10

Selected Applications in Object Recognition and Computer Vision

Over the past two years or so, tremendous progress has been made in applying deep learning techniques to computer vision, especially in the field of object recognition. The success of deep learning in this area is now commonly accepted by the computer vision community. It is the second area in which the application of deep learning techniques is successful, following the speech recognition area as we reviewed and analyzed in Sections 2 and 7.

Excellent surveys on the recent progress of deep learning for computer vision are available in the NIPS-2013 tutorial (https://nips.cc/Conferences/2013/Program/event.php?ID=4170 with video recording at http://research.microsoft.com/apps/video/default.aspx?id=206976&l=i) and slides at http://cs.nyu.edu/~fergus/presentations/nips2013_final.pdf, and also in the CVPR-2012 tutorial (http://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12). The reviews provided in this section below are based partly on these tutorials, in connection with the earlier deep learning material in this monograph. Another excellent source which this section draws from is the most recent Ph.D. thesis on the topic of deep learning for computer vision [434].

Over many years, object recognition in computer vision has been relying on hand-designed features such as SIFT (scale invariant feature transform) and HOG (histogram of oriented gradients), akin to the reliance of speech recognition on hand-designed features such as MFCC and PLP. However, features like SIFT and HOG only capture low-level edge information. The design of features to effectively capture mid-level information such as edge intersections or high-level representation such as object parts becomes much more difficult. Deep learning aims to overcome such challenges by automatically learning hierarchies of visual features in both unsupervised and supervised manners directly from data. The review below categorizes the many deep learning methods applied to computer vision into two classes: (1) unsupervised feature learning where the deep learning is used to extract features only, which may be subsequently fed to relatively simple machine learning algorithm for classification or other tasks; and (2) supervised learning methods where end-to-end learning is adopted to jointly optimize feature extractor and classifier components of the full system when large amounts of labeled training data are available.

10.1 Unsupervised or generative feature learning

When labeled data are relatively scarce, unsupervised learning algorithms have been shown to learn useful visual feature hierarchies. In fact, prior to the demonstration of remarkable successes of CNN architectures with supervised learning in the 2012 ImageNet competition, much of the work in applying deep learning methods to computer vision had been on unsupervised feature learning. The original unsupervised deep autoencoder that exploits DBN pre-training was developed and demonstrated by Hinton and Salakhutdinov [164] with success on the image recognition and dimensionality reduction (coding) tasks of MNIST with only 60,000 samples in the training set; see details of this task in http://yann.lecun.com/exdb/mnist/ and an analysis in [78]. It is interesting to note that the gain of coding efficiency using the DBN-based autoencoder on the image data over the conventional method of principal component analysis as demonstrated in [164] is very similar to

the gain reported in [100] and described in Section 4 of this monograph on the speech data over the traditional technique of vector quantization. Also, Nair and Hinton [265] developed a modified DBN where the top-layer model uses a third-order Boltzmann machine. This type of DBN is applied to the NORB database — a three-dimensional object recognition task. An error rate close to the best published result on this task is reported. In particular, it is shown that the DBN substantially outperforms shallow models such as SVMs. In [358], two strategies to improve the robustness of the DBN are developed. First, sparse connections in the first layer of the DBN are used as a way to regularize the model. Second, a probabilistic de-noising algorithm is developed. Both techniques are shown to be effective in improving robustness against occlusion and random noise in a noisy image recognition task. DBNs have also been successfully applied to create compact but meaningful representations of images [360] for retrieval purposes. On this large collection image retrieval task, deep learning approaches also produced strong results. Further, the use of a temporally conditional DBN for video sequence and human motion synthesis were reported in [361]. The conditional RBM and DBN make the RBM and DBN weights associated with a fixed time window conditioned on the data from previous time steps. The computational tool offered in this type of temporal DBN and the related recurrent networks may provide the opportunity to improve the DBN–HMMs towards efficient integration of temporal-centric human speech production mechanisms into DBN-based speech production model.

