, all slightly different, that would not occur if these differences were captured by the appropriate invariance. Even in the results reported here it is possible to see how the representation is growing too fast and becoming noisy partly because of the lack of invariance. We hope to be able to incorporate more types of invariance in later versions of these ideas. In Chapter 4 we demonstrate that invariant versions of these features can also be learned using a similar learning heuristic.

Another important improvement will be a more robust sparsening procedure. The method used here, which is based on comparing pixel-level representations of higher-order structure, becomes problematic once the high-level representations have a lot of invariance. Using a statistical distance seems like an obvious next step and we will have to investigate this along with invariance. Some type of local maxima procedure will likely still be important. The number of compositions quickly becomes too large to exhaustively search as we have done here. A local gradient method for good suspicious coincidences might simultaneously search the space and create sparseness.

One of the long term goals is model updating. After a new level of structure has been discovered, the model should be updated to take into account these new dependencies. In the updated model, this structure will no longer be unusual and higher-level dependencies can be investigated. This will presumably lead to a better probabilistic description of images and make possible improved image processing algorithms – the whole point of this endeavor. Early indications suggest that incorporating invariance will involve many of the same technical issues as model updating, so we may be trying to solve all of these problems simultaneously.

Bibliography

[1] Horace Barlow. What is the computational goal of the neocortex? In Christof Koch and Joel L. Davis, editors, *Large-Scale Neuronal Theories of the Brain*, pages 1–22. MIT Press, Cambridge, 1994.

[2] Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October 1998.

[3] Anthony J. Bell and Terrence J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[4] Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 46:1–38, 1977.

[6] A. Desolneux, L. Moisan, and J.-M. Morel. Computational gestalts and perception thresholds. *Journal of Physiology - Paris*, 97(2–3):311–322, 2003.

[7] François Fleuret and Donald Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41:85–107, 2001.

[8] P. Földiák. Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, 64:165–170, 1990.

[9] J. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.

[10] W.S. Geisler, J.S. Perry, B.J. Super, and D.P. Gallogly. Edge co-occurrence in natural images predicts contour grouping performance. *Vision Research*, 41:711–724, 2001.

[11] D. Geman and A. Koloydenko. Invariant statistics and coding of natural microimages. In *Proceedings, IEEE Workshop on Statistical and Computational Theories of Vision*, Fort Collins, CO, June 1999.

[12] Geoffrey E. Hinton. Products of experts. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 1, pages 1–6, Edinburgh, U.K., 1999.

[13] Patrik O. Hoyer and Aapo Hyvärinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42:1593–1605, 2002.

[14] Shih-Hsiu Huang. *Compositional approach to recognition using multi-scale computations*. PhD thesis, Division of Applied Mathematics, Brown University, 2001.

[15] Peter Huber. Projection pursuit. *Annals of Statistics*, 13(2):435–475, 1985.

[16] A. Hyvärinen, P.O. Hoyer, and J. Hurri. Extensions of ICA as models of natural images and visual processing. In *Proceedings of the International Symposium on Independent Component Analysis and Blind Source Separation (ICA2003)*, pages 963–974, Nara, Japan, 2003.

[17] Aapo Hyvärinen and Patrik O. Hoyer. A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41:2413–2423, 2001.

[18] Aapo Hyvärinen, Patrik O. Hoyer, and Mika Inki. Topographic independent component analysis. *Neural Computation*, 13:1527–1558, 2001.

[19] Yan Karklin and Michael S. Lewicki. Learning higher-order structures in natural images. *Network: Computation in Neural Systems*, 14:483–499, 2003.

[20] Guy Mayraz and Geoffrey E. Hinton. Recognizing handwritten digits using hierarchical products of experts. *IEEE Transactions on Pattern Analysis and Machine Vision*, 24(2):189–197, February 2002.

[21] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[22] C. G. Phillips, S. Zeki, and H. B. Barlow. Localization of function in the cerebral cortex: past, present and future. *Brain*, 107(1):327–361, March 1984.

[23] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 1997.

[24] Daniel Frederic Potter. *Compositional Pattern Recognition*. PhD thesis, Division of Applied Mathematics, Brown University, 1999.

[25] Stefan Roth and Michael J. Black. Field of experts: A framework for learning image priors with applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005 (submitted).

[26] M. F. Tappen, B. C. Russell, and W. T. Freeman. Efficient graphical models for processing images. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 673–680, Washington, DC, 2004.

[27] J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London B*, 265:359–366, 1998.
`http://hlab.phys.rug.nl/archive.html`.

[28] Max Welling, Richard S. Zemel, and Geoffrey E. Hinton. Probabilistic independent components analysis. *IEEE Transactions on Neural Networks*, 15(4):838–849, July 2004.

[29] Song Chun Zhu and David Mumford. Prior learning and gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250, November 1997.

[30] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, November 1997.

[31] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.

