Multi-column Deep Neural Networks for Image Classification Supplementary Online Material

Dan Cireşan, Ueli Meier and Jürgen Schmidhuber
IDSIA-USI-SUPSI
Galleria 2, 6928 Manno-Lugano, Switzerland
{dan,ueli,juergen}@idsia.ch

1. Experiment details

1.1. NIST SD 19

The confusion matrix of the 62 characters task (Fig. 1) shows that most of the errors are due to confusions between digits and letters and between lower- and upper-case letters.

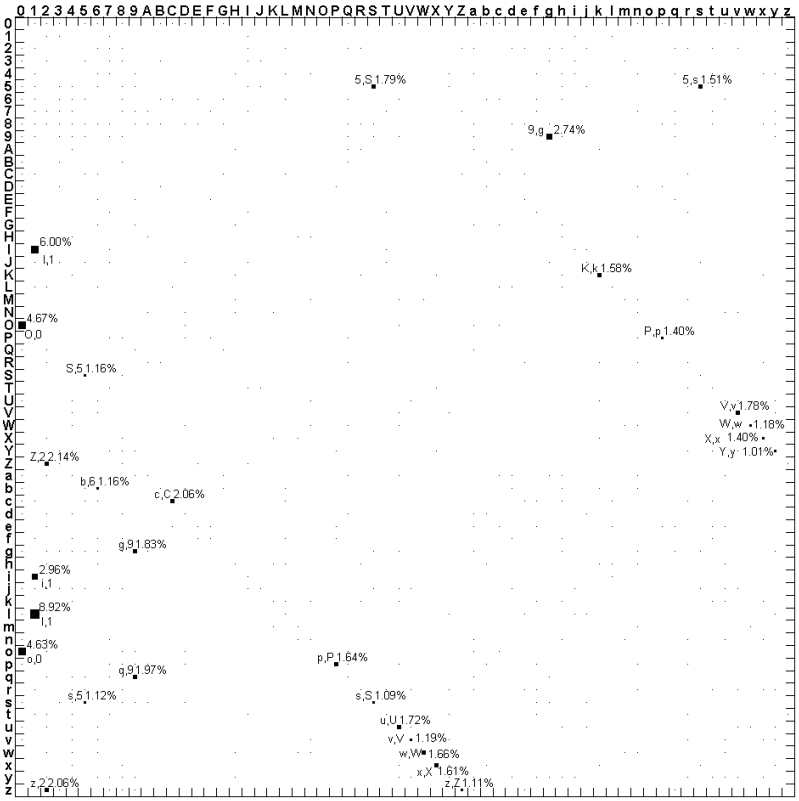

Figure 1. Confusion matrix of the NIST SD 19 MCDNN trained on the 62-class task: correct labels on vertical axis; detected labels on horizontal axis. Square areas are proportional to error numbers, shown as relative percentages of the total error number. For convenience, class labels are written beneath the errors. Errors below 1% of the total error number are not detailed.

Not very surprisingly, the confusion matrix for the digit task (Fig. 2) shows that confusions between fours and nines are the most common error source.

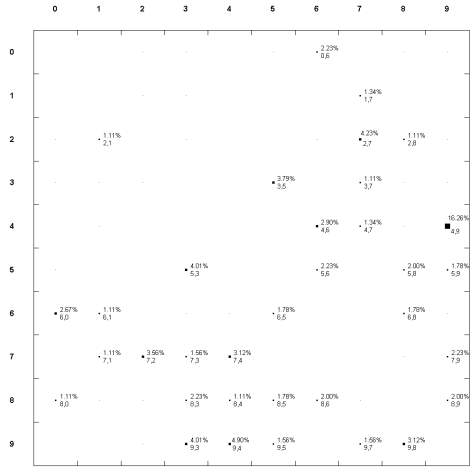

Figure 2. Confusion matrix for the NIST SD 19 MCDNN trained on the digit task: correct labels on vertical axis; detected labels on horizontal axis. Square areas are proportional to error numbers, shown as relative percentages of the total error number. Class labels are shown beneath the errors. Errors below 1% of the total error number are shown as dots without any details.

For the 52 letter task (case sensitive) the confusion matrix (Fig. 3) shows that the MCDNN has mainly problems with upper- and lower-case confusions of the same letter. Other hard-to-distinguish classes are: 'q' and 'g', 'l' and 'i'.

