Convolutional Dictionary Learning and Feature Design

Lawrence Carin

Duke University

16 September 2014

1. Background
2. Convolutional Dictionary Learning
3. Hierarchical, Deep Architecture
4. Convolutional Neural Network
5. Summary

Outline

1. Background
2. Convolutional Dictionary Learning
3. Hierarchical, Deep Architecture
4. Convolutional Neural Network
5. Summary

Motivation

- In sensing systems, the representation used for the incoming data may significantly impact performance
- This is typically termed feature extraction, which is often performed with hand-crafted features
- Ideally the data representation is performed jointly with the learning task
- Much recent success using "deep" architectures, and convolutional neural networks
- Have made connections to recent ideas in dictionary learning and sparse signal representations, with goal of demystifying

Dictionary Learning

- There has been significant recent interest in dictionary learning for image representation
- Consider an image, which we represent via the matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$
- The overall, large image is represented by a set of patches $\{\boldsymbol{x}_i\}_{i=1,N}$, which may be overlapping

Dictionary Learning: Formulation

- Given a set of image patches $\{\boldsymbol{x}_i\}_{i=1,N}$, we may infer a dictionary representation

$$\boldsymbol{x}_i = \mathbf{D}\boldsymbol{w}_i + \boldsymbol{\epsilon}_i$$

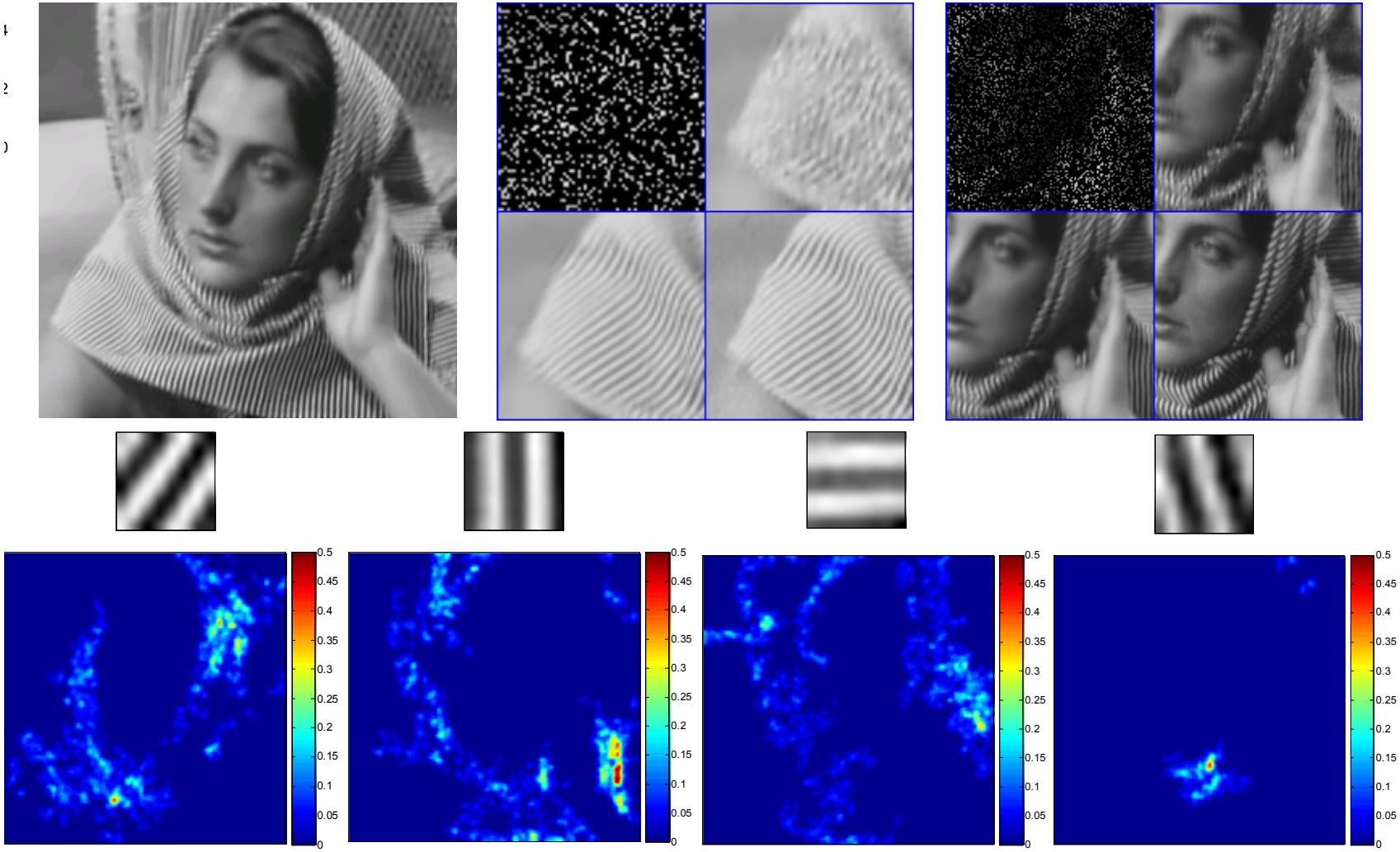
  where $\boldsymbol{x}_i \in \mathbb{R}^p$, $\mathbf{D} \in \mathbb{R}^{p \times K}$, and $\boldsymbol{w}_i \in \mathbb{R}^K$ is sparse