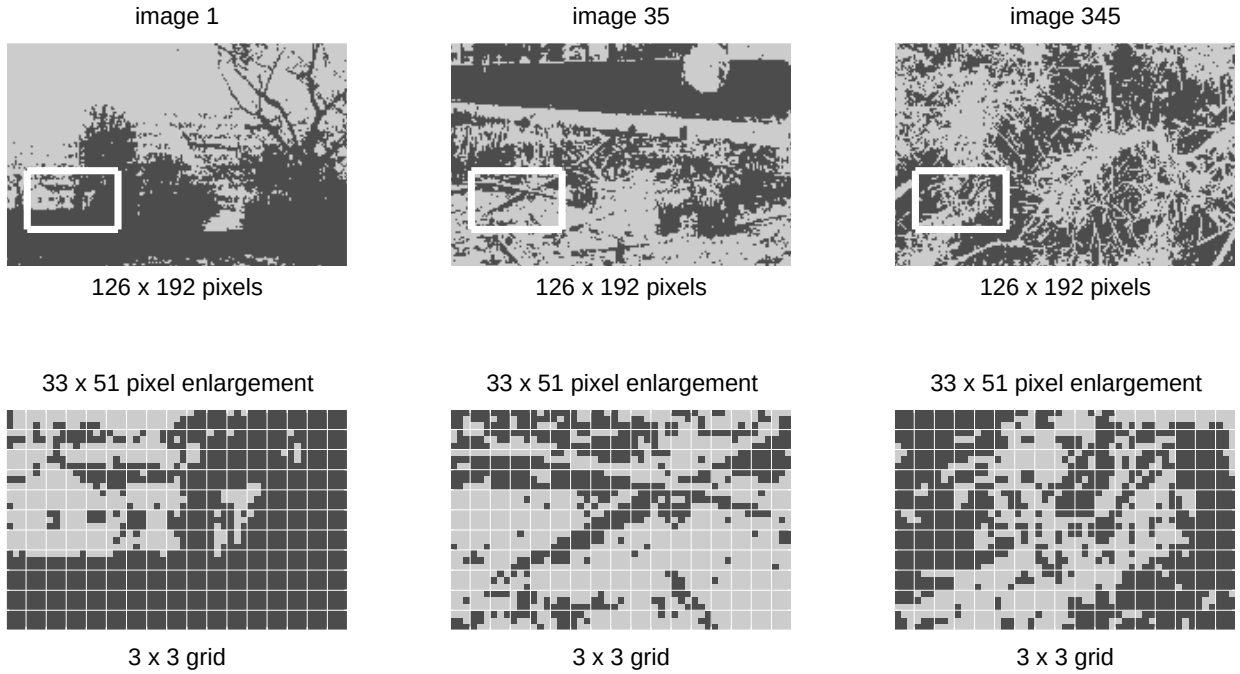

Figure 3.1: Sample images with enlargements [27].

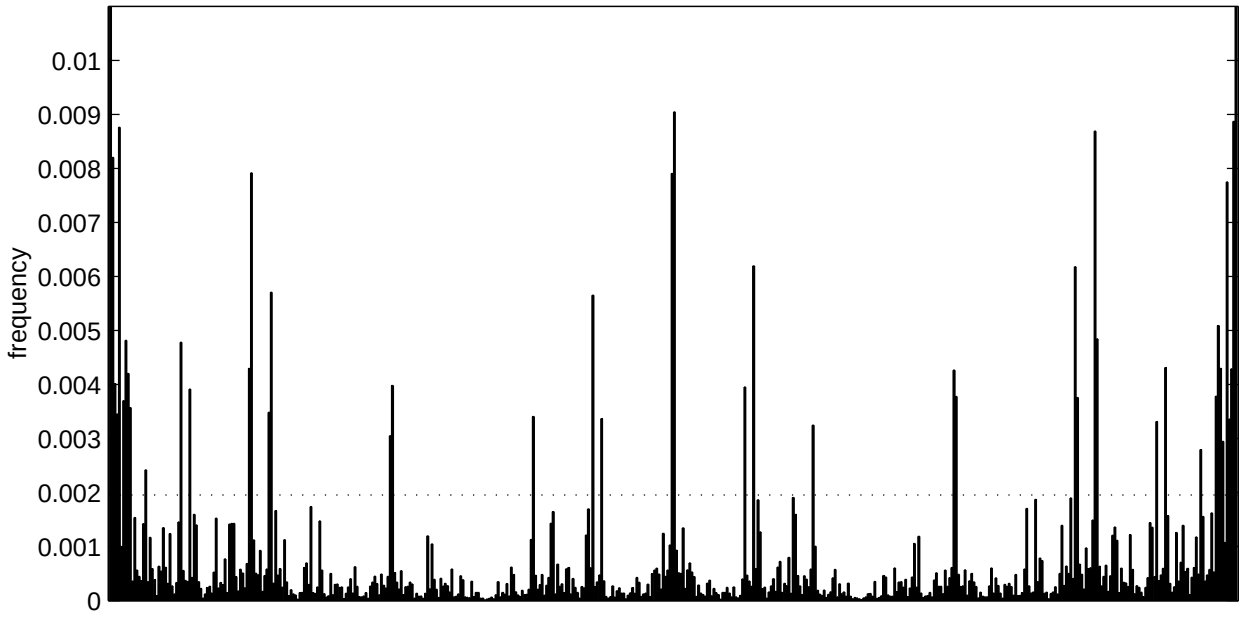

Figure 3.2: The empirical probability distribution of all 3×3 binary patches ($2^9 = 512$ total). The dotted line is $1/512$. The entropy of this distribution is 4.65 bits (uniform is 9 bits).