Deep learning methods have a rich family, including hierarchical probabilistic and generative models (neural networks or otherwise). One most recent example of this type developed and applied to facial expression datasets is the stochastic feed-forward neural networks that can be learned efficiently and that can induce a rich multiple-mode distribution in the output space not possible with the standard, deterministic neural networks [359]. In Figure 10.1, we show the architecture of a typical stochastic feed-forward neural network with four hidden layers with mixed deterministic and stochastic neurons (left) used to model multi-mode distributions illustrated on the right. The stochastic network here is a deep, directed graphical model, where the generation

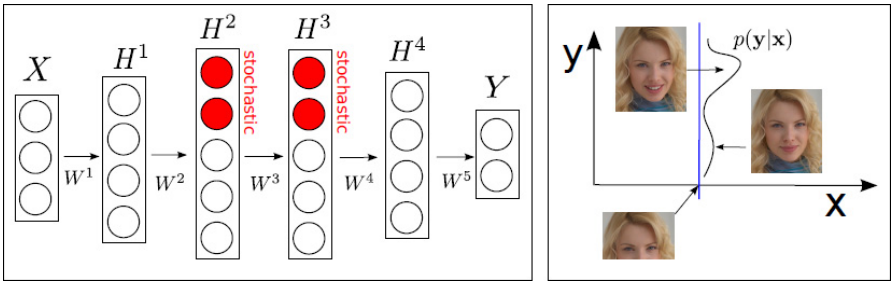

Figure 10.1: Left: A typical architecture of the stochastic feed-forward neural network with four hidden layers. Right: Illustration of how the network can produce a distribution with two distinct modes and use them to represent two or more different facial expressions $\mathbf{y}$ given a neutral face $\mathbf{x}$. [after [359], @NIPS].

process starts from input $\mathbf{x}$, a neural face, and generates the output $\mathbf{y}$, the facial expression. In face expression classification experiments, the learned unsupervised hidden features generated from this stochastic network are appended to the image pixels and helped to obtain superior accuracy to the baseline classifier based on the conditional RBM/DBN [361].

Perhaps the most notable work in the category of unsupervised deep feature learning for computer vision (prior to the recent surge of the work on CNNs) is that of [209], a nine-layer locally connected sparse autoencoder with pooling and local contrast normalization. The model has one billion connections, trained on the dataset with 10 million images downloaded from the Internet. The unsupervised feature learning methods allow the system to train a face detector without having to label images as containing a face or not. And the control experiments show that this feature detector is robust not only to translation but also to scaling and out-of-plane rotation.

Another set of popular studies on unsupervised deep feature learning for computer vision are based on deep sparse coding models [226]. This type of deep models produced state-of-the-art accuracy results on the ImageNet object recognition tasks prior to the rise of the CNN architectures armed with supervised learning to perform joint feature learning and classification, which we turn to now.

10.2 Supervised feature learning and classification

The origin of the applications of deep learning to object recognition tasks can be traced to the convolutional neural networks (CNNs) in the early 90s; see a comprehensive overview in [212]. The CNN-based architectures in the supervised learning mode have captured intense interest in computer vision since October 2012 shortly after the ImageNet competition results were released (http://www.image-net.org/challenges/LSVRC/2012/). This is mainly due to the huge recognition accuracy gain over competing approaches when large amounts of labeled data are available to efficiently train large CNNs using GPU-like high-performance computing platforms. Just like DNN-based deep learning methods have outperformed previous state-of-the-art approaches in speech recognition in a series of benchmark tasks including phone recognition, large-vocabulary speech recognition, noise-robust speech recognition, and multi-lingual speech recognition, CNN-based deep learning methods have demonstrated the same in a set of computer vision benchmark tasks including category-level object recognition, object detection, and semantic segmentation.

The basic architecture of the CNN described in [212] is shown in Figure 10.1. To incorporate the relative invariance of the spatial relationship in typical image pixels with respect to the location, the CNN uses a convolutional layer with local receptive fields and with tied filter weights, much like 2-dimensional FIR filters in image processing. The output of the FIR filters is then passed through a nonlinear activation function to create activation maps, followed by another nonlinear pooling (labeled as "subsampling" in Figure 10.2) layer that reduces the data rate while providing invariance to slightly different input images. The output of the pooling layer is fed to a few fully connected layers as in the DNN discussed in earlier chapters. The whole architecture above is also called the deep CNN in the literature.

