
Augmenting Supervised Neural Networks with Unsupervised Objectives for Large-scale Image Classification

Yuting Zhang
Kibok Lee
Honglak Lee

Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

YUTINGZH@UMICH.EDU
KIBOK@UMICH.EDU
HONGLAK@EECS.UMICH.EDU

Abstract

Unsupervised learning and supervised learning are key research topics in deep learning. However, as high-capacity supervised neural networks trained with a large amount of labels have achieved remarkable success in many computer vision tasks, the availability of large-scale labeled images reduced the significance of unsupervised learning. Inspired by the recent trend toward revisiting the importance of unsupervised learning, we investigate joint supervised and unsupervised learning in a large-scale setting by augmenting existing neural networks with decoding pathways for reconstruction. First, we demonstrate that the intermediate activations of pretrained large-scale classification networks preserve almost all the information of input images except a portion of local spatial details. Then, by end-to-end training of the entire augmented architecture with the reconstructive objective, we show improvement of the network performance for supervised tasks. We evaluate several variants of autoencoders, including the recently proposed “what-where” autoencoder that uses the encoder pooling switches, to study the importance of the architecture design. Taking the 16-layer VGGNet trained under the ImageNet ILSVRC 2012 protocol as a strong baseline for image classification, our methods improve the validation-set accuracy by a noticeable margin.

1. Introduction

Unsupervised and supervised learning have been two associated key topics in deep learning. One important application of deep unsupervised learning over the past decade was to pretrain a deep neural network, which was then finetuned with supervised tasks (such as classification). Many deep unsupervised models were proposed, such as stacked (denoising) autoencoders (Bengio et al., 2007; Vin-

cent et al., 2010), deep belief networks (Hinton et al., 2006; Lee et al., 2009), sparse encoder-decoders (Ranzato et al., 2007; Kavukcuoglu et al., 2010), and deep Boltzmann machines (Salakhutdinov & Hinton, 2009). These approaches significantly improved the performance of neural networks on supervised tasks when the amount of available labels were not large.

However, over the past few years, supervised learning without any unsupervised pretraining has achieved even better performance, and it has become the dominating approach to train deep neural networks for real-world tasks, such as image classification (Krizhevsky et al., 2012) and object detection (Girshick et al., 2016). Purely supervised learning allowed more flexibility of network architectures, e.g., the inception unit (Szegedy et al., 2015) and the residual structure (He et al., 2016), which were not limited by the modeling assumptions of unsupervised methods. Furthermore, the recently developed batch normalization (BN) method (Ioffe & Szegedy, 2015) has made the neural network learning further easier. As a result, the once popular framework of unsupervised pretraining has become less significant and even overshadowed (LeCun et al., 2015) in the field.

Several attempts (e.g., Ranzato & Szummer (2008); Larochelle & Bengio (2008); Sohn et al. (2013); Goodfellow et al. (2013)) had been made to couple the unsupervised and supervised learning in the same phase, making unsupervised objectives able to impact the network training after supervised learning took place. These methods unleashed new potential of unsupervised learning, but they have not yet been shown to scale to large amounts of labeled and unlabeled data. Rasmus et al. (2015) recently proposed an architecture that is easy to couple with a classification network by extending the stacked denoising autoencoder with lateral connections, i.e., from encoder to the same stages of the decoder, and their methods showed promising semi-supervised learning results. Nonetheless, the existing validations (Rasmus et al., 2015; Pezeshki et al., 2016) were mostly on small-scale datasets like MNIST. Recently, Zhao et al. (2015) proposed the “what-

where” autoencoder (SWWAE) by extending the stacked convolutional autoencoder using Zeiler et al. (2011)’s “unpooling” operator, which recovers the locational details (which was lost due to max-pooling) using the pooling switches from the encoder. While achieving promising results on the CIFAR dataset with extended unlabeled data (Torralba et al., 2008), SWWAE has not been demonstrated effective for larger-scale supervised tasks.

In this paper, inspired by the recent trend toward simultaneous supervised and unsupervised neural network learning, we augment challenge-winning neural networks with decoding pathways for reconstruction, demonstrating the feasibility of improving high-capacity networks for large-scale image classification. Specifically, we take a segment of the classification network as the encoder and use the mirrored architecture as the decoding pathway to build several autoencoder variants. The autoencoder framework is easy to construct by augmenting an existing network without involving complicated components. Decoding pathways can be trained either separately from or together with the encoding/classification pathway by the standard stochastic gradient descent methods without special tricks, such as noise injection and activation normalization.

This paper first investigates reconstruction properties of the large-scale deep neural networks. Inspired by Dosovitskiy & Brox (2016), we use the auxiliary decoding pathway of the stacked autoencoder to reconstruct images from intermediate activations of the pretrained classification network. Using SWWAE, we demonstrate better image reconstruction qualities compared to the autoencoder using the unpooling operators with *fixed* switches, which upsamples an activation to a fixed location within the kernel. This result suggests that the intermediate (even high-level) feature representations preserve nearly all the information of the input images except for the locational details “neutralized” by max-pooling layers.

Based on the above observations, we further improve the quality of reconstruction, an indication of the mutual information between the input and the feature representations (Vincent et al., 2010), by finetuning the *entire* augmented architecture with supervised and unsupervised objectives. In this setting, the image reconstruction loss can also impact the classification pathway. To the contrary of conventional beliefs in the field, we demonstrate that the unsupervised learning objective posed by the auxiliary autoencoder is an effective way to help the classification network obtain better local optimal solutions for supervised tasks. To the best of our knowledge, this work is the first to show that unsupervised objective can improve the image classification accuracy of deep convolutional neural networks on large-scale datasets, such as ImageNet (Deng et al., 2009). We summarize our main contributions as follows:

- We show that the feature representations learned by high-capacity neural networks preserve the input information extremely well, despite the spatial invariance induced by pooling. Our models can perform high-quality image reconstruction (i.e., “inversion”) from intermediate activations with the unpooling operator using the known switches from the encoder.
- We successfully improve the large-scale image classification performance of a state-of-the-art classification network by finetuning the augmented network with a reconstructive decoding pathway to make its intermediate activations preserve the input information better.
- We study several variants of the resultant autoencoder architecture, including instances of SWWAE and more basic versions of autoencoders, and provide insight on the importance of the pooling switches and the layer-wise reconstruction loss.