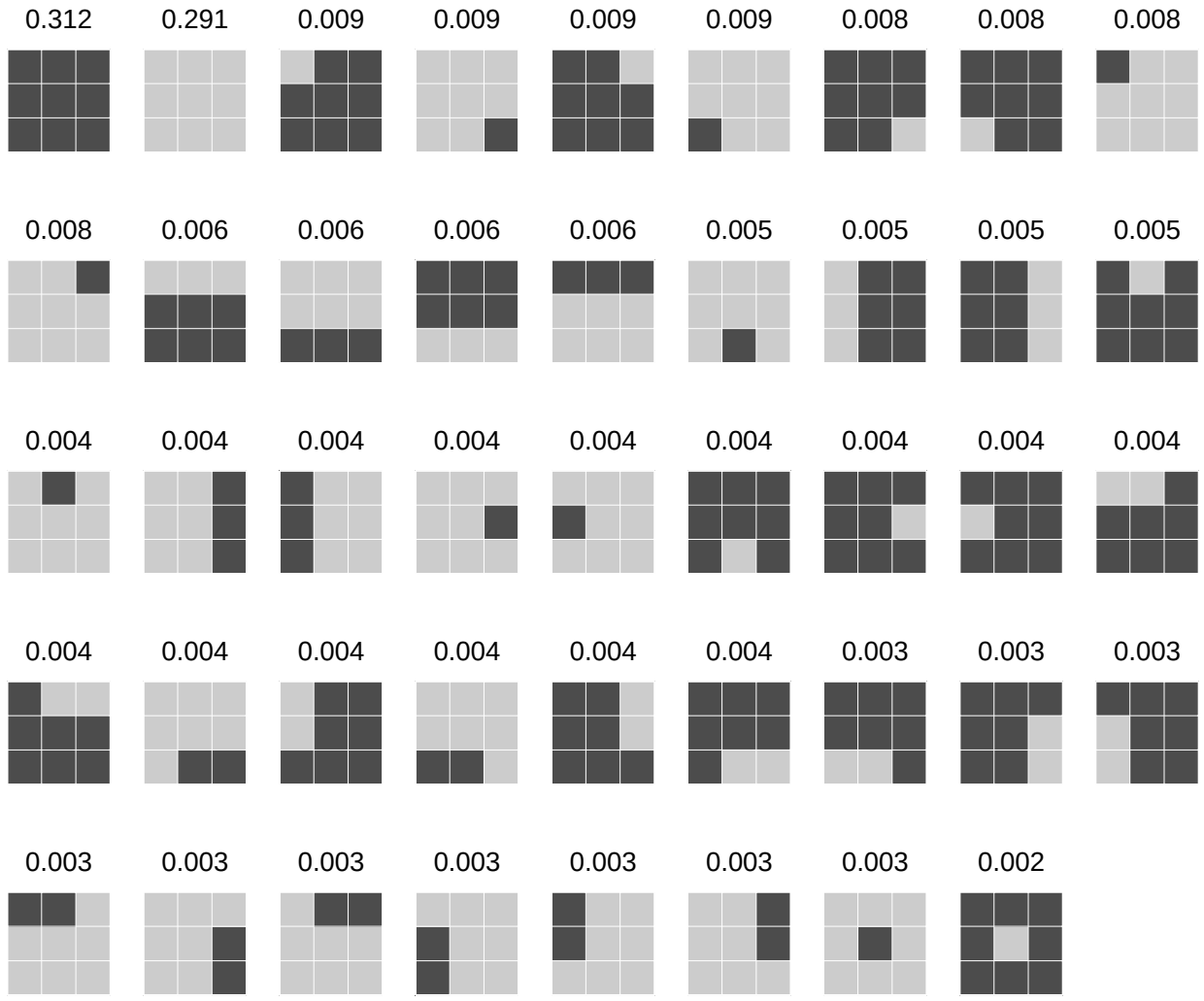

Figure 3.3: The 44 images patches that were suspicious coincidences and their empirical probabilities.

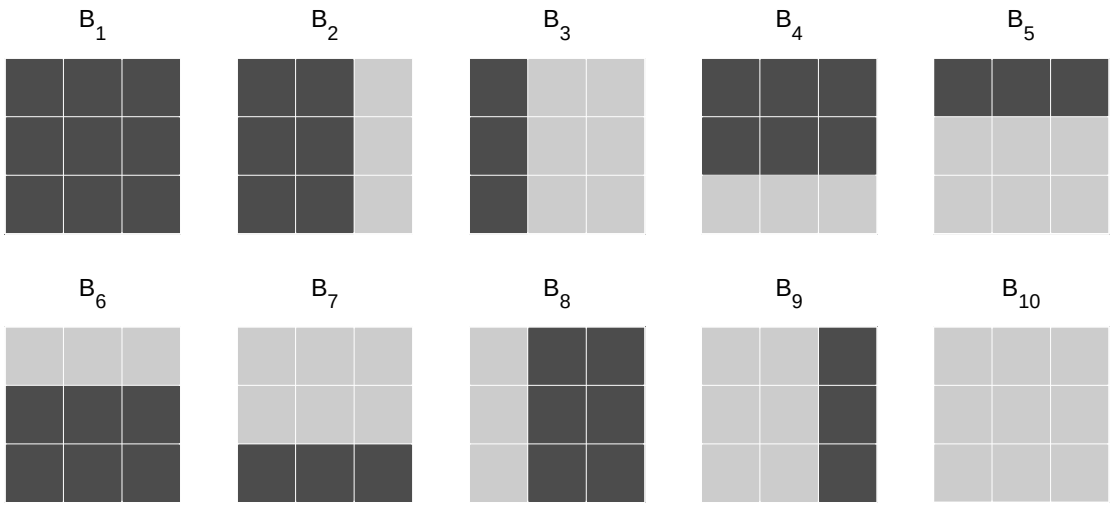

Figure 3.4: The 10 representative patches B_s found by sparsening the suspicious coincidences in Figure 3.3.

