
Sparse Coding on Stereo Video for Object Detection

Sheng Y. Lundquist
Portland State University
New Mexico Consortium
shenglundquist@gmail.com

Melanie Mitchell
Portland State University
Santa Fe Institute

Garrett T. Kenyon
Los Alamos National Laboratory
New Mexico Consortium

Abstract

Deep Convolutional Neural Networks (DCNN) require millions of labeled training examples for image classification and object detection tasks, which restrict these models to domains where such datasets are available. In this paper, we explore the use of unsupervised sparse coding applied to stereo-video data to help alleviate the need for large amounts of labeled data. We show that replacing a typical supervised convolutional layer with an unsupervised sparse-coding layer within a DCNN allows for better performance on a car detection task when only a limited number of labeled training examples is available. Furthermore, the network that incorporates sparse coding allows for more consistent performance over varying initializations and ordering of training examples when compared to a fully supervised DCNN. Finally, we compare activations between the unsupervised sparse-coding layer and the supervised convolutional layer, and show that the sparse representation exhibits an encoding that is depth selective, whereas encodings from the convolutional layer do not exhibit such selectivity. These results indicate promise for using unsupervised sparse-coding approaches in real-world computer vision tasks in domains with limited labeled training data.

1 Introduction

Over the last decade, Deep Convolutional Neural Networks (DCNN) trained with supervised learning have emerged as the dominant paradigm for computer vision. These networks have shown impressive results on computer vision tasks such as image labeling and object detection (e.g., [18, 20]). However, one drawback of DCNNs is that they rely on large collections of training data that have been annotated by humans, e.g., ImageNet [3]. As such, DCNNs are restricted to domains for which large datasets of labeled examples are available.

We explore the use of unsupervised sparse coding within a supervised DCNN to help alleviate the need for an abundance of training labels. Typically, the encodings (defined as the collection of activation values calculated from a layer) in DCNNs are computed via convolutions followed by a nonlinearity, and for a given task the weights are trained via supervised learning. In contrast, sparse coding [15] aims to infer efficient, non-redundant encodings of a given input (e.g., photographs and videos), and the weights are trained via an unsupervised method. Inspired by theories of efficient coding in neural computation [1], sparse coding has been shown to exhibit similar properties to biological neurons in early stages of mammalian visual processing [15, 22]. Modeling the accuracy and precision of biology in visual perception could provide novel insights into computer vision tasks, such as object detection. Additionally, the efficiency in representation exhibited by sparse coding can be advantageous in specialized, non-Von Neumann hardware architectures. Indeed, there has been work in implementations of sparse coding on low-power neuromorphic [9] and quantum [13] hardware.

One domain in which non-redundant encoding should be useful is in multi-view sensing. An efficient encoding should account for correlated offsets between different views of visual features. These

offsets represent disparity in stereo images and optic flow in consecutive frames from a video. It follows that an encoding that accounts for such offsets should have some notion of depth [5, 16].

In this paper, we compare two types of convolutional network models that differ only in the first layer: (1) a *sparse-coding* network, in which the weights and activations in the first layer are computed via unsupervised sparse coding, and (2) a *supervised* network, in which the first-layer weights are learned via supervised training and activations are computed using these layer weights. In both network models, the weights and activations in all subsequent layers are computed via supervised convolutional layers. We show that sparse-coding networks is able to achieve better performance on a vehicle detection task on stereo-video data with a minimal amount of training labels. Additionally, we show that the performance of sparse-coding networks is more consistent—i.e., more robust to randomized order of training data and random initializations—than comparable supervised networks. Finally, we show that activations in the first (sparse-coding) layer in our sparse-coding networks are depth selective, which may provide an explanation for the differences in performance we observe in this study.

1.1 Related Work

Lundquist et al. [11] demonstrated that representations of stereo images obtained through sparse coding allow for an encoding that achieves better performance than a convolutional layer in the task of pixel-wise depth estimation. The authors show that the sparse encoding is inherently depth selective, whereas the convolutional encoding is not. In this paper, we extend this work to encode stereo-video clips and compare encodings on a vehicle-detection task.