2. Related work

In terms of using image reconstruction to improve classification, our work is related to supervised sparse coding and dictionary learning work, which is known to extract sparse local features from image patches by sparsity-constrained reconstruction loss functions. The extracted sparse features are then used for classification purposes. Mairal et al. (2009) proposed to combine the reconstruction loss of sparse coding and the classification loss of sparse features in a unified objective function. Yang et al. (2010) extended this supervised sparse coding with max-pooling to obtain translation-invariant local features.

Zeiler et al. (2010) proposed deconvolutional networks for unsupervised feature learning that consist of multiple layers of convolutional sparse coding with max-pooling. Each layer is trained to reconstruct the output of the previous layer. Zeiler et al. (2011) further introduced the “unpooling with switches” layer to deconvolutional networks to enable end-to-end training.

As an alternative to sparse coding and discriminative convolutional networks, autoencoders (Bengio, 2009) are another class of models for representation learning, in particular for the non-linear principal component analysis (Dong & McAvoy, 1996; Scholz & Vigário, 2002) by minimizing the reconstruction errors of a bottlenecked neural network. The stacked autoencoder (SAE) (Bengio et al., 2007) is amenable for hierarchical representation learning. With pooling-induced sparsity bottlenecks (Makhzani & Frey, 2015), the convolutional SAE (Masci et al., 2011) can learn features from middle-size images. In these unsupervised feature learning studies, sparsity is the key regularizer to induce meaningful features in a hierarchy.

By injecting noises or corruptions to the input, denoising autoencoders (Vincent et al., 2008; 2010) can learn robust filters to recover the uncorrupted input. Valpola (2015) further added noises to intermediate layers of denoising auto-encoders with lateral connections, which was called “ladder network”. Rasmus et al. (2015) combined a classification task with the ladder network for semi-supervised learning, and they showed improved classification accuracy on MNIST and CIFAR-10. Here, supervision from the labeled data is the critical objective that prevents the autoencoder from learning trivial features.

Zhao et al. (2015) proposed the SWWAE, a convolutional autoencoder with unpooling layer, and combined it with classification objective for semi-supervised learning. This model integrates a discriminative convolutional network (for classification) and a deconvolutional network (for reconstruction) and can be regarded as a unification of deconvolutional networks, autoencoders and discriminative convolutional networks. They demonstrated promising results on small scale datasets such as MNIST, SVHN and STL10.

Improving representation learning with auxiliary tasks is not new (Sudderth & Kergosien, 1990). The idea behind is that the harder the tasks are, the better representations a network can learn. As an alternative to the autoencoder, Lee et al. (2015)’s “deeply supervised network” incorporated classification objectives for intermediate layers, was able to improve the top-layer classification accuracy for reasonably large-scale networks (Wang et al., 2015). In earlier work, Ranzato & Szummer (2008) conducted layer-wise training by both classification and reconstruction objectives. Recently, more task-specific unsupervised objectives for image and video representation learning were developed by using spatial context (Doersch et al., 2015) and video continuity (Wang & Gupta, 2015). In contrast, autoencoder-based methods are applicable in more general scenarios.

3. Methods

In this section, we describe the training objectives and architectures of the proposed augmented network. In Section 3.1, we briefly review the architectures of recent networks for vision tasks, and present the general form of our method. In Section 3.2, we augment the classification network with auxiliary pathways composed of deconvolutional architectures to build fully mirrored autoencoders, on which we specify the auxiliary objective functions.

3.1. Unsupervised loss for intermediate representations

Deep neural networks trained with full supervision achieved the state-of-the-art image classification performance. Commonly used network architectures

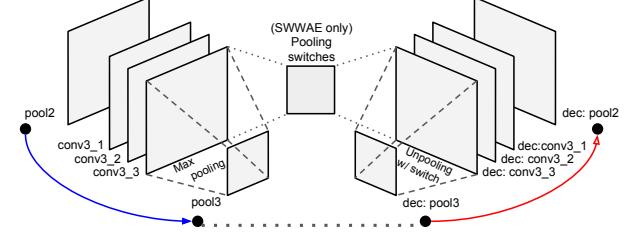


Figure 1. Example micro-architectures in macro-layers (the 3rd macro-layer of VGGNet and its mirrored decoder). *Encoder*: a number of convolutional layers followed by a max-pooling layer. *Decoder*: the same number of deconvolutional layers preceded by an unpooling layer, where the known pooling switches given by the associated pooling layer are used for SWVAE.

(Krizhevsky et al., 2012) contain a single pathway of convolutional layers succeeded by nonlinear activation functions and interleaved with max-pooling layers to gradually transform features into high-level representations and gain spatial invariance at different scales. Recent networks (Simonyan & Zisserman, 2015; Szegedy et al., 2015; He et al., 2016; Szegedy et al., 2016) often nest a group of convolutional layers before applying a max-pooling layer. As these layers work together as the feature extractor for a particular scale, we refer to the group as a *macro-layer* (see the left half of Figure 1). Fully-connected inner-product layer and/or global average-pooling layer follow the convolution-pooling macro-layers to feed the top-layer classifier. A network of L convolution-pooling macro-layers is defined as

$$a_l = f_l(a_{l-1}; \phi_l), \text{ for } l = 1, 2, \dots, L+1, \quad (1)$$

where $a_0 = x$ is the input, $f_l(l = 1, 2, \dots, L)$ with the parameter ϕ_l is the l^{th} macro-layer, and f_{L+1} denotes the rest of the network, including the inner-product and classification layers. The classification loss is $C(x, y) = \ell(a_{L+1}, y)$, where y is the ground truth label, and ℓ is the cross-entropy loss when using a softmax classifier.