- Typically the dictionary is over-complete, meaning $K > p$, and for images we often set $p = 8 \cdot 8 \cdot 3 = 192$

- Typical setup:

$$\hat{\mathbf{D}}, \{\hat{\boldsymbol{w}}_i\} = \text{argmin}_{\mathbf{D},\{\boldsymbol{w}_i\}} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \mathbf{D}\boldsymbol{w}_i\|_2^2 + \lambda_1 \sum_{k=1}^{K} \|\boldsymbol{d}_k\|_2^2 + \lambda_2 \sum_{i=1}^{N} \|\boldsymbol{w}_i\|_1$$

Inferred Dictionary and Activations



Outline

1. Background
2. Convolutional Dictionary Learning
3. Hierarchical, Deep Architecture
4. Convolutional Neural Network
5. Summary

Convolutional Dictionary Learning

- Using the patch representation yields many redundant dictionary elements, that are simply shifts of a basic dictionary type

- Now perform dictionary learning directly on the entire image $\mathbf{X}$:

$$\mathbf{X} = \sum_{k=1}^{K} \boldsymbol{d}_k * \mathbf{W}_k + \mathbf{E}$$

where $\boldsymbol{d}_k$ is again a dictionary element over a small patch size, but now $\mathbf{W}_k$ is a sparse activation map *over the entire image*

- Solution methodology

$$\hat{\mathbf{D}}, \{\hat{\mathbf{W}}_k\} = \text{argmin}_{\mathbf{D},\{\mathbf{W}_k\}} \|\mathbf{X} - \sum_{k=1}^{K} \boldsymbol{d}_k * \mathbf{W}_k\|_F^2 + \lambda_1 \sum_{k=1}^{K} \|\boldsymbol{d}_k\|_2^2 + \lambda_2 \sum_{k=1}^{K} \|\mathbf{W}_k\|_1$$

Efficient Implementation

$$\hat{\mathbf{D}}, \{\hat{\mathbf{W}}_k\} = \text{argmin}_{\mathbf{D},\{\mathbf{W}_k\}} \|\mathbf{X} - \sum_{k=1}^{K} \boldsymbol{d}_k * \mathbf{W}_k\|_F^2 + \lambda_1 \sum_{k=1}^{K} \|\boldsymbol{d}_k\|_2^2 + \lambda_2 \sum_{k=1}^{K} \|\mathbf{W}_k\|_1$$