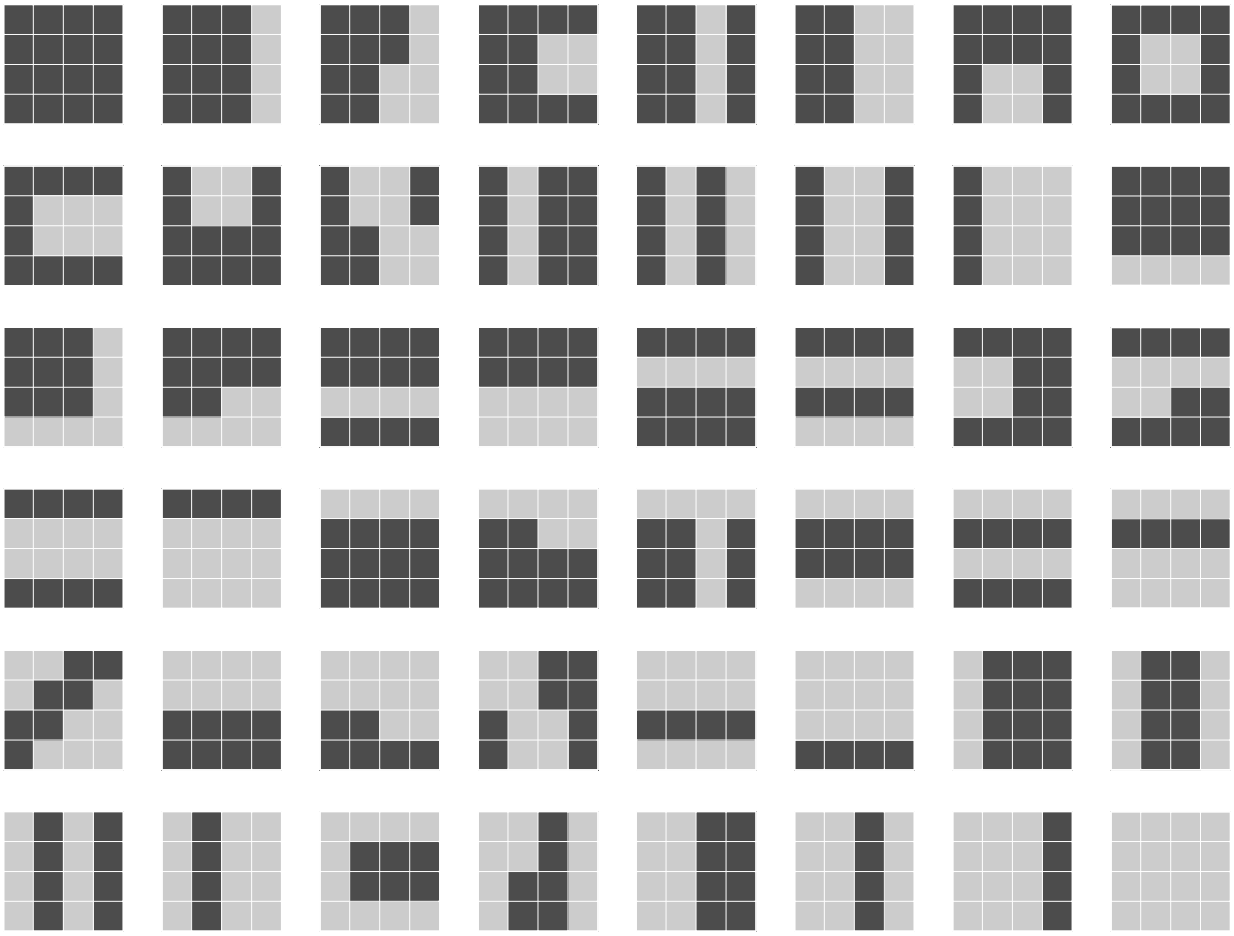

Figure 3.5: The 48 representative 4×4 patches found in binary images using the methods of Section 3.2.1. These are not used in this chapter, but a similar set derived from a slightly different data set is used in Chapter 4.

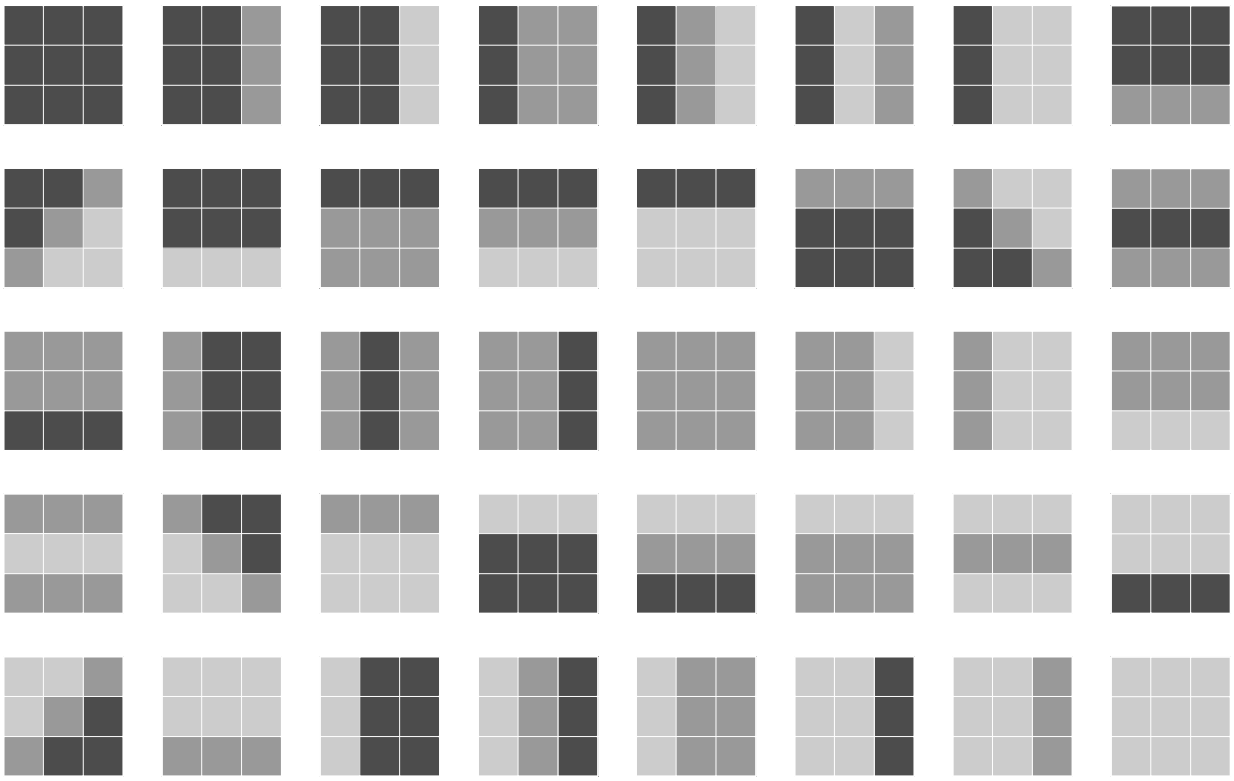

Figure 3.6: The 40 representative 3×3 patches found in *ternary* images using the methods of Section 3.2.1. These are not used here.