Deep models with convolution structure such as CNNs have been found effective and have been in use in computer vision and image recognition since 90s [57, 185, 192, 198, 212]. The most notable advance was achieved in the 2012 ImageNet LSVRC competition, in which

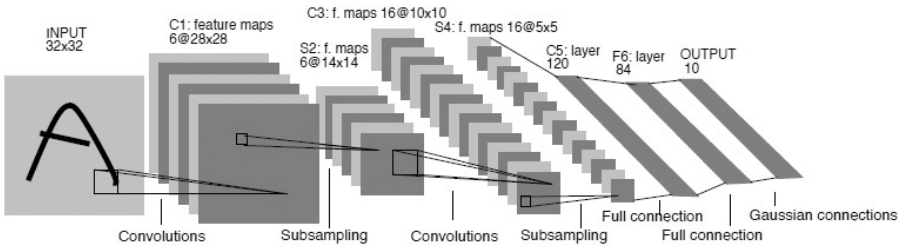

Figure 10.2: The original convolutional neural network that is composed of multiple alternating convolution and pooling layers followed by fully connected layers. [after [212], @IEEE].

the task is to train a model with 1.2 million high-resolution images to classify unseen images to one of the 1000 different image classes. On the test set consisting of 150k images, the deep CNN approach described in [198] achieved the error rates considerably lower than the previous state-of-the-art. Very large deep-CNNs are used, consisting of 60 million weights, and 650,000 neurons, and five convolutional layers together with max-pooling layers. Additional two fully-connected layers as in the DNN described previously are used on top of the CNN layers. Although all the above structures were developed separately in earlier work, their best combination accounted for major part of the success. See the overall architecture of the deep CNN system in Figure 10.3. Two additional factors contribute to the final success. The first is a powerful regularization technique called "dropout"; see details in [166] and a series of further analysis and improvement in [10, 13, 240, 381, 385]. In particular, Warde-Farley et al. [385] analyzed the disentangling effects of dropout and showed that it helps because different members of the bag share parameters. Applications of the same "dropout" techniques are also successful for some speech recognition tasks [65, 81]. The second factor is the use of non-saturating neurons or rectified linear units (ReLU) that compute $f(x) = \max(x, 0)$, which significantly speeds up the overall training process especially with efficient GPU implementation. This deep-CNN system achieved a winning top-5 test error rate of 15.3% using extra training data from ImageNet Fall 2011 release, or 16.4% using only supplied training data in ImageNet-2012,

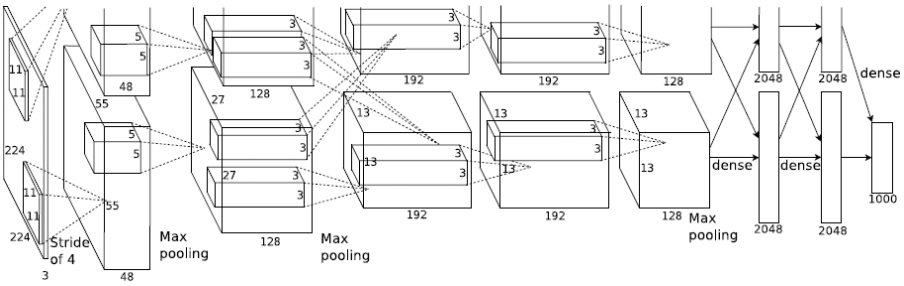

Figure 10.3: The architecture of the deep-CNN system which won the 2012 ImageNet competition by a large margin over the second-best system and the state of the art by 2012. [after [198], @NIPS].

significantly lower than 26.2% achieved by the second-best system which combines scores from many classifiers using a set of hand-crafted features such as SIFT and Fisher vectors. See details in http://www.image-net.org/challenges/LSVRC/2012/oxford_vgg.pdf about the best competing method. It is noted, however, that the Fisher-vector-encoding approach has recently been extended by Simonyan et al. [329] via stacking in multiple layers to form deep Fisher networks, which achieve competitive results with deep CNNs at a smaller computational learning cost.

The state of the art performance demonstrated in [198] using the deep-CNN approach is further improved by another significant margin during 2013, using a similar approach but with bigger models and larger amounts of training data. A summary of top-5 test error rates from 11 top-performing teams participating in the 2013 ImageNet ILSVRC competition is shown in Figure 10.4, with the best result of the 2012 competition shown to the right most as the baseline. Here we see rapid error reduction on the same task from the lowest pre-2012 error rate of 26.2% (non-neural networks) to 15.3% in 2012 and further to 11.2% in 2013, both achieved with deep-CNN technology. It is also interesting to observe that all major entries in the 2013 ImageNet ILSVRC competition is based on deep learning approaches. For example, the Adobe system shown in Figure 10.4 is based on the deep-CNN reported in [198] including the use of dropout. The network

