21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Convolutional Neural Networks Applied to House Numbers Digit Classification

Pierre Sermanet, Soumith Chintala and Yann LeCun
The Courant Institute of Mathematical Sciences - New York University
{*sermanet,soumith,yann*}*@cs.nyu.edu*

Abstract

We classify digits of real-world house numbers using convolutional neural networks (ConvNets). ConvNets are hierarchical feature learning neural networks whose structure is biologically inspired. Unlike many popular vision approaches that are hand-designed, ConvNets can automatically learn a unique set of features optimized for a given task. We augmented the traditional ConvNet architecture by learning multi-stage features and by using Lp pooling and establish a new state-of-the-art of 95.10% *accuracy on the SVHN dataset* (48% *error improvement). Furthermore, we analyze the benefits of different pooling methods and multi-stage features in ConvNets. The source code and a tutorial are available at eblearn.sf.net.*

1. Introduction

Character recognition in documents can be considered a solved task for computer vision, whether handwritten or typed. It is however a harder problem in the context of complex natural scenes like photographs where the best current methods lag behind human performance, mainly due to non-contrasting backgrounds, low resolution, de-focused and motion-blurred images and large illumination differences (Figure 1).

[9] recently introduced a new digit classification dataset of house numbers extracted from street level images. It is similar in format to the popular MNIST dataset[8] (10 digits, 32x32 inputs), but an order of magnitude bigger (600,000 labeled digits), contains color information and various natural backgrounds.

Previous approaches in classifying characters and digits from natural images used multiple hand-crafted features [3] and template-matching [15]. In contrast, ConvNets learn features all the way from pixels to the classifier. [9] demonstrated the superiority of learned features over hand-designed ones. Such superiority was also previously shown among others in a traffic sign classification challenge [14] where two independent teams obtained the best performance against various other approaches using ConvNets [12, 2]. [9] also show superior results with unsupervised learning, we however only report results with fully-supervised training. We obtain a 4.5 points improvement in accuracy (with 95.10% accuracy) over the previous state-of-the-art of 90.6%. We use the traditional ConvNet architecture augmented with different pooling methods and with multi-stage features [12]. This work was implemented with the EBLearn [1] C++ open-source framework [11].

Figure 1. 32x32 cropped samples from the classification task of the SVHN dataset. Each sample is assigned only a single digit label (0 to 9) corresponding to the center digit.

[1] http://eblearn.sf.net

2 Architecture

The ConvNet architecture is composed of repeatedly stacked feature stages. Each stage contains a convolution module, followed by a pooling/subsampling module and a normalization module. While traditional pooling modules in ConvNet are either average or max poolings, we use an Lp pooling here. The normalization module is subtractive only as opposed to subtractive and divisive, i.e. the mean value of each neighborhood is subtracted to the output of each stage (but not divided by the standard deviation as it decreases performance with this dataset). Finally, multi-stage features are also used as opposed to single-stage features. This architecture is trained using stochastic gradient descent (SGD) with the Levenberg-Marquardt diagonal approximation to the Hessian [7].

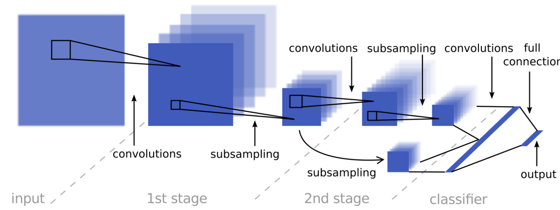

Figure 3. A 2-stage ConvNet architecture where Multi-Stage features (MS) are fed to a 2-layer classifier. The 1st stage features are branched out, subsampled again and then concatenated to 2nd stage features.

2.1 Lp-Pooling

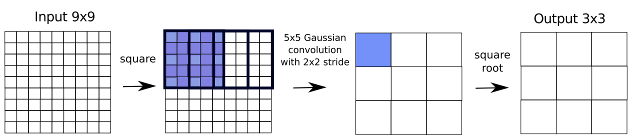

Figure 2. L2-pooling applied to a 9x9 feature map with a 3x3 Gaussian kernel and 2x2 stride

Lp pooling is a biologically inspired pooling layer modelled on complex cells [13, 5] who's operation can be summarized in equation (1), where G is a Gaussian kernel, I is the input feature map and O is the output feature map. It can be imagined as giving an increased weight to stronger features and suppressing weaker features. Two special cases of Lp pooling are notable. $P = 1$ corresponds to a simple Gaussian averaging, whereas $P = \infty$ corresponds to max-pooling (i.e only the strongest signal is activated). Lp-pooling has been used previously in [6, 16] and a theoretical analysis of this method is described in [1].

$$O = (\sum\sum I(i,j)^P \times G(i,j))^{1/P} \qquad (1)$$

Figure 2 demonstrates a simple example of L2-pooling.