- By moving away from the patched-based solution, we remove many redundant dictionary elements which are shifted and clipped versions of basic types
- By using FFTs, the convolution operation may be made fast
- We learn the dictionary $\hat{\mathbf{D}}$ "offline," based on many training images
- For a test image $\mathbf{D}$ is fixed, and we must solve

$$\{\hat{\mathbf{W}}_k\} = \text{argmin}_{\{\mathbf{W}_k\}} \|\mathbf{X} - \sum_{k=1}^{K} \boldsymbol{d}_k * \mathbf{W}_k\|_F^2 + \lambda_2 \sum_{k=1}^{K} \|\mathbf{W}_k\|_1$$

- Still relatively expensive at test time; we return to this

Outline

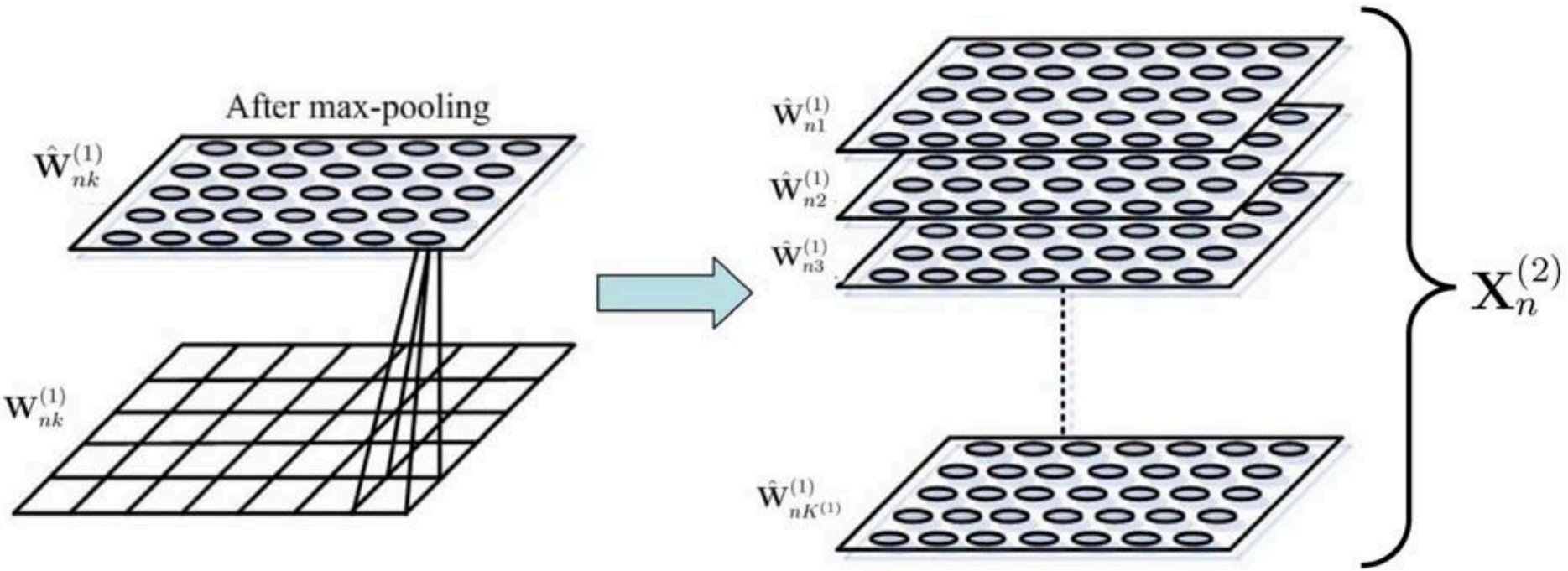
1. Background
2. Convolutional Dictionary Learning
3. Hierarchical, Deep Architecture
4. Convolutional Neural Network
5. Summary