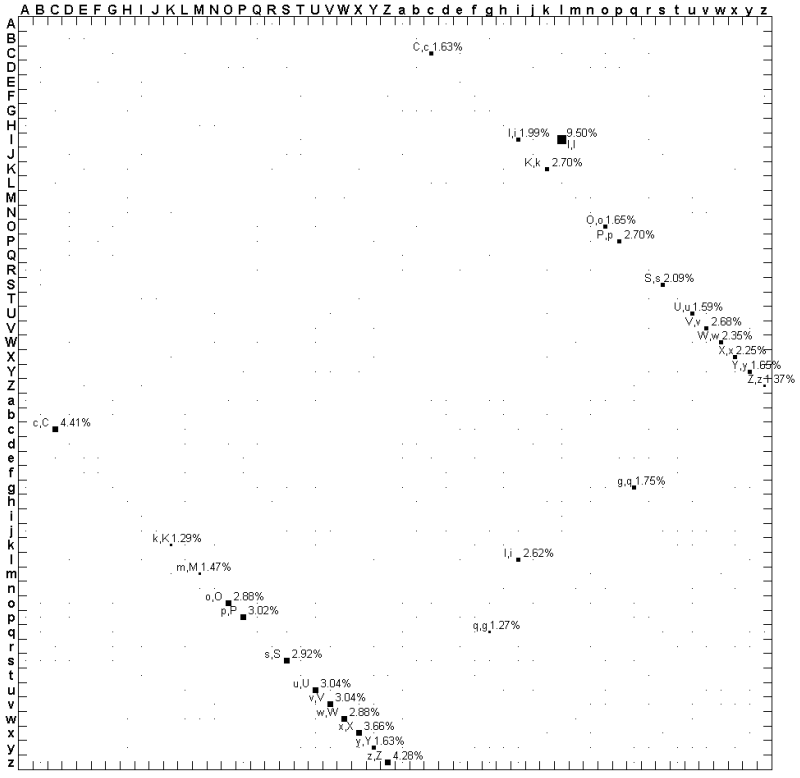

Figure 3. Confusion matrix for the NIST SD 19 MCDNN trained on all 52 letters: correct labels on vertical axis; detected labels on horizontal axis. Square areas are proportional to error numbers, shown as relative percentages of the total error number. Class labels are shown beneath the errors. Errors below 1% of the total error number are shown as dots without any details.

For the upper-case letter task the confusion matrix (Figure 4) shows that the MCDNN has problems with letters of similar

shape, i.e. 'D', and 'O', 'V' and 'U' etc. The total error of 1.82% is very low though.

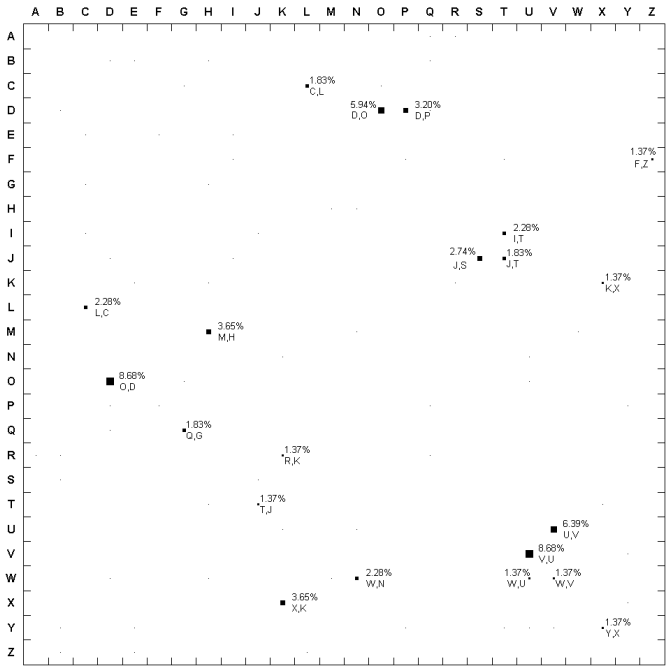

Figure 4. Confusion matrix for the NIST SD 19 MCDNN trained on uppercase letters: correct labels on vertical axis; detected labels on horizontal axis. Square areas are proportional to error numbers, shown as relative percentages of the total error number. Class labels are shown beneath the errors. Errors below 1% of the total error number are shown as dots without any details.

For the lower-case letter task the confusion matrix (Fig. 5) shows that like with upper-case letters, the MCDNN has problems with letters of similar shapes, i.e. 'g', and 'q', 'v' and 'u' etc. But the total error is much higher (7.47%) than for the upper-case letters task.

For the merged-case letter task (37 classes) the confusion matrix (Figure 6) shows that the MCDNN has mostly problems with letters of similar shapes, i.e. 'l', and 'i'. All upper-lower-case confusions of identical letters from the 52 class task vanish, the error shrinks by a factor of almost three down to 7.99%.

The experiments on different subsets of the 62 character task clearly show that it is very hard to distinguish between small and capital letters. Also, digits 0 and 1 are hard to separate from letters O and I. Many of these problems could be alleviated by incorporating context where possible.