A closely related study by Coates et al. [2] compared unsupervised and supervised methods with two-layer networks on an image classification task. The authors demonstrated that an unsupervised layer does not outperform a comparable layer with a convolutional encoding. While our experimental results agree with this finding in the case of two layers, we find that limiting the number of labeled training examples or adding an additional supervised layer in the sparse-coding network allows the network to outperform the supervised network.

Recent work by Lotter et al. [10] shares the motivation of utilizing unsupervised learning to alleviate the need for labeled training data. Specifically, the authors use unsupervised learning to predict future frames of a video. They additionally show that their network achieves better performance than standard DCNNs when each is trained on only a limited amount of training data. In contrast to future frame prediction, our work aims to achieve image representation through sparse coding. Additionally, Lotter et al. uses a recurrent neural network [7] as the backbone of their network, whereas we use sparse coding as the unsupervised learning algorithm.

Other work [4, 17] explores the use of unsupervised learning techniques within a supervised network. However, most work in this area does not explore natural scenes (instead, focusing on datasets such as MNIST for handwritten digit recognition). Here, we extend this work to the domain of stereo video captured “in the wild”. Additionally, we explicitly compare performance between the use of unsupervised learning versus supervised learning within a DCNN.

There has been other work in unsupervised learning of multi-view data [8, 12, 14]. In contrast, our work aims to explicitly compare unsupervised learning to supervised learning for the task of vehicle detection in multi-view data.

2 Sparse Coding

Sparse coding aims to represent an input (e.g., an image) as a linear combination of basis vectors drawn from a provided set (referred to as a *dictionary*) such that the original signal is recoverable with minimal degradation. Each basis vector is weighted by a scalar coefficient (referred to as an *activation*), and the set of activations are taken to be the encoding of the input. Sparse coding constrains the activations to be sparse (i.e., to have few nonzero activations), such that resulting activations are non-redundant. Here, inferring the sparse set of activations is an optimization problem, unlike encodings calculated via supervised convolutional layers.

Formally, sparse coding aims to minimize the cost function

$$J(\mathbf{I}, \mathbf{a}, \Phi) = \frac{1}{2} \overbrace{(\|\mathbf{I} - \mathbf{a} \otimes \Phi\|_2)^2}^{\text{Reconstruction error}} + \lambda \overbrace{\|\mathbf{a}\|_1}^{\text{Sparsity term}}. \quad (1)$$

Specifically, the algorithm aims to minimize the difference between a given input \mathbf{I} and a reconstruction, where the difference is measured by Euclidean distance (i.e., $\|\cdot\|_2$, or the L_2 norm). The reconstruction $\mathbf{a} \otimes \Phi$ is calculated via a linear combination of basis vectors drawn from a dictionary Φ , weighted by activation coefficients \mathbf{a} . Here, \otimes denotes the transposed convolution operation [21]. The sparsity term constrains the activations \mathbf{a} to be sparse, by measuring the sum of the absolute value of \mathbf{a} (i.e., $\|\cdot\|_1$, or the L_1 norm)¹. λ is a hyperparameter that controls the trade-off between the reconstruction error and sparsity.

The process of sparse coding that minimizes Equation 1, given a training set, is broken into two parts: (1) encoding an input by finding a set of activations \mathbf{a} for input \mathbf{I} , and (2) learning a set of basis vectors (i.e., a dictionary) Φ for the dataset. Encoding input \mathbf{I} involves inferring activations \mathbf{a} by minimizing Equation 1 with respect to \mathbf{a} while holding Φ fixed, for which we use the biologically informed Locally Competitive Algorithm (LCA) [19]. The final activations \mathbf{a} are taken to be the output activations of the corresponding input \mathbf{I} . A sparse-coding layer can replace a convolutional layer in a DCNN by using activations \mathbf{a} as the output of the layer. Learning a dictionary for sparse coding, analogous to learning filter weights in a DCNN, involves minimizing the cost function with respect to Φ while holding \mathbf{a} fixed via backpropagation of Equation 1. In the domain of images, dictionaries learned from sparse coding tend to represent oriented edges [15]. In our method, the input is first encoded using LCA, which is followed by one gradient descent step of minimizing the cost function with respect to Φ while holding \mathbf{a} fixed. Basis vectors are normalized to have unit L_2 norm after each update. Updating Φ is repeated for multiple input batches until convergence.