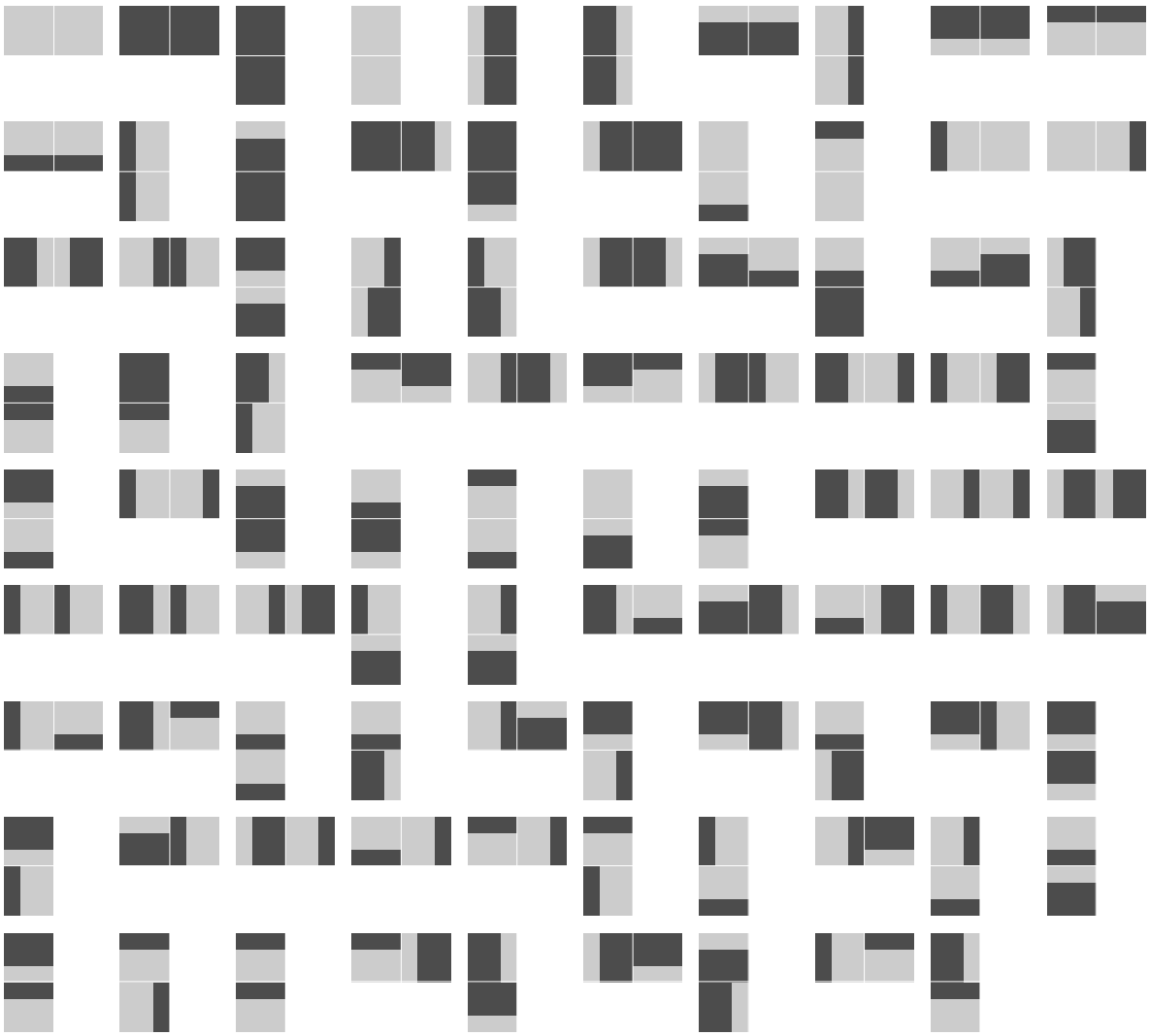

Figure 3.7: The 89 suspicious coincidences composed of two neighboring representative patches B_s. MDL ranked in decreasing order from left to right.

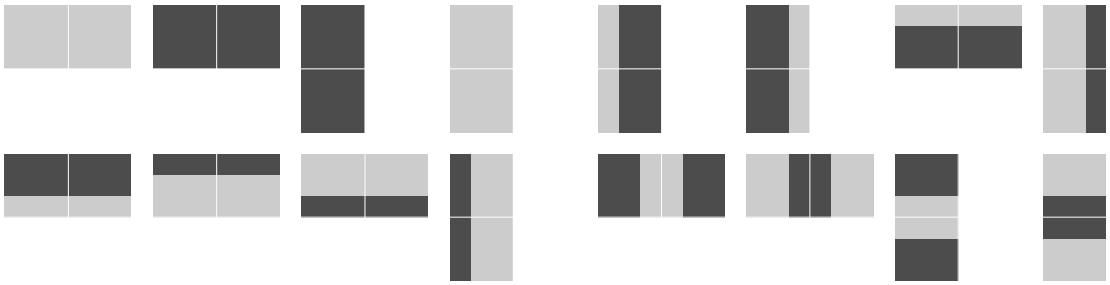

Figure 3.8: The 16 representative 2-patch compositions, found by sparsening the representation in Figure 3.7. MDL ranked in decreasing order from left to right.