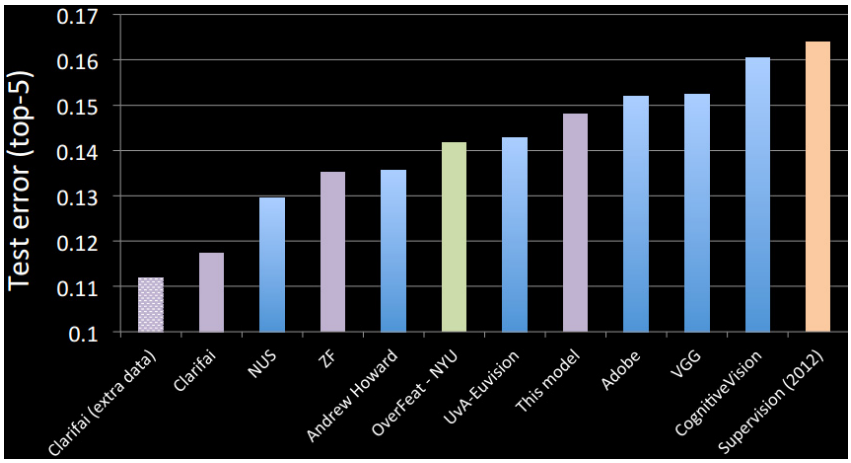

Figure 10.4: Summary results of ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013), representing the state-of-the-are performance of object recognition systems. Data source: http://www.image-net.org/challenges/LSVRC/2013/results.php.

architecture is modified to include more filters and connections. At test time, image saliency is used to obtain 9 crops from original images, which are combined with the standard five multiview crops. The NUS system uses a non-parametric, adaptive method to combine the outputs from multiple shallow and deep experts, including deep-CNN, kernel, and GMM methods. The VGG system is described in [329] and uses a combination of the deep Fisher vector network and the deep-CNN. The ZF system is based on a combination of a large CNN with a range of different architectures. The choice of architectures was assisted by visualization of model features using a deconvolutional network as described by Zeiler et al. [437], Zeiler and Fergus [435, 436], and Zeiler ([434]). The CognitiveVision system uses an image classification scheme based on a DNN architecture. The method is inspired by cognitive psychophysics about how the human vision system first learns to classify the basic-level categories and then learns to classify categories at the subordinate level for fine-grained object recognition. Finally, the best-performing system called Clarifai in Figure 10.4 is based on a large and deep CNN with dropout regularization. It

augments the amount of training data by down-sampling images to 256 pixels. The system contains a total of 65M parameters. Multiple such models were averaged together to further boost performance. The main novelty is to use the visualization technique based on the deconvolutional networks as described in [434, 437] to identify what makes the deep model perform well, based on which a powerful deep architecture was chosen. See more details of these systems in http://www.image-net.org/challenges/LSVRC/2013/results.php.

While the deep CNN has demonstrated remarkable classification performance on object recognition tasks, there has been no clear understanding of why they perform so well until recently. Zeiler and Fergus [435, 436] conducted research to address just this issue, and then used the gained understanding to further improve the CNN systems, which yielded excellent performance as shown in Figure 10.4 with labels "ZF" and "Clarifai." A novel visualization technique is developed that gives insight into the function of intermediate feature layers of the deep CNN. The technique also sheds light onto the operation of the full network acting as a classifier. The visualization technique is based on a deconvolutional network, which maps the neural activities in intermediate layers of the original convolutional network back to the input pixel space. This allows the researchers to examine what input pattern originally caused a given activation in the feature maps. Figure 10.5 (the top portion) illustrates how a deconvolutional network is attached to each of its layers, thereby providing a closed loop back to image pixels as the input to the original CNN. The information flow in this closed loop is as follows. First, an input image is presented to the deep CNN in a feed-forward manner so that the features at all layers are computed. To examine a given CNN activation, all other activations in the layer are set to zero and the feature maps are passed as input to the attached deconvolutional network's layer. Then, successive operations, opposite to the feed-forward computation in the CNN, are carried out including unpooling, rectifying, and filtering. This allows the reconstruction of the activity in the layer beneath that gave rise to the chosen activation. These operations are repeated until input layer is reached. During unpooling, non-invertibility of the max pooling operation in the CNN is