Let x_1, x_2, \dots, x_N denote a set of training images associated with categorical labels y_1, y_2, \dots, y_N . The neural network is trained by minimizing $\frac{1}{N} \sum_{i=1}^N C(x_i, y_i)$, where we omit the L2-regularization term on the parameters. Though this objective can effectively learn a large-scale network by gradient descent with a huge amount of labeled data, it has two limitations. On the one hand, the training of lower intermediate layers might be problematic, because the gradient signals from the top layer can become vanished (Hochreiter et al., 2001) on its way to the bottom layer. Regularization by normalization (Ioffe & Szegedy, 2015) can alleviate this problem, but will also lead to large yet noisy gradients when networks are deep (He et al., 2016). On the other hand, the data space is infor-

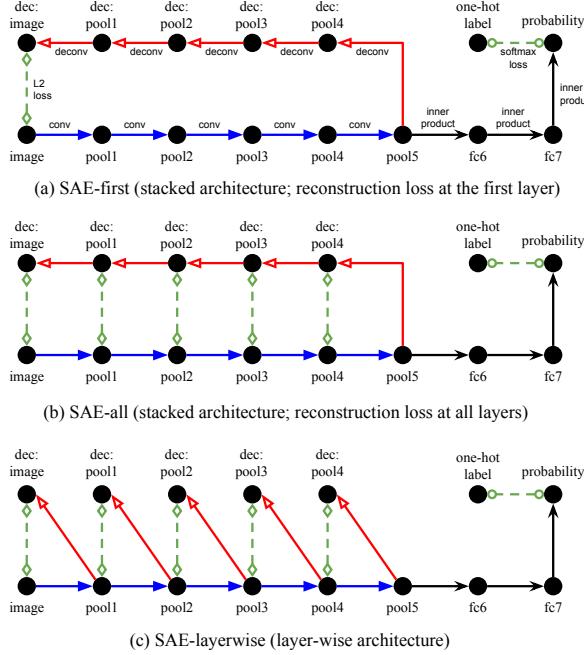


Figure 2. Model architectures of networks augmented with autoencoders. ● : nodes; → : encoder macro-layer; ← : decoder macro-layer; → : inner-product layer; ▲ ▼ : reconstruction loss; ○ □ : classification loss.

mative by itself, but the fully supervised objective guides the representation learning purely by the labels.

A solution to both problems is to incorporate auxiliary unsupervised training objectives to the intermediate layers. More specifically, the objective function becomes

$$\frac{1}{N} \sum_{i=1}^N (C(x_i, y_i) + \lambda U(x_i)), \quad (2)$$

where $U(\cdot)$ is the unsupervised objective function associating with one or more auxiliary pathways that are attached to the convolution-pooling macro-layers in the original classification network.

3.2. Network augmentation with autoencoders

Given the network architecture for classification defined in Eq. (1), we take the sub-network composed of all the convolution-pooling macro-layers as the encoding pathway, and generate a fully mirrored decoder network as an auxiliary pathway of the original network. The inner-product layers close to the top-level classifier may be excluded from the autoencoder, since they are supposed to be more task-relevant.

Taking a network of five macro-layers as an example (e.g., VGGNet), Figure 2a shows the network augmented with a stacked autoencoder. The decoding starts from the pooled

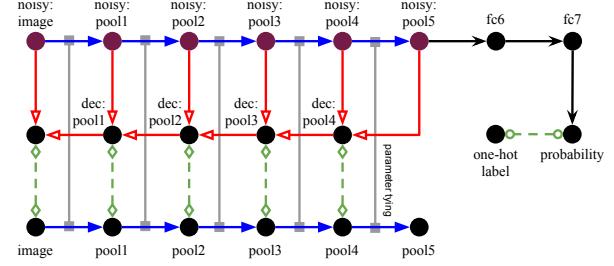


Figure 3. Ladder network architectures Rasmus et al. (2015). ● : nodes; ● : noisy nodes; → : encoder macro-layer; ← : decoder macro-layer; → : inner-product layer; ▲ ▼ : reconstruction loss; ○ □ : classification loss; ─ ─ : parameter tying.

feature map from the 5th macro-layer (pool5) all the way down to the image input. Reconstruction errors are measured at the network input (i.e., the first layer) so that we term the model as “SAE-first”. More specifically, the decoding pathway is

$$\hat{a}_L = a_L, \hat{a}_{l-1} = f_l^{dec}(\hat{a}_l; \psi_l), \hat{x} = \hat{a}_0. \quad (3)$$

with the loss $U_{\text{SAE-first}}(x) = \|\hat{x} - x\|_2^2$. Here, ψ_l 's are decoder parameters.

The auxiliary training signals of SAE-first emerge from the bottom of the decoding pathway, and they get merged with the top-down signals for classification at the last convolution-pooling macro-layer into the encoder pathway. To allow more gradient to flow directly into the preceding macro-layers, we propose the “SAE-all” model by replacing the unsupervised loss by $U_{\text{SAE-all}}(x) = \sum_{l=0}^{L-1} \gamma_l \|\hat{a}_l - a_l\|_2^2$, which makes the autoencoder have an even better mirrored architecture by matching activations for all the macro-layer (illustrated in Figure 2b).