2.2 Multi-Stage Features

Multi-Stage features (MS) are obtained by branching out outputs of all stages into the classifier (Figure 3). They provide richer representations compared to Single-Stage features (SS) by adding complementary information such as local textures and fine details lost by higher levels. MS features have consistently improved performance in other work [4, 12, 10] and in this work as well (Figure 4). However we observe minimal gains on this dataset compared to other types of objects such as pedestrians and traffic signs (Table 1). The likely explanation for this observation is that gains are correlated to the amount of texture and multi-scale characteristics of the objects of interest.

3. Experiments

3.1. Data Preparation

The SVHN classification dataset [9] contains 32x32 images with 3 color channels. The dataset is divided into three subsets: train set, extra set and test set. The extra set is a large set of easy samples and train set is a smaller set of more difficult samples. Since we are given no information about how the sampling of these images was done, we assume a random order to construct our validation set. We compose our validation set with $2/3$ from training samples (400 per class) and $1/3$ from extra samples (200 per class), yielding a total of 6000 samples. This distribution allows to measure success on easy samples but puts more emphasis on difficult ones. The training and testing sets contain respectively 598388 and 26032 samples.

Samples are pre-processed with a local contrast normalization (with a 7x7 kernel) on the Y channel of the YUV space followed by a global contrast normalization over each channel. No sample distortions were used to improve invariance. For some experiments, a padding of 2 pixels with zero value was added to each side of the input image in order to center the first stage's 5x5 filters onto image borders.

3.2 Architecture Details

The ConvNet has 2 stages of feature extraction and a two-layer non-linear classifier. The first convolution layer produces 16 features with 5x5 convolution filters while the second convolution layer outputs 512 features with 7x7 filters. The output to the classifier also includes inputs from the first layer, which provides lo-

Task	Single-Stage features	Multi-Stage features	Improvement %
Pedestrians detection (INRIA) [10]	14.26%	9.85%	31%
Traffic Signs classification (GTSRB) [12]	1.80%	0.83%	54%
House Numbers classification (SVHN)	5.54%	5.36%	3.2%

Table 1. Error rates improvements of multi-stage features over single-stage features for different types of objects detection and classification. Improvements are significant for multi-scale and textured objects such as traffic signs and pedestrians but minimal for house numbers.

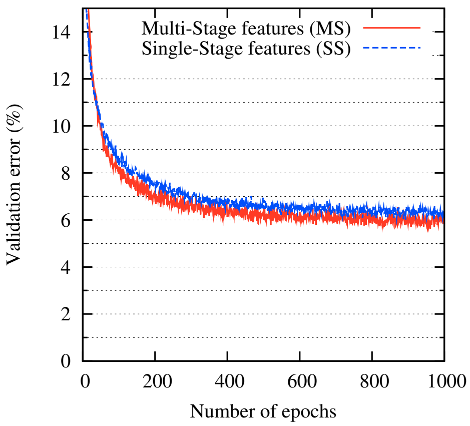

Figure 4. Improvement of Multi-Stage features (MS) over Single-Stage features (SS) in error rate on the validation set. MS features provide a slight error improvement over SS features.

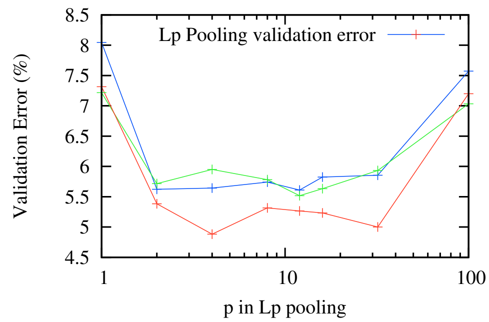

Figure 5. Error rate of Lp-pooling on 3 cross-validation sets for $p = 1, 2, 4, 8, 12, 16, 32, \infty$ ($p = \infty$ is represented as $p = 100$ for convenience). These validation errors are reported after 1000 training epochs.

cal features/motifs to reinforce the global features. The classifier is a 2-layer non-linear classifier with 20 hidden units. Hyper-parameters such as learning rate, regularization constant and learning rate decay were tuned on the validation set. We use stochastic gradient descent as our optimization method and shuffle our dataset after each training iteration.