1.2. Traffic signs

High contrast variation among the images calls for normalization. We test the following standard contrast normalizations:

- **Image Adjustment (Imadjust)** increases image contrast by mapping pixel intensities to new values such that 1% of the data is saturated at low and high intensities [1].
- **Histogram Equalization (Histeq)** enhances contrast by transforming pixel intensities such that the output image histogram is roughly uniform [1].
- **Adaptive Histogram Equalization (Adapthisteq)** operates (unlike Histeq) on tiles rather than the entire image, we tiled the image in 8 nonoverlapping regions of 6x6 pixels. Each tile's contrast is enhanced such that its histogram becomes roughly uniform [1].
- **Contrast Normalization (Conorm)** enhances edges, filtering the input image by a difference of Gaussians, using a filter size of 5x5 pixels [2].

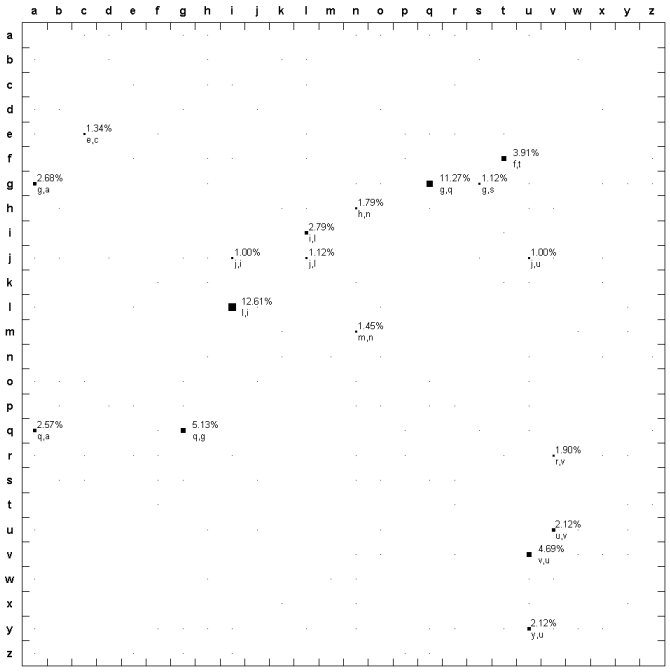

Figure 5. Confusion matrix for the NIST SD 19 MCDNN trained on lowercase letters: correct labels on vertical axis; detected labels on horizontal axis. Square areas are proportional to error numbers, shown as relative percentages of the total error number. Class labels are shown beneath the errors. Errors below 1% of the total error number are shown as dots without any details.

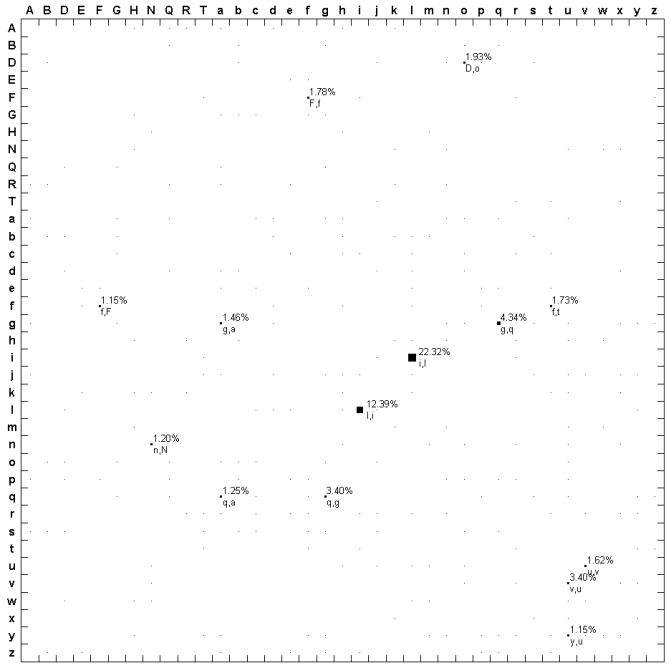

Figure 6. Confusion matrix for the NIST SD 19 MCDNN trained on merged letters (37 classes): correct labels on vertical axis; detected labels on horizontal axis. Square areas are proportional to error numbers, shown as relative percentages of the total error number. Class labels are shown beneath the errors. Errors below 1% of the total error number are shown as dots without any details.

Note that the above normalizations, except Conorm, are performed in a color space with image intensity as one of its components. For this purpose we transform the image from RGB- to Lab-space, then perform normalization, then transform the result back to RGB-space. The effect of the four different normalizations is summarized in Figure 7, where histograms of pixel intensities together with original and normalized images are shown.