In Figure 2c, we propose one more autoencoder variant with layer-wise decoding architecture, termed “SAE-layerwise”. It reconstructs the output activations of every macro-layer to its input. The auxiliary loss of SAE-layerwise is the same as SAE-all, i.e., $U_{\text{SAE-layerwise}}(x) = U_{\text{SAE-all}}(x)$, but the decoding pathway is replaced by $\hat{a}_{l-1} = f_l^{dec}(a_l; \psi_l)$.

SAE-first/all encourages top-level convolution features to preserve as much information as possible. In contrast, the auxiliary pathways in SAE-layerwise focus on inverting the clean intermediate activations (from the encoder) to the input of the associated macro-layer, admitting parallel layer-wise training. We investigated both in Section 4.3 and take SAE-layerwise decoders as architectures for efficient pre-training.

In Figure 1, we illustrate the detailed architecture of $f_3(\cdot)$ and $f_3^{dec}(\cdot)$ for Simonyan & Zisserman (2015)'s 16-layer VGGNet. Inspired by Zeiler et al. (2011), we use Zhao

et al. (2015)’s SWVAE as the default for the micro-architecture. More specifically, we record the pooling switches (i.e., the locations of the local maxima) in the encoder, and unpool activations by putting the elements at the recorded locations and filling the blanks with zeros. Unpooling with known switches can recover the local spatial variance eliminated by the max-pooling layer, avoiding the auxiliary objectives from deteriorating the spatial invariance of the encoder filters, which is arguably important for classification. We studied the autoencoders with fixed and known unpooling switch, respectively. In Section 4.2 we efficiently trained the autoencoders augmented from a pre-trained deep non-BN network, where the decoder is hard to learn from scratch.

Rasmus et al. (2015)’s ladder network (Figure 3) is a more sophisticated way to augment existing sequential architectures with autoencoders. It is featured by the lateral connections (vertical in Figure 3) and the combinator functions that merge the lateral and top-down activations. Due to the lateral connections, noise must be added to the encoder; otherwise, the combinator function can trivially copy the clean activations from the encoder. In contrast, no autoencoder variant used in our work has “lateral” connections, which makes the overall architectures of our models simpler and more standard. In SWVAE, the pooling switch connections do not bring the encoder input directly to the decoder, so they cannot be taken as the lateral connections like in the “ladder network”. Moreover, noise injection is also unnecessary for our models. We leave it as an open question whether denoising objectives can help with the augmented (what-where) autoencoder for large-scale data.

4. Experiments

In this section, we evaluated different variants of the augmented network for image reconstruction and classification on ImageNet ILSVRC 2012 dataset, using the training set for training, and validation set for evaluation. Our experiments were mainly based on the 16-layer VGGNet (Simonyan & Zisserman, 2015).¹ To compare with existing methods on inverting neural networks (Dosovitskiy & Brox, 2016), we also partially used Krizhevsky et al. (2012)’s network, termed AlexNet, trained on ILSVRC2012 training set. Our code and trained models can be obtained at <http://www.ytzheng.net/software/recon-dec/>

4.1. Training procedure

Training a deep neural network is non-trivial. Therefore, we propose the following strategy to make the networks

¹The pretrained network was obtained from http://www.robots.ox.ac.uk/~vgg/research/very_deep/.

augmented from the classification network efficiently trainable.

1. We initialized the encoding pathway with the pre-trained classification network, and the decoding pathways with Gaussian random initialization.
2. For any variant of the augmented network, we fixed the parameters for the classification pathway and trained the layer-wise decoding pathways of the SAE-layerwise network.
3. For SAE-first/all, we initialized the decoding pathway with the pretrained SAE-layerwise parameters and finetuned the decoder. (Skip this step for SAE-layerwise.)
4. We finetuned all the decoding and the encoding/classification pathways together with a reduced learning rate.

Up to Step 3, we trained the decoding pathways with the classification pathway fixed. For all the four steps, we trained the networks by mini-batch stochastic gradient descent (SGD) with the momentum 0.9.

In Step 2, the SAE-layerwise model has separate sub-pathways for decoding, so the training can be done in parallel for every macro-layer. The decoding sub-network for each macro-layer was relatively “shallow” so that it is easy to learn. We found the learning rate annealing not critical for SAE-layerwise pretraining. Proper base learning rates could make it sufficiently converged within 1 epoch. The chosen layer-wise learning rates VGGNet were summarized in Appendix A1 (Table A-1). We used a small mini-batch size of 16 for SGD.

For very deep networks, training the decoding pathways of SAE-first/all from random initialization is difficult when batch normalization is absent (e.g., in the VGGNet). Initializing with SAE-layerwise as in Step 3 is critical to efficiently train the stacked decoding pathways of SAE-first and SAE-all.

For SAE-all (Step 3, 4) and SAE-layerwise (Step 4), we balanced the reconstruction loss among different macro-layer, where the criterion was to make the weighted loss for every layer comparable to each other. We summarized the balancing weights for VGGNet in Appendix A1 (Table A-1). The SGD mini-batch size was set to a larger value (here, 64) in Step 4 for better stability.

We adopted commonly used data augmentation schemes. As to VGGNet, we randomly resized the image to [256, 512] pixels with respect to the shorter edge, and then randomly cropped a 224×224 patch (or its horizontally mirrored image) to feed into the network. As to AlexNet,

Layer	image	pool1	pool2	conv3	conv4	pool5	fc6	fc7	fc8
Dosovitskiy & Brox (2016)									
SWWAE-first (known unpooling switches)									

Figure 4. AlexNet reconstruction on ImageNet ILSVRC2012 validation set. See Appendix A2.5 (Figure A-4) for more results.

Layer	image	pool1	pool2	pool3	pool4	pool5
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						

Figure 5. VGGNet reconstruction on ImageNet ILSVRC2012 validation set. See Appendix A2.5 (Figure A-4) for more results.

we followed Krizhevsky et al. (2012)’s data augmentation scheme, cropping an image at the center to make it square with the shorter edge unchanged, resizing the square to 256×256 , and randomly sampling a 227×227 patch or its horizontally mirrored counterpart to feed the network. We ignored the RGB color jittering so as to always take ground truth natural images as the reconstruction targets.