3 Experiments

We compare fully supervised networks with networks incorporating an unsupervised sparse-coding layer by testing performance on a vehicle detection task using stereo video. We test the effect of training set size by varying the number of labeled training examples available to the network. We used the KITTI object detection dataset [6] for experiments. The dataset contains approximately 7000 examples, which we split into 6000 for training and 1000 for testing. Each example consists of three stereo frames ordered in time, with bounding box annotations for various objects in the left camera’s last frame as ground truth. We normalize the stereo-video inputs to have zero mean and unit standard deviation and we downsampled them to be 256×64 pixels. We concatenated stereo inputs such that the input contains six features, i.e., RGB inputs from both left and right cameras. We kept time in a separate dimension for three-dimensional convolutions (for convolutional layers) or transposed convolution (for sparse-coding layers) across the time, height, and width axes of the input.

We generated the ground truth for our task by sliding a 32×16 -pixel non-overlapping window across the left camera’s last frame. We considered a window to be a positive instance if the window overlaps with any part of a car, van, or truck bounding box provided by the ground truth. The final output of each network is the probability of a window containing a vehicle, for all windows in the frame. We use the cross entropy between the ground truth and estimated probabilities as the supervised cost function to train all supervised layers within the network.

We tested various encoding schemes along with various weight initializations for the first layer of n -layer networks, as follows:

- **ConvSup:** Convolutional encoding. Weights are initialized randomly and learned via supervised training for car detection.
- **SparseUnsup:** Unsupervised sparse encoding to learn activations and weights.
- **ConvRand:** Convolutional encoding. Weights are initialized randomly and are not updated. This gives a random-weight baseline for the networks.

¹The L_1 norm is used as a surrogate to the L_0 norm (i.e., the number of nonzero elements), as Equation 1 is non-convex with respect to \mathbf{a} if the L_1 norm is replaced with an L_0 norm.

- **ConvUnsup**: Convolutional encoding. Weights are initialized from weights learned via unsupervised sparse coding and are not updated. This control tests the effect of weights versus encoding scheme on performance.
- **ConvFinetune**: Convolutional encoding. Weights are initialized from weights learned via unsupervised sparse coding. The weights were additionally trained via supervised training for car detection. This control is similar to ConvUnsup but tests the effect of additional training on the first-layer weights.

Once the first layer is set to one of the five possible options, the remaining $n - 1$ layers contain convolutional layers learned via backpropagation of the supervised loss. Each model was repeated six times with different random initial conditions and random presentations of the training data to get a range in performance for each network.

All models, experiments, and hyperparameters are available online².

4 Results

Table 1 shows the area under precision-versus-recall curve (AUC) of all models trained on all available training data, each tested with two, three, and four layers. Each network was trained on the car-detection task six times, with random weight initialization in higher layers and random ordering of training examples, in order to obtain the range of AUC values ("scores"). The range columns in Table 1 gives the difference between the maximum and minimum scores over the six runs. Here, we find that **SparseUnsup** performs worse than **ConvSup** and **ConvFinetune** with two layers, which agrees with the findings of Coates et al. [2]. However, **SparseUnsup** outperforms **ConvSup** in networks with three or more layers. This difference in performance due to the number of layers is likely because of the lack in capacity of the two layer model to map input to detection.

One key finding is that **SparseUnsup** is much more consistent (i.e., much less susceptible to random initial conditions and ordering of training examples) compared to all other models, as shown by the low range of AUC scores. Interestingly, **SparseUnsup** has lower range in performance than **ConvUnsup**, where both models use unsupervised weights learned via sparse coding and only differ in encoding scheme. This suggests that inferring activations in sparse coding is likely the reason for the additional consistency in performance.