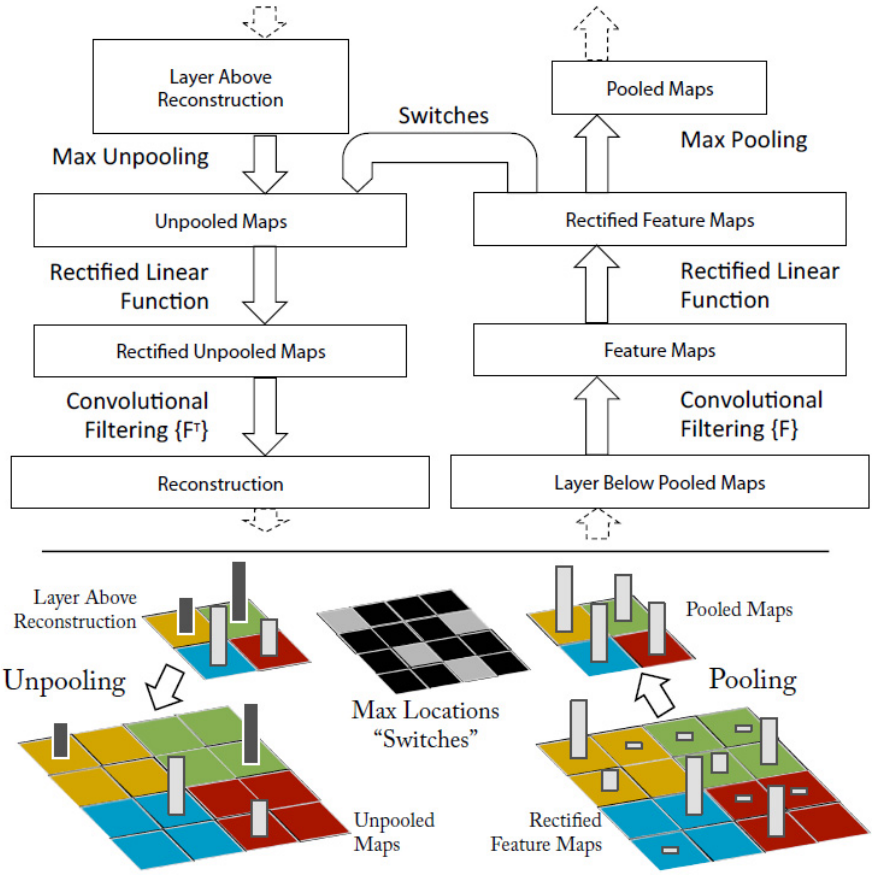

Figure 10.5: The top portion shows how a deconvolutional network's layer (left) is attached to a corresponding CNN's layer (right). The d econvolutional network reconstructs an approximate version of the CNN features from the layer below. The bottom portion is an illustration of the unpooling operation in the deconvolutional network, where "Switches" are used to record the location of the local max in each pooling region during pooling in the CNN. [after [436], @arXiv].

resolved by an approximate inverse, where the locations of the maxima within each pooling region are recorded in a set of "switch" variables. These switches are used to place the reconstructions from the layer above into appropriate locations, preserving the structure of the stimulus. This procedure is shown at the bottom portion of Figure 10.5.

In addition to the deep-CNN architecture described above, the DNN architecture has also been shown to be highly successful in a number of computer vision tasks [54, 55, 56, 57]. We have not found in the literature on direct comparisons among the CNN, DNN, and other related architectures on the identical tasks.

Finally, the most recent study on supervised learning for computer vision shows that the deep CNN architecture is not only successful for object/image classification discussed earlier in this section but also successful for objection detection in the whole images [128]. The detection task is substantially more complex than the classification task.

As a brief summary of this chapter, deep learning has made huge inroads into computer vision, soon after its success in speech recognition discussed in Section 7. So far, it is the supervised learning paradigm based on the deep CNN architecture and the related classification techniques that are making the greatest impact, showcased by the ImageNet competition results from 2012 and 2013. These methods can be used for not only object recognition but also many other computer vision tasks. There has been some debate as to the reasons for the success of these CNN-based deep learning methods, and about their limitations. Many questions are still open as to how these methods can be tailored to certain computer vision applications and how to scale up the models and training data. Finally, we discussed a number of studies on unsupervised and generative approaches of deep learning to computer vision and image modeling problems in the earlier part of this chapter. Their performance has not been competitive with the supervised learning approach on object recognition tasks with ample training data. To achieve long term and ultimate success in computer vision, it is likely that unsupervised learning will be needed. To this end, many open problems in unsupervised feature learning and deep learning need to be addressed and much more research need to be carried out.