Multiscale Convolutional Dictionary Learning

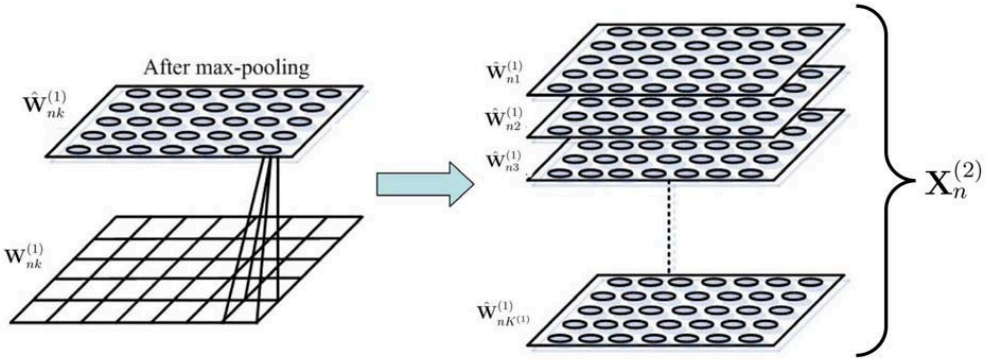
$$\hat{\mathbf{D}}, \{\hat{\mathbf{W}}_k\} = \text{argmin}_{\mathbf{D},\{\mathbf{W}_k\}} \|\mathbf{X} - \sum_{k=1}^{K} \boldsymbol{d}_k * \mathbf{W}_k\|_F^2 + \lambda_1 \sum_{k=1}^{K} \|\boldsymbol{d}_k\|_2^2 + \lambda_2 \sum_{k=1}^{K} \|\mathbf{W}_k\|_1$$

- The mapping $\mathbf{X} \to \{\mathbf{W}_k\}$ constitutes a set of K feature (dictionary activation) maps
- The scale of these features are dictated by the size of the dictionary elements $\boldsymbol{d}_k$
- We can use the feature "stack" $\{\mathbf{W}_k\}_{k=1,K}$ as new input data, and then we can perform dictionary learning on these features
- Useful to perform "pooling" when going to the next layer, to increase the effective size of the dictionary elements at the next level in the hierarchy

Multiscale Convolutional Dictionary Learning



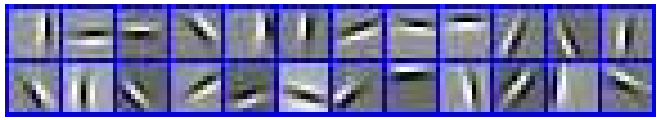
Unsupervised Feature Learning



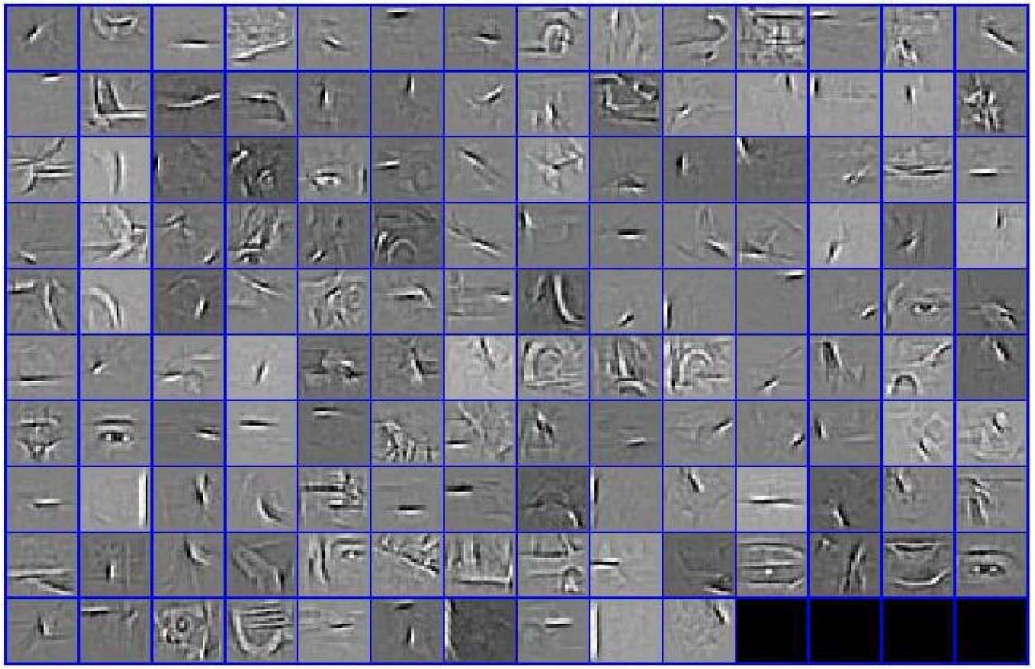
- The multiscale dictionary elements may be learned in an unsupervised manner
- Allows leveraging of massive quantities of unlabeled, but relevant data
- Fine-scale tuning may be performed subsequently, using labels or rewards (in the RL case)
- Connect label/action/policy to features at the top of the hierarchy