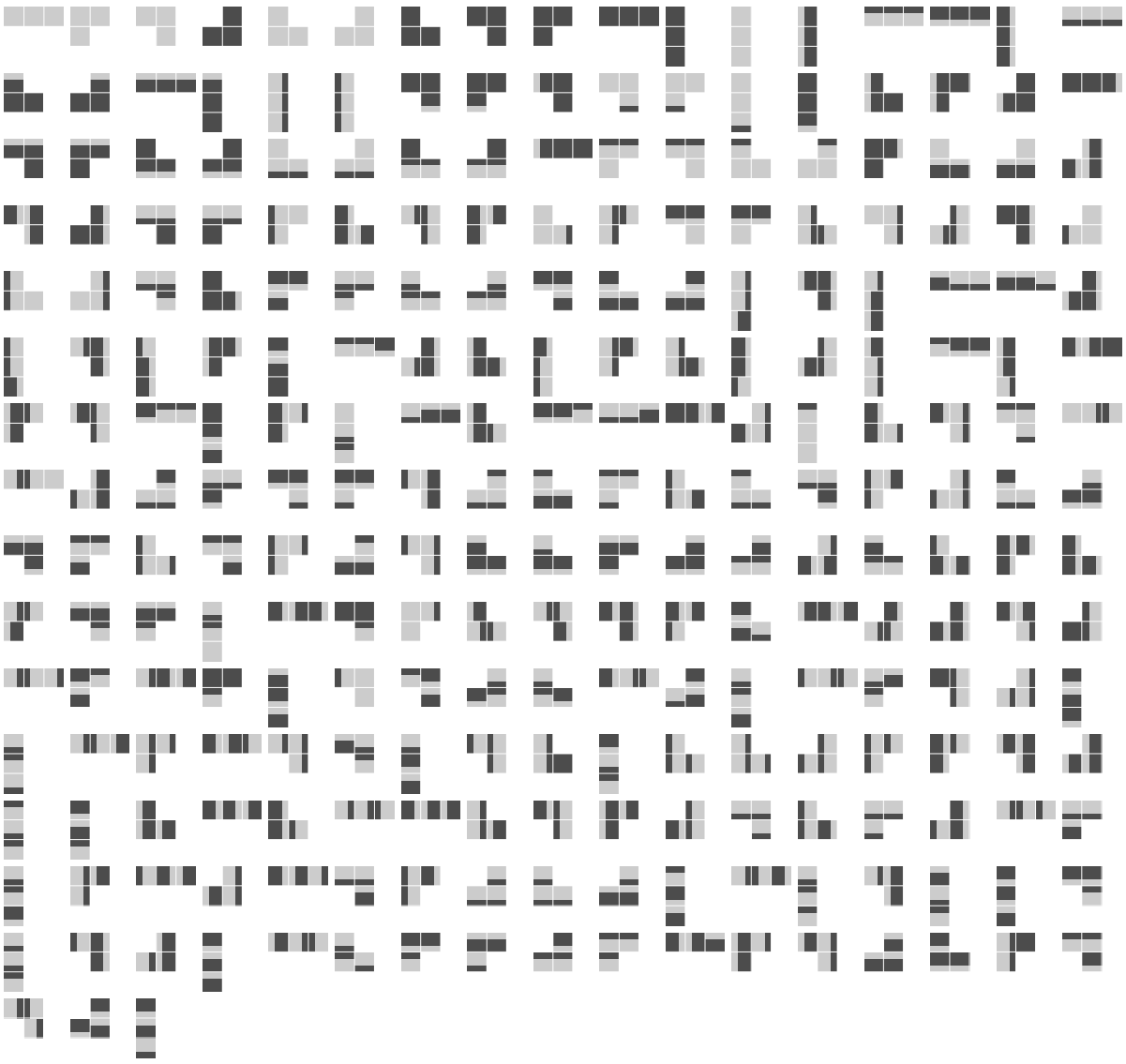

Figure 3.9: The 258 suspicious coincidences (after removing exact pixel repeats) made with three neighboring representative patches B_s and formed by composing one of the single patches in Figure 3.4 with one of the double patches in Figure 3.8. MDL ranked in decreasing order from left to right.