Our implementation was based on the Caffe framework (Jia et al., 2014).

4.2. Image reconstruction via decoding pathways

Using reconstructive decoding pathways, we can visualize the learned hierarchical features by inverting a given classification network, which is a useful way to understand the learned representations. The idea of reconstructing the encoder input from its intermediate activations was first explored by Dosovitskiy & Brox (2016), in contrast to visualizing a single hidden node (Zeiler & Fergus, 2014) and dreaming out images (Mahendran & Vedaldi, 2015). As the best existing method for inverting neural networks with no skip link, it used unpooling with fixed switches to upsample the intermediate activation maps. This method demonstrated how much information the features produced by each layer could preserve for the input. As shown in Figure 4 (the top row), not surprisingly, the details of the input image gradually diminished as the representations went through higher layers.

The commonly used classification network mainly consists of convolution/inner-product and max-pooling operators. Based only on Dosovitskiy & Brox (2016)’s visualization, it is hard to tell how much the two types of operators contribute to the diminishing of image details, respectively. Note that our SAE-first architecture is comparable to Dosovitskiy & Brox (2016)’s model except for the better mirrored architectures between the encoder and decoder, which allow extending to SWWAE. Using the SWWAE-first network (“what-where” version of SAE-first), we were able to revert the max-pooling more faithfully, and to study the amount of information that the convolutional filters and inner-product coefficients preserved.

To compare with Dosovitskiy & Brox (2016), we augmented AlexNet to the corresponding SWWAE-first architecture.² Unlike in Section 3, we built SWWAE-first network starting from every layer, i.e., decoding pathway could start from conv1 to fc8. Each macro-layer in AlexNet included exactly one convolutional or inner-product layer. We trained the decoding pathway with the encoding/classification pathway fixed.

As shown in Figure 4, the images reconstructed from any

²The decoding pathway almost fully mirrored the classification network except the first layer (conv1). This convolutional layer used the stride 4 rather than 1, which approximates two additional 2×2 pooling layers. Therefore, we used three deconvolutional layers to inverse the conv1 layer.

layer, even including the top 1000-way classification layer, were almost visually perfect.³ Only the local contrast and color saturation became slightly different from the original images as the layer went higher. The surprisingly good reconstruction quality suggests that the features produced by AlexNet preserved nearly all the information of the input except for the spatial invariance gained by the max-pooling layers.

As commonly believed, learning task-relevant features for classification and preserving information were conflicting to some extent, since the “nuisance” should be removed for supervised tasks. According to our experiments, the locational details in different scales were almost the only information significantly neutralized by the deep neural network. For the convolutional and inner-product layers, it seems important to encode the input into a better (e.g., task-relevant) form without information loss.

We conducted similar experiments based on the 16-layer VGGNet. As no results using the unpooling with fixed switches had been reported yet, we trained the decoding pathways for both SAE-first (with fixed unpooling switches) and SWWAE-first (with known unpooling switches). We described the detailed training strategy in Section 4.3. In Figure 5, we showed the reconstruction examples up to the 5th macro-layer (the 13th layer). Images reconstructed by SAE-first were blurry for higher layers. In contrast, SWWAE-first could well recover the shape details from the pool5 features. In addition, the SWWAE-first model could also reasonably reconstruct non-ImageNet and even non-natural images like text screenshots, depth maps, and cartoon pictures, as shown in Appendix A2.5 (Figure A-3). These results suggest that the high-level feature representations were also adaptable to other domains.

Since the architecture was much deeper than AlexNet, VGGNet resulted in noisier reconstruction. Assuming the ability of preserving information as a helpful property for deep neural network, we took the reconstruction loss as an auxiliary objective function for training the classification network, as will be described in Section 4.3.

4.3. Image classification with augmented architectures

We took as the baseline the 16-layer VGGNet (Simonyan & Zisserman (2015)’s Model D), one of the best open source convolutional neural networks for large-scale image classification.

We needed only to use the classification pathway for testing. We report results with the following two schemes for sampling patches to show both more ablative and more

³For the fc6 and fc7 layers, we applied inner-product followed by relu nonlinearity; for the fc8 layer, we applied only inner-product, but not softmax nonlinearity.

practical performance on single networks.

Single-crop We resized the test image, making its shorter edge 256 pixels, and used only the single 224 × 224 patch (without mirroring) at the center to compute the classification score. It allowed us to examine the tradeoff between training and validation performance without complicated post-processing.

Convolution We took the VGGNet as a fully convolutional network and used a global average-pooling to fuse the classification scores obtained at different locations in the grid. The test image was resized to 256 pixels for the shorter edge and mirrored to go through the convolution twice. It was a replication of Section 3.2 of (Simonyan & Zisserman, 2015).

We report the experimental results in Table 1. Several VGGNet (classification pathway only) results are presented to justify the validity of our baseline implementation. As a replication of Simonyan & Zisserman (2015)’s “single-scale” method, our second post-processing scheme could achieve similar comparable accuracy. Moreover, finetuning the pretrained VGGNet model further without the augmented decoding network using the same training procedure did not lead to significant performance change.

As a general trend, all of the networks augmented with autoencoders outperformed the baseline VGGNet by a noticeable margin. In particular, compared to the VGGNet baseline, the SWWAE-all model reduced the top-1 errors by 1.66% and 1.18% for the single-crop and convolution schemes, respectively. It also reduced the top-5 errors by 1.01% and 0.81%, which are 10% and 9% relative to the baseline errors.