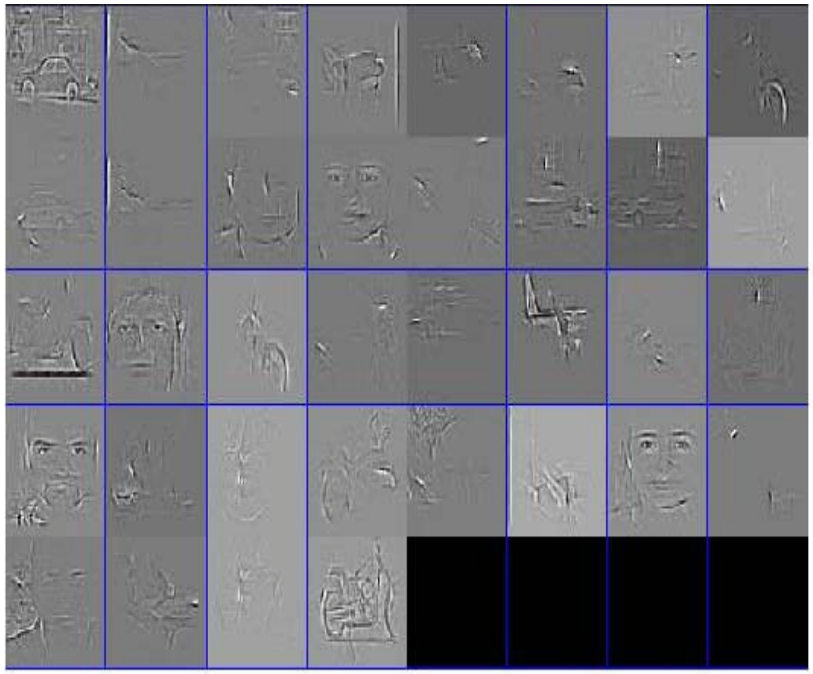
Learned Convolutional Dictionary Elements

- Applied to Caltech 101 database
- Layer 1 dictionary elements:



- Second (left) and third (right) layer dictionary elements:





Outline

1. Background
2. Convolutional Dictionary Learning
3. Hierarchical, Deep Architecture
4. Convolutional Neural Network
5. Summary

Fast Approximate Analysis

- Recall that, for a test $\mathbf{X}$, using previously learned convolutional dictionary $\mathbf{D}$, we must solve

$$\{\hat{\mathbf{W}}_k\} = \text{argmin}_{\{\mathbf{W}_k\}} \|\mathbf{X} - \sum_{k=1}^{K} \boldsymbol{d}_k * \mathbf{W}_k\|_F^2 + \lambda_2 \sum_{k=1}^{K} \|\mathbf{W}_k\|_1$$

- Relatively expensive

- Instead implement a convolutional filterbank followed by a pointwise nonlinear function

$$\hat{g}(\cdot), \hat{\boldsymbol{f}}_k = \text{argmin}_{g,\boldsymbol{f}_k} \|\mathbf{W}_k - g(\mathbf{X} * \boldsymbol{f}_k)\|_F$$

- In practice we have a series of convolutional filterbanks, followed by a nonlinear function, and then a pooling operation

- The pooling plays an important role in achieving robustness, and accuracy of the above approximation