For the pooling layers, we compare Lp-pooling for the value $p = 1, 2, 4, 8, 12, 16, 32, \infty$ on the validation set and use the best performing pooling on the final testing. The performance of different pooling methods on the validation set can be seen in Figure 5. Insights from [1] tell us that the optimal value of p varies for different input spaces and there is no single globally optimal value for p. With validation data, we observe that $p = 2, 4, 12$ give the best performance ($5.62\%, 5.64\%$ and 5.61% respectively). Max-pooling ($p = \infty$) yielded a validation error rate of 7.57%.

4 Results & Future Work

Our experiments demonstrate a clear advantage of Lp pooling with $1 < p < \infty$ on this dataset, in validation (Figure 5) and test (L2 pooling is 3.58 points superior to average pooling in Table 2). With L4 pooling, we obtain a state-of-the-art performance on the test set with an accuracy of 95.10% compared to the previous best accuracy of 90.6% (Table 2). Padding around inputs improves accuracy by 0.13% points from the 94.97% non-padded accuracy. This is likely explained by digit edges being very close to the image borders as seen in Figure 6. Padding allows centered edge filters to fire correctly at the borders. We also show that using multi-stage features gives only a slight increase in performance, compared to the performance increase seen in other vision applications. In concordance with [9], we show by comparing "L2" and "L2 / Smaller training" that using the full dataset (604388 samples) rather than the hard dataset (73257 samples) improves the accuracy by about 3 points.

Additionally, it is important to note that our approach is trained fully supervised only, whereas the best previous methods are unsupervised learning methods (k-means, auto-encoders). We shall, in the future, run experiments with unsupervised learning to determine the gains of ConvNet architectures versus others. Figure 6 shows the validation samples with highest energy. Many of these seem to exhibit large scale variations, future work could address this problem by introducing artificial scale, elastic and noise deformations during training.

Algorithm	SVHN-Test Accuracy
Binary Features (WDCH)	63.3%
HOG	85.0%
Stacked Sparse Auto-Encoders	89.7 %
K-Means	90.6%
ConvNet / MS / Average Pooling	**90.94%**
ConvNet / MS / L2 / Smaller training	**91.55%**
ConvNet / SS / L2	**94.46%**
ConvNet / MS / L2	**94.64%**
ConvNet / MS / L12	**94.89%**
ConvNet / MS / L4	**94.97%**
ConvNet / MS / L4 / Padded	**95.10%**
Human Performance	98.0%

Table 2. Performance reported by [9] with the additional Supervised ConvNet with state-of-the-art accuracy of 95.10%.

Figure 6. Preprocessed Y channel of validation samples with highest energy (i.e. highest error) with the 94.64% accuracy L2-pool based multi-stage ConvNet.

References

[1] Y. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in vision algorithms. In *Proc. International Conference on Machine learning*, 2010.

[2] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. A committee of neural networks for traffic sign classification. In *International Joint Conference on Neural Networks*, pages 1918–1921, 2011.

[3] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, February 2009.

[4] J. Fan, W. Xu, Y. Wu, and Y. Gong. Human tracking using convolutional neural networks. *Neural Networks, IEEE Transactions on*, 21(10):1610 –1623, 2010.

[5] A. Hyvrinen and U. Kster. Complex cell pooling and the statistics of natural images. In *Computation in Neural Systems,*, 2005.

[6] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In *Proc. International Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[8] Y. Lecun and C. Cortes. The MNIST database of handwritten digits.

[9] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[10] P. Sermanet, K. Kavukcuoglu, and Y. LeCun. Traffic signs and pedestrians vision with multi-scale convolutional networks. In *Snowbird Machine Learning Workshop, 2011*.

[11] P. Sermanet, K. Kavukcuoglu, and Y. LeCun. Eblearn: Open-source energy-based learning in c++. In *Proc. International Conference on Tools with Artificial Intelligence*. IEEE, 2009.

[12] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Proceedings of International Joint Conference on Neural Networks*, 2011.

[13] E. P. Simoncelli and D. J. Heeger. A model of neuronal responses in visual area mt, 1997.

[14] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pages 1453–1460, 2011.

[15] T. Yamaguchi, Y. Nakano, M. Maruyama, H. Miyao, and T. Hananoi. Digit classification on signboards for telephone number recognition. In *ICDAR*, pages 359–363, 2003.

[16] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *in IEEE Conference on Computer Vision and Pattern Recognition*, 2009.