To the best of our knowledge, this work provides the first experimental results to demonstrate the effectiveness of unsupervised learning objectives for improving the state-of-the-art image classification performance on large-scale realistic datasets. For SWWAE-all, the validation accuracy in Table 1 was achieved in ∼16 epochs, which took 4–5 days on a workstation with 4 Nvidia Titan X GPUs. Taking pretrained VGGNet as the reference, 75% of the relative accuracy improvement (∼1.25% absolute top-1 accuracy improvement) could be achieved in ∼4 epochs (∼1 day).

Apart from the general performance gain due to reconstructive decoding pathways, the architecture changes could result in relatively small differences. Compared to SWWAE-layerwise, SWWAE-all led to slightly higher accuracy, suggesting the usefulness of posing a higher requirement on the top convolutional features for preserving the input information. The slight performance gain of SWWAE-all over SAE-all with fixed unpooling switches indicates that the switch connections could alleviate the difficulty

Sampling	Single-crop (center patch, no mirroring)				Convolution	
	Errors		Top-1		Top-5	
	Model	Train	Val.	Train	Val.	Validation
VGGNet [†]	–	–	–	–	27.0*	8.8*
VGGNet [†]	–	–	–	–	26.8**	8.7**
VGGNet	17.43	29.05	4.02	10.07	26.97	8.94
SAE-first	15.36	27.70	3.13	9.28	26.09	8.30
SAE-all	15.64	27.54	3.23	9.17	26.10	8.21
SAE-layerwise	16.20	27.60	3.42	9.19	26.06	8.17
SWWAE-first	15.10	27.60	3.08	9.23	25.87	8.14
SWWAE-all	15.67	27.39	3.24	9.06	25.79	8.13
SWWAE-layerwise	15.42	27.53	3.32	9.10	25.97	8.20

[†] The numbers in the last rows are from Table 3 (Model D) in Simonyan & Zisserman (2015) (the most comparable to our settings).⁴

* from a slightly different model trained with single-scale (256px) data augmentation. ** Test scale is 384px.

Table 1. Classification errors on ImageNet ILSVRC-2012 validation dataset based on 16-layer VGGNet. SAE models use the unpooling with fixed switches, and SWWAE models uses the unpooling with known switches.

of learning a stacked convolutional autoencoder. In the meanwhile, it also suggests that, without pooling switches, the decoding pathway can benefit the classification network learning similarly. Using the unpooling with fixed switches, the decoding pathway may not be limited for reconstruction, but can also be designed for the structured outputs that are not locationally aligned with the input images (e.g., adjacent frames in videos, another viewpoint of the input object).

To figure out whether the performance gain was due to the potential regularization effects of the decoding pathway or not, we evaluated the networks on 50,000 images randomly chosen from the training set. Interestingly, the networks augmented with autoencoders achieved lower training errors than the baseline VGGNet. Hence, rather than regularizing, it is more likely that the auxiliary unsupervised loss helped the CNN to find better local optima in supervised learning. Compared to SAE/SWWAE-all, SAE/SWWAE-first led to lower training errors but higher validation errors, a typical symptom of slight overfitting. Thus, incorporating layer-wise reconstruction loss was an effective way to regularize the network training.

We provide more discussion for the decoding pathways in Appendix A2, including image reconstruction results after finetuning the augmented networks (Appendix A2.5), training curves (Appendix A2.2), and comparison between the pretrained and finetuned convolution filters (Appendix A2.1).

⁴In our experiments, the 16-layer VGGNet (Simonyan & Zisserman (2015)'s Model D) achieved 10.07% for the single-crop scheme and 8.94% for the convolution scheme (in a single scale), which is comparable to 8.8% in Table 3 of (Simonyan & Zisserman, 2015). In that table, the best reported number for the Model D was 8.1%, but it is trained and tested using a different resizing and cropping method, thus not comparable to our results.

5. Conclusion

We proposed a simple and effective way to incorporate unsupervised objectives into large-scale classification network learning by augmenting the existing network with reconstructive decoding pathways. Using the resultant autoencoder for image reconstruction, we demonstrated the ability of preserving input information by intermediate representation as an important property of modern deep neural networks trained for large-scale image classification. We leveraged this property further by training the augmented network composed of both the classification and decoding pathways. This method improved the performance of the 16-layer VGGNet, one of the best existing networks for image classification by a noticeable margin. We investigated different variants of the autoencoder, and showed that 1) the pooling switch connections between the encoding and decoding pathways were helpful, but not critical for improving the performance of the classification network in large-scale settings; 2) the decoding pathways mainly helped the supervised objective reach a better optimum; and 3) the layer-wise reconstruction loss could effectively regularize the solution to the joint objective. We hope this paper will inspire further investigations on the use of unsupervised learning in a large-scale setting.

Acknowledgements

This work was funded by Software R&D Center, Samsung Electronics Co., Ltd; ONR N00014-13-1-0762; and NSF CAREER IIS-1453651. We also thank NVIDIA for donating K40c and TITAN X GPUs. We thank Jimei Yang, Seunghoon Hong, Ruben Villegas, Wenling Shang, Kihyuk Sohn, and other collaborators for helpful discussions.