Figure 1 gives the performance of **SparseUnsup** and **ConvSup** while varying the number of labeled training examples that each network was trained on. Here, the unsupervised weights were learned from all available data, without using training labels. We find that **SparseUnsup** achieves better performance than **ConvSup** across all numbers of labeled training examples tested for three and four layer networks. Additionally, **SparseUnsup** achieves better performance with two layer networks when the number of training examples is limited to only 100 training examples. Finally, **SparseUnsup** is much more consistent (as shown by the low range in AUC scores) in performance than that of **ConvSup**, for all models tested.

We compare activation maps for the first layer of **SparseUnsup**, **ConvUnsup**, and **ConvSup** in Figure 2. We find that the sparse-coding activations are selective to certain depths. For example, in Figure 2 for **SparseUnsup**, the top row shows a fast moving edge detector with a large binocular shift that corresponds to image features close to the camera, whereas the bottom row shows a static edge detector with no binocular shift that corresponds to image features far from the camera. In contrast, no convolutional layers show depth selectivity.

To control for the difference in number of nonzero activations, we applied a threshold to convolutional activations to match the number of nonzero activations in sparse coding, as shown in Figure 2 **ConvSup Sparse Control**. Sparse controls for **ConvUnsup** produced no nonzero activations on this input. Here, we show that sparsity-controlled activations from **ConvSup** do not show depth selectivity, as activations are active on image features at different depths. In all, these results may explain the differences in performance between convolutional layers and sparse-coding layers.

²<https://github.com/slundqui/TFSparseCode/>

Model	2 layers	Range	3 layers	Range	4 layers	Range
ConvSup	0.672	0.045	0.672	0.021	0.681	0.086
SparseUnsup	0.619	0.004	0.681	0.009	0.693	0.014
ConvRand	0.467	0.009	0.574	0.052	0.592	0.033
ConvUnsup	0.561	0.044	0.609	0.028	0.609	0.033
ConvFinetune	0.660	0.020	0.641	0.081	0.691	0.117

Table 1: Area under precision-versus-recall curve (AUC) for all models tested with varying depths. Models were ran six times with different initial conditions. Each value represents the median AUC score, and the range represents the difference between the highest score and the lowest score. **Chance** scores at .221 AUC.

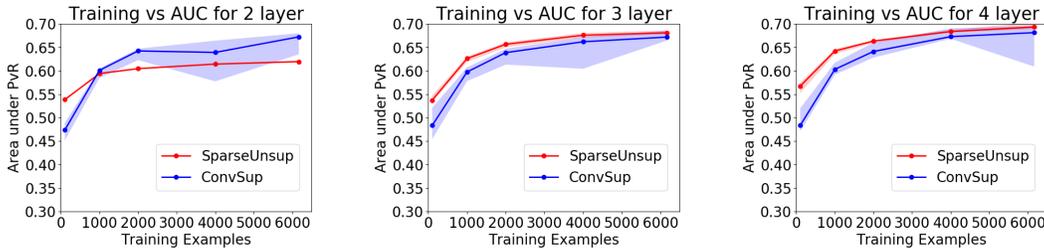


Figure 1: Number of training examples available versus AUC score for **SparseUnsup** (red) and **ConvSup** (blue) for two (left), three (middle), and four (right) layer networks. Each point is the median score over six independent runs, with the area between the maximum and minimum score filled in. Best viewed in color.