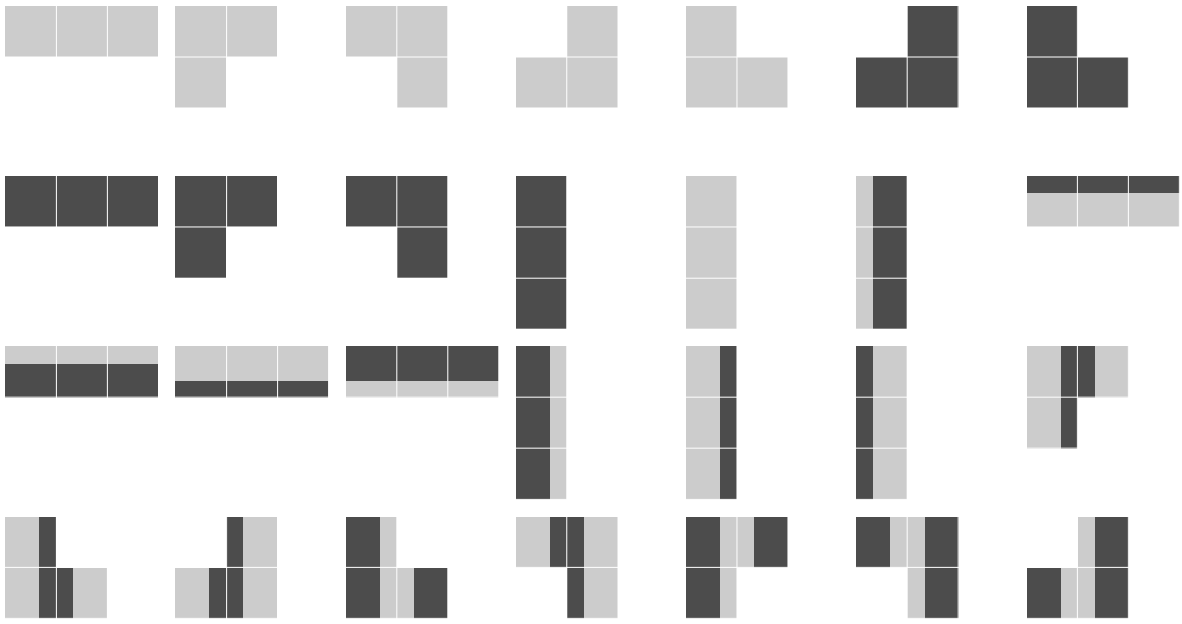

Figure 3.10: The 28 representative 3-patch compositions, found by sparsening the representation in Figure 3.9. MDL ranked in decreasing order from left to right.

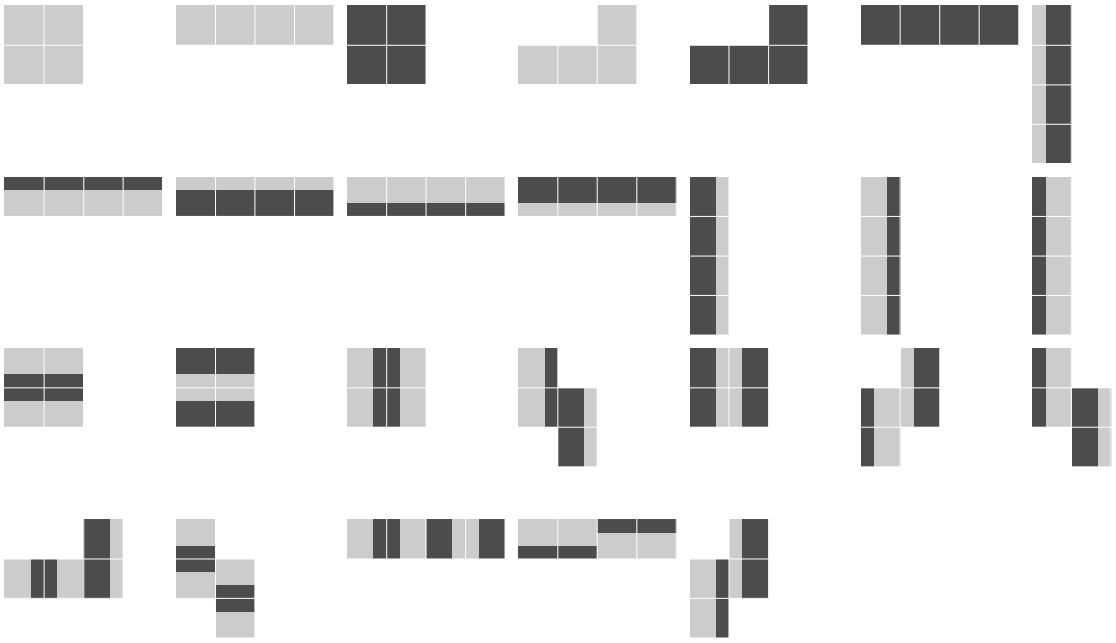

Figure 3.11: The 26 sparsened 4-patch compositions, MDL ranked in decreasing order from left to right.

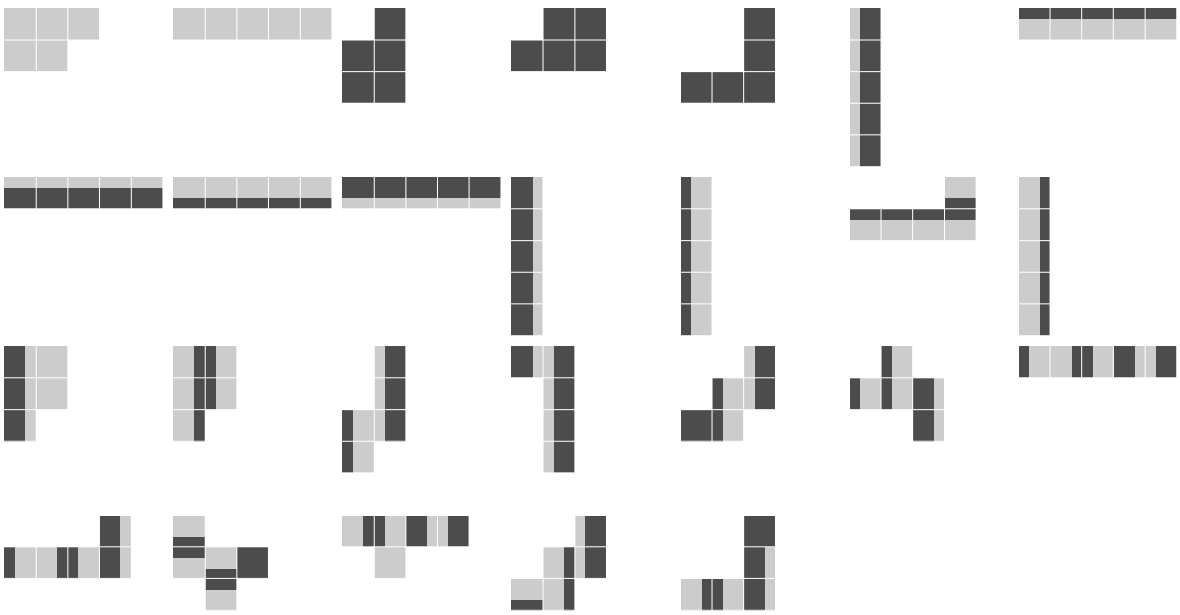

Figure 3.12: The 26 sparsened 5-patch compositions, MDL ranked in decreasing order from left to right.