References

- Bengio, Y. Learning deep architectures for ai. *Foundation and Trends in Machine Learning*, 2(1):1–127, January 2009.
- Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In *NIPS*, 2007.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.
- Dong, D. and McAvoy, T. J. Nonlinear principal component analysis based on principal curves and neural networks. *Computers & Chemical Engineering*, 20(1):65–78, 1996.
- Dosovitskiy, A. and Brox, T. Inverting visual representations with convolutional networks. In *CVPR*, 2016.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, Jan 2016.
- Goodfellow, I., Mirza, M., Courville, A., and Bengio, Y. Multi-prediction deep boltzmann machines. In *NIPS*, 2013.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.
- Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
- Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Networks*. 2001.
- Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.
- Kavukcuoglu, K., Ranzato, M. A., and LeCun, Y. Fast inference in sparse coding algorithms with applications to object recognition. *arXiv:1010.3467*, 2010.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- Larochelle, H. and Bengio, Y. Classification using discriminative restricted boltzmann machines. In *ICML*, 2008.
- LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.
- Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. Deeply-supervised nets. In *AISTATS*, 2015.
- Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009.
- Mahendran, A. and Vedaldi, A. Understanding deep image representations by inverting them. In *CVPR*, 2015.
- Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., and Bach, F. R. Supervised dictionary learning. In *NIPS*, 2009.
- Makhzani, A. and Frey, B. J. Winner-take-all autoencoders. In *NIPS*, 2015.
- Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, 2011.
- Pezeshki, M., Fan, L., Brakel, P., Courville, A., and Bengio, Y. Deconstructing the ladder network architecture. *arXiv:1506.02351*, 2016.
- Ranzato, M. A. and Szummer, M. Semi-supervised learning of compact document representations with deep networks. In *ICML*, 2008.
- Ranzato, M. A., Huang, F. J., Boureau, Y.-L., and LeCun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.
- Rasmus, A., Valpola, H., Honkala, M., Berglund, M., and Raiko, T. Semi-supervised learning with ladder network. In *NIPS*, 2015.
- Salakhutdinov, R. and Hinton, G. E. Deep boltzmann machines. In *AISTATS*, 2009.
- Scholz, M. and Vigário, R. Nonlinear pca: a new hierarchical approach. In *ESANN*, 2002.
- Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- Sohn, K., Zhou, G., Lee, C., and Lee, H. Learning and selecting features jointly with point-wise gated Boltzmann machines. In *ICML*, 2013.

- Sudderth, S. and Kergosien, Y. Rule-injection hints as a means of improving network performance and learning time. *Neural Networks*, 412:120–129, 1990.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *CVPR*, 2015.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. 2016.
- Torralba, A., Fergus, R., and Freeman, W. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, Nov 2008.
- Valpola, H. From neural PCA to deep unsupervised learning. In *Advances in Independent Component Analysis and Learning Machines (Chapter 8)*, pp. 143 – 171. 2015.
- Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, December 2010.
- Wang, L., Lee, C.-Y., Tu, Z., and Lazebnik, S. Training deeper convolutional networks with deep supervision. *arXiv:1505.02496*, 2015.
- Wang, X. and Gupta, A. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.
- Yang, J., Yu, K., and Huang, T. Supervised translation-invariant sparse coding. In *CVPR*, 2010.
- Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *ECCV*, 2014.
- Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R. Deconvolutional networks. *CVPR*, 2010.
- Zeiler, M., Taylor, G., and Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011.
- Zhao, J., Mathieu, M., Goroshin, R., and Lecun, Y. Stacked what-where auto-encoders. *arXiv:1506.02351*, 2015.

Appendices

A1. Parameters for VGGNet-based models

Macro-layer	Learning rate	Loss weighting ¹
	SAE-layerwise	SAE-layerwise/all
1	3×10^{-9}	1×10^{-4}
2	1×10^{-8}	1×10^{-12}
3	3×10^{-12}	1×10^{-12}
4	1×10^{-12}	1×10^{-12}
5	1×10^{-11}	1×10^{-10}

LR: learning rate; ¹ the top-level softmax is weighted by 1.

Table A-1. Layer-wise training parameters for networks augmented from VGGNet

We report the learning parameters for 16-layer VGGNet-based model in Table A-1. We chose the learning rates that lead to the largest decrease in the reconstruction loss in the first 2000 iterations for each layer. The “loss weighting” are balancing factors for reconstruction losses in different layers varied to make them comparable in magnitude. In particular, we computed image reconstruction loss against RGB values normalized to [0,1], which are different in scale from intermediate features. We also did not normalize the reconstruction loss with feature dimensions for any layer.

A2. More experimental results and discussions

A2.1. Learned filters

Compared to the baseline VGGNet, the finetuned SWWAE-all model demonstrated $\sim 35\%$ element-wise relative change of the filter weights on average for all the layers. A small portion of the filters showed stronger contrast after finetuning. Qualitatively, the finetuned filters kept the pretrained visual shapes. In Figure A-1, we visualize the first-layer 3×3 convolution filters.

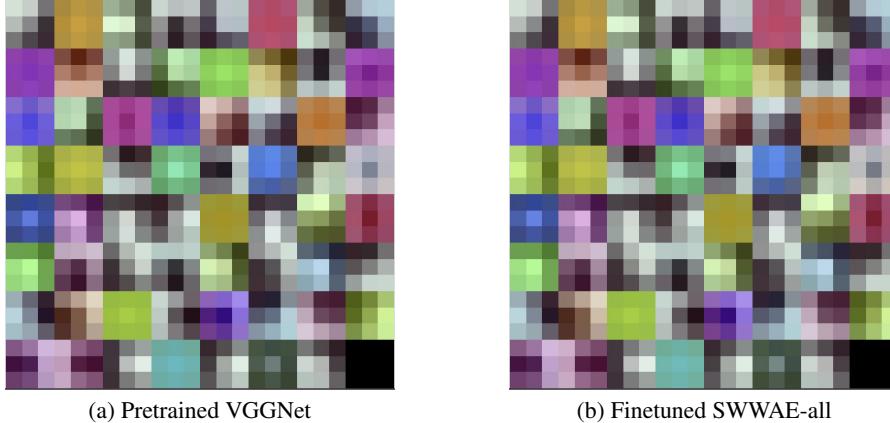


Figure A-1. Visualization of the normalized first-layer convolution filters in 16-layer VGGNet-based network. The filters of the SWWAE-all model had nearly the same patterns to those of the pretrained VGGNet, but showed stronger contrast. It is more clear see the difference if displaying the two images alternatively in the same place. (online example: <http://www.ytzheng.net/files/publications/2016-icml-recon-dec/filters/>)

A2.2. Training curve

In Figure A-2, we report the training curves of validation accuracy for SWWAE-all, where the pretrained VGGNet classification network and decoder network were taken as the starting point.