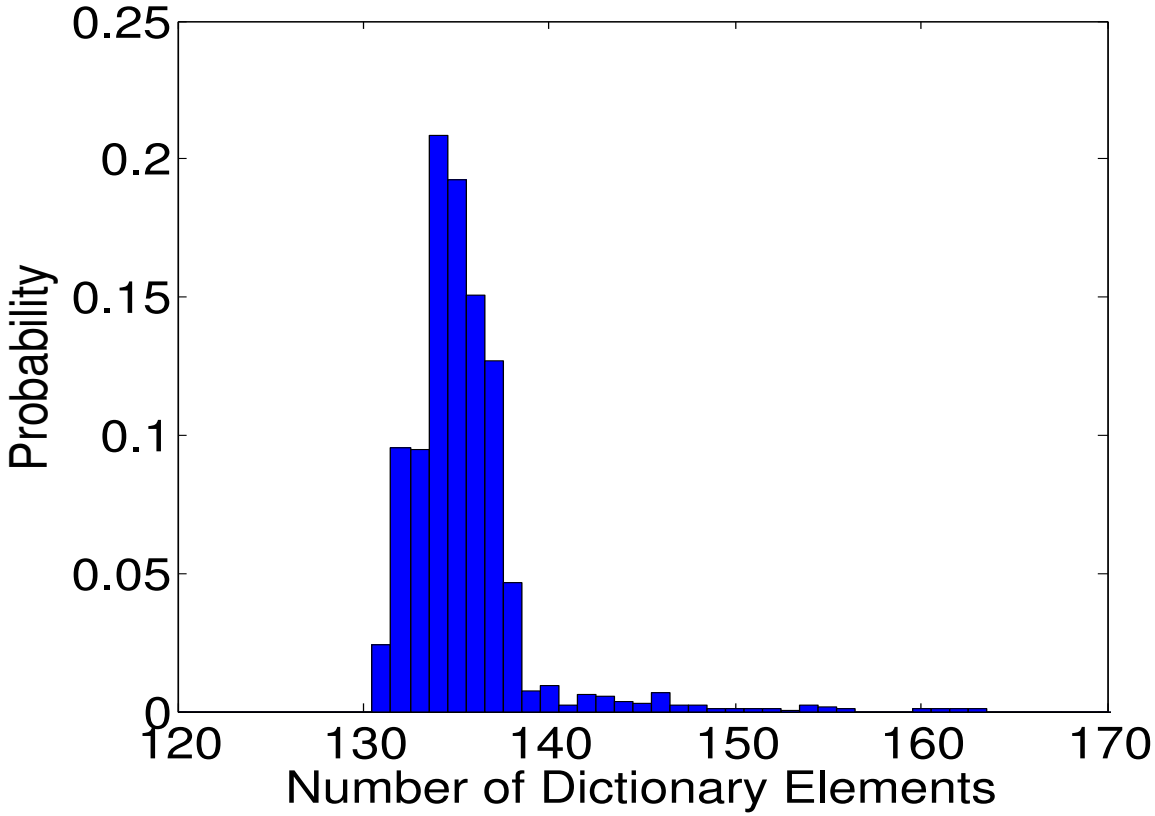
Our Implementation Methodology

- When learning the model architecture, sparsity plays a key role
- Have utilized a structure related to adaptive Lasso, via

$$
\begin{aligned}
w_j &\sim \frac{1}{2}\sqrt{\tau/\gamma_j}\exp(-|w_j|\sqrt{\tau/\gamma_j}) \\
&\sim \int \mathcal{N}(w_j; 0, \tau^{-1}\alpha^{-1})\mathsf{InvGa}(\alpha; 1, (2\gamma_j)^{-1})d\alpha
\end{aligned}
$$

- Impose sparsity promotion via a heavy-tailed gamma process on γ_j
- Bayesian setup used to infer the number of filterbanks at each level of the hierarchy

Bayesian Inferred Dictionary Usage



Outline

1. Background
2. Convolutional Dictionary Learning
3. Hierarchical, Deep Architecture
4. Convolutional Neural Network
5. Summary