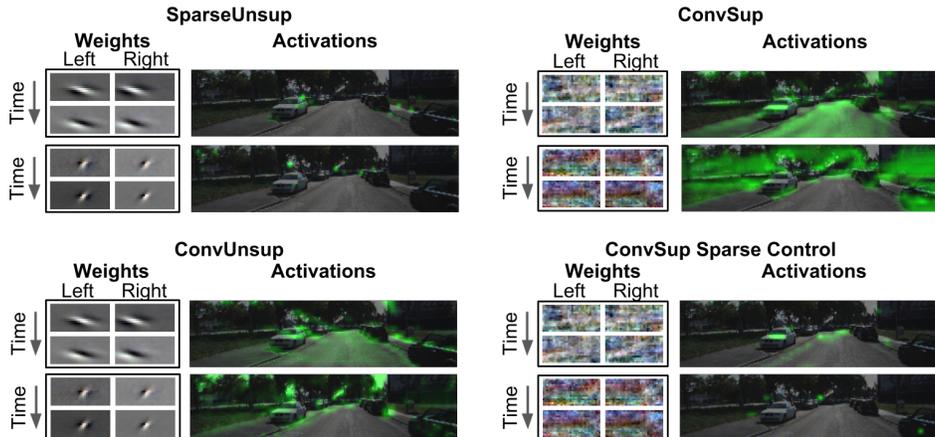


Figure 2: Nonzero activations of example weights overlaid on the input image. Magnitude of pixel values in green correspond to magnitude of activations. **SparseUnsup**: Activations for near tuned (top) and far tuned (bottom) weights for the sparse-coding layer. **ConvSup Sparse Control**: Activations from **ConvSup** with a threshold applied such that the number of activations matched that of sparse coding across the dataset. Best viewed in color.

5 Conclusion

We have shown that a neural network that incorporates unsupervised learning is able to outperform a fully supervised network when there exists limited labeled training data. Additionally, we show that performance of fully supervised networks can vary substantially when compared to networks with a sparse-coding layer. Finally, we compare activations and show that depth selective activations emerge from applying sparse coding to stereo-video data. In all, these results show that unsupervised sparse coding can be useful in domains where there exists a limited amount of available labeled training data.

Acknowledgments

This research is funded by the DARPA Cooperative Agreement Award HR0011-13-2-0015. We would like to thank Dylan M. Paiton for valuable conversations pertaining to this manuscript.

References

- [1] Horace B Barlow. Unsupervised learning. *Neural Computation*, 1(3):295–311, 1989.
- [2] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 215–223, 2011.
- [3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.
- [4] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.
- [5] Steven H Ferris. Motion parallax and absolute distance. *Journal of experimental psychology*, 95(2):258, 1972.
- [6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [8] Patrik O Hoyer and Aapo Hyvärinen. Independent component analysis applied to feature extraction from colour and stereo images. *Network: computation in neural systems*, 11(3):191–210, 2000.
- [9] Phil Knag, Jung Kuk Kim, Thomas Chen, and Zhengya Zhang. A sparse coding neural network asic with on-chip learning for feature extraction and encoding. *IEEE Journal of Solid-State Circuits*, 50(4):1070–1079, 2015.
- [10] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.
- [11] Sheng Y Lundquist, Dylan M Paiton, Peter F Schultz, and Garrett T Kenyon. Sparse encoding of binocular images for depth inference. In *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 121–124. IEEE, 2016.
- [12] Roland Memisevic and Christian Conrad. Stereopsis via deep learning. In *Advances in Neural Information Processing Systems (NIPS) Workshop on Deep Learning*, volume 1, page 2, 2011.
- [13] N.T.T. Nguyen and G.T. Kenyon. Solving sparse representations for object classification using quantum d-wave 2x machine. In *International workshop on Post-Moore’s Era Supercomputing (MPES)*, 2016.
- [14] Bruno A Olshausen. Sparse coding of time-varying natural images. In *International Conference on Independent Component Analysis and Blind Source Separation*, pages 603–608. Citeseer, 2000.
- [15] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.
- [16] Ning Qian. Binocular disparity and the perception of depth. *Neuron*, 18(3):359–368, 1997.

- [17] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *International conference on Machine learning (ICML)*, pages 759–766. ACM, 2007.
- [18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.
- [19] Christopher Rozell, Don Johnson, Richard Baraniuk, and Bruno Olshausen. Locally competitive algorithms for sparse approximation. In *IEEE International Conference on Image Processing (ICIP)*, volume 4, pages IV–169. IEEE, 2007.
- [20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.
- [21] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE, 2010.
- [22] Mengchen Zhu and Christopher J Rozell. Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system. *PLoS computational biology*, 9(8):e1003191, 2013.