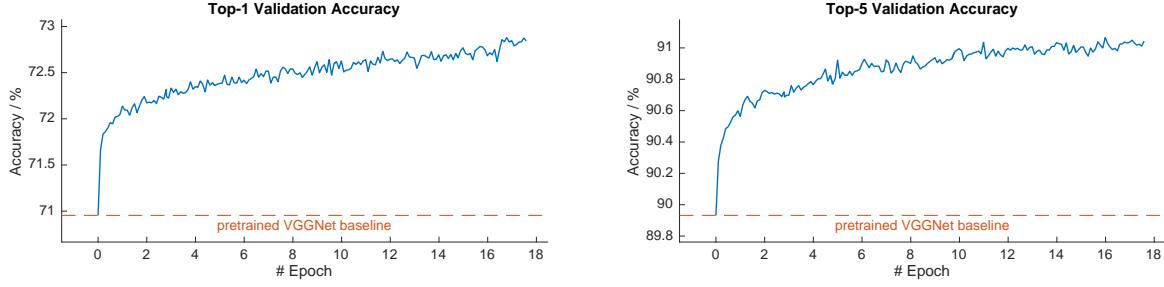


Figure A-2. Training curves for the single-crop validation accuracy of VGGNet-based SWWAE-all models.

A2.3. Selection of different model variants

The performance for different variants of the augmented network are comparable, but we can still choose the best available one. In particular, we provide following discussions.

- Since the computational costs were similar for training and the same for testing, we can use the best available architecture depending on tasks. For example, when using decoding pathways for spatially corresponded tasks like reconstruction (as in our paper) and segmentation, we can use the SWWAE. For more general objectives like predicting next frames, where pooling switches are non-transferrable, we can still use ordinary SAEs to get competitive performance.
- S(WW)AE-first has less hyper-parameters than S(WW)AE-all, and can be trained first for quick parameter search. It can be switched to *-all for better performance.

A2.4. Ladder networks

We tried training a ladder network following the same procedures of pretraining auxiliary pathways and finetuning the whole network as for our models, which is also similar to Rasmus et al. (2015)'s strategy. We used the augmented multi-layer perceptron (AMLP) combinator, which Pezeshki et al. (2016) proposed as the best combinator function. Different from the previous work conducted on the variants of MNIST dataset, the pretrained VGGNet does not have batch normalization (BN) layers, which pushed us to remove the BN layers from the ladder network. However, BN turned out to be critical for proper noise injection, and the non-BN ladder network did not perform well. It might suggest that our models are easier to pair with a standard convolutional network and train on large-scale datasets.

A2.5. Image reconstruction

In Figure A-3, we visualize the images reconstructed by the pretrained decoder of SWWAE-first and the final models for SWWAE-first/all, and reported the L2 reconstruction loss on the validation set. Finetuning the entire networks also resulted in better reconstruction quality, which is consistent with our assumption that enhancing the ability of preserving input information can lead to better features for image classification. Since the shape details had already been well recovered by the pretrained decoder, the finetuned SWWAE-first/all mainly improved the accuracy of colors. Note that the decoder learning is more difficult for SWWAE-all than SWWAE-first, which explains its slightly higher reconstruction loss and better regularization ability.

In Figure A-4 and A-5, we showed more examples for reconstructing input images from pretrained neural network features for AlexNet and VGGNet.

Model	L2 Loss	ImageNet				Non-ImageNet ¹			
Ground truth	-								
SWWAE-first (Pretrained, fixing encoder)	513.4								
SWWAE-first (Finetuned with encoder)	462.2								
SWWAE-all (Finetuned with encoder)	493.0								

¹ The first three images are from morguefile.com; the fourth is a screenshot of Wikipedia; the fifth is a depth image from NYU dataset; the last is used with permission from Debbie Ridpath Ohi at Inkygirl.com

Figure A-3. Image reconstruction from pool5 features to images. The reconstruction loss is computed on the ILSVRC2012 validation set and measured with L2-distance with the ground truth (RGB values are in [0, 1]). The first 2 example images are from the ILSVRC2012 validation set (excluding the 100 categories). The rest are not in ImageNet.

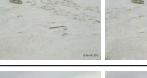
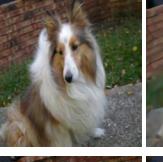
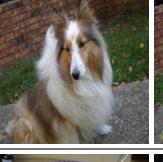
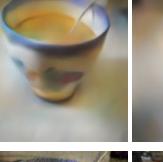
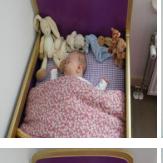
Layer	image	pool1	pool2	conv3	conv4	pool5	fc6	fc7	fc8
Dosovitskiy & Brox (2016) (fixed unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									
SWVAE-first (known unpooling switches)									

Figure A-4. AlexNet reconstruction on ImageNet ILSVRC2012 validation set. (Best viewed when zoomed in on a screen.)

Layer	image	pool1	pool2	pool3	pool4	pool5
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						

Layer	image	pool1	pool2	pool3	pool4	pool5
SAE-first (fixed unpooling switches)						
						
SWWAE-first (known unpooling switches)						
						
SAE-first (fixed unpooling switches)						
						
SWWAE-first (known unpooling switches)						
						
SAE-first (fixed unpooling switches)						
						

Layer	image	pool1	pool2	pool3	pool4	pool5
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						
SAE-first (fixed unpooling switches)						
SWWAE-first (known unpooling switches)						

Figure A-5. VGGNet reconstruction on ImageNet ILSVRC2012 validation set. (Best viewed when zoomed in on a screen.)