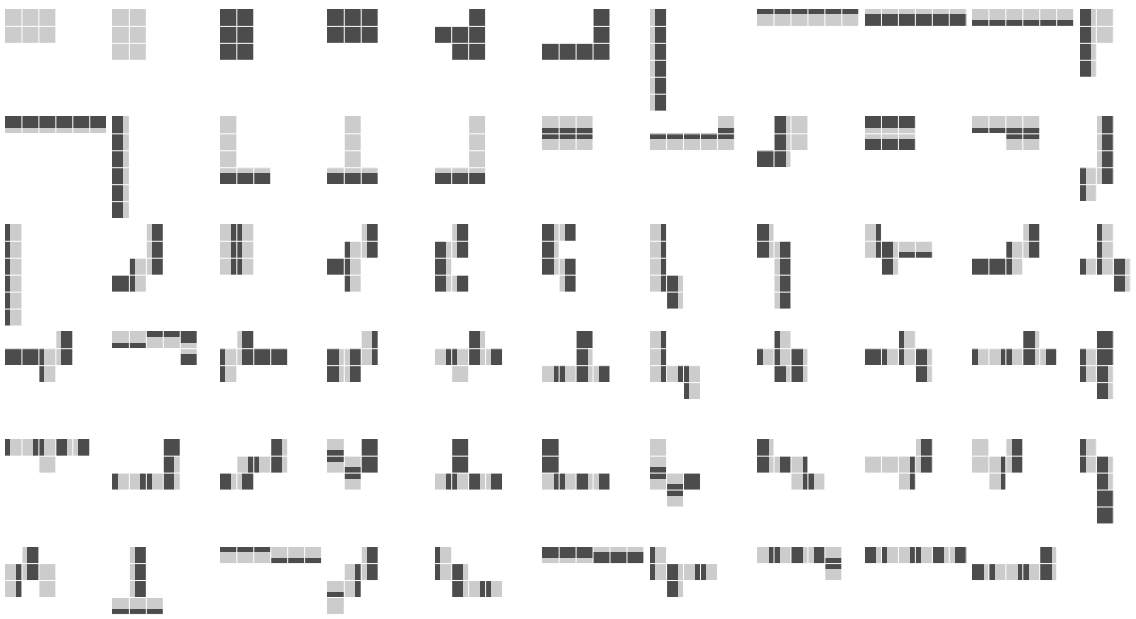

Figure 3.13: The 65 sparsened 6-patch compositions, MDL ranked in decreasing order from left to right. At this level the representation becomes noisy, but this can be remedied with more training examples.

Figure 3.14: The 81 sparsened 7-patch compositions, MDL ranked in decreasing order from left to right.

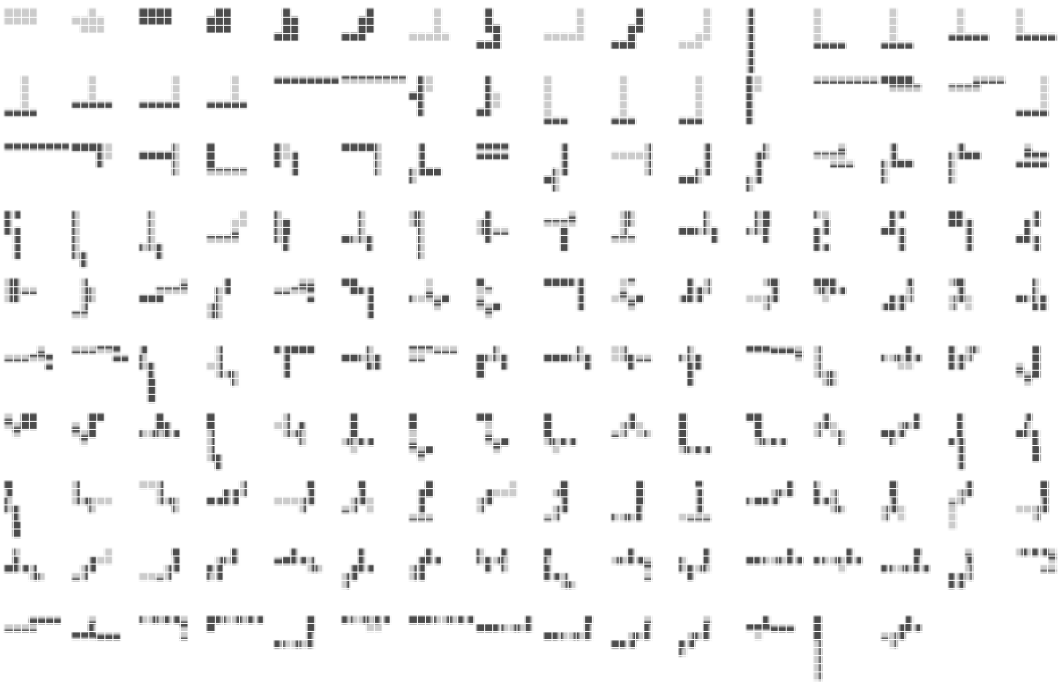

Figure 3.15: The 158 sparsened 8-patch compositions, MDL ranked in decreasing order from left to right.