The DNN have three maps for the input layer, one for each color channel (RGB). The rest of the net architecture is detailed in Table 1. We use a 10-layer architecture with very small max-pooling kernels.

Table 1. 10 layer DNN architecture used for recognizing traffic signs.

Layer	Type	# maps & neurons	kernel
0	input	3 maps of 48x48 neurons	
1	convolutional	100 maps of 42x42 neurons	7x7
2	max pooling	100 maps of 21x21 neurons	2x2
3	convolutional	150 maps of 18x18 neurons	4x4
4	max pooling	150 maps of 9x9 neurons	2x2
5	convolutional	250 maps of 6x6 neurons	4x4
6	max pooling	250 maps of 3x3 neurons	2x2
9	fully connected	300 neurons	1x1
10	fully connected	43 neurons	1x1

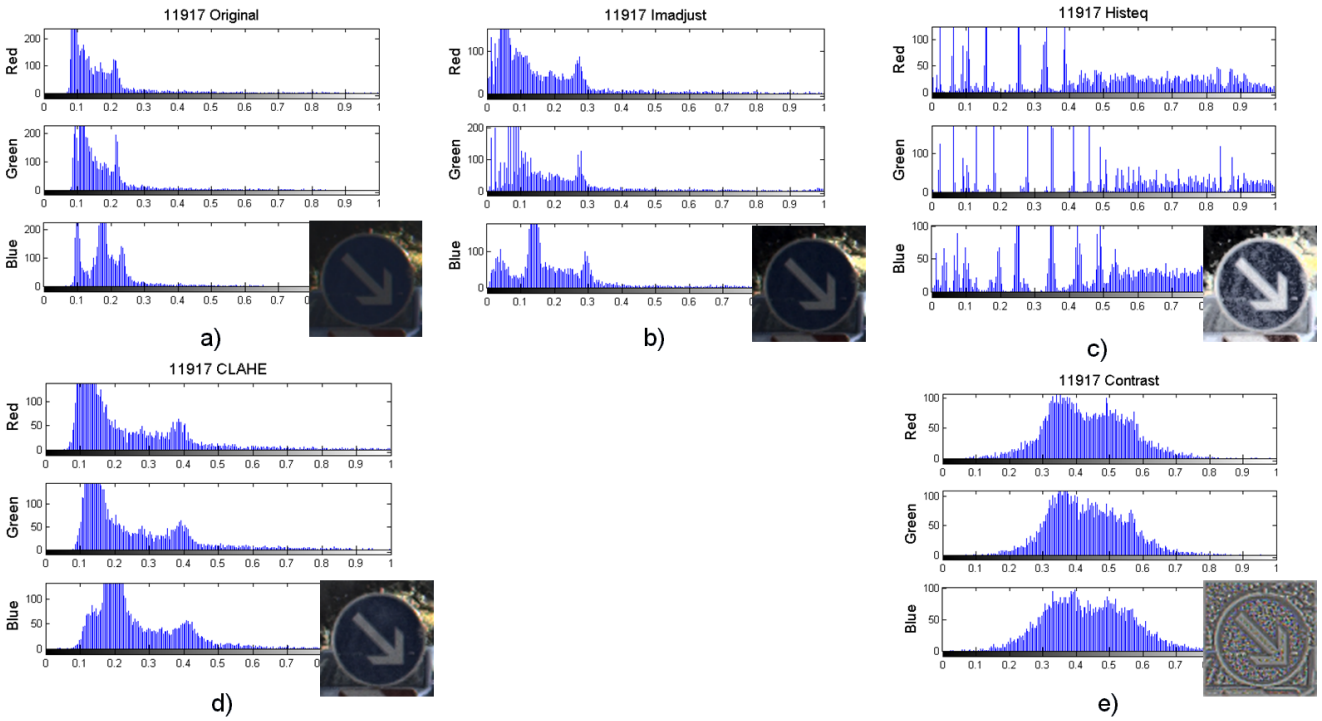

Figure 7. Preprocessing.

1.3. CIFAR10

Table 2. 10 layer DNN architecture used for CIFAR 10.

Layer	Type	# maps & neurons	kernel
0	input	3 maps of 32x32 neurons	
1	convolutional	300 maps of 30x30 neurons	3x3
2	max pooling	300 maps of 15x15 neurons	2x2
3	convolutional	300 maps of 14x14 neurons	2x2
4	max pooling	300 maps of 7x7 neurons	2x2
5	convolutional	300 maps of 6x6 neurons	2x2
6	max pooling	300 maps of 3x3 neurons	2x2
7	convolutional	300 maps of 2x2 neurons	2x2
8	max pooling	300 maps of 1x1 neurons	2x2
9	fully connected	300 neurons	1x1
10	fully connected	100 neurons	1x1
11	fully connected	10 neurons	1x1

1.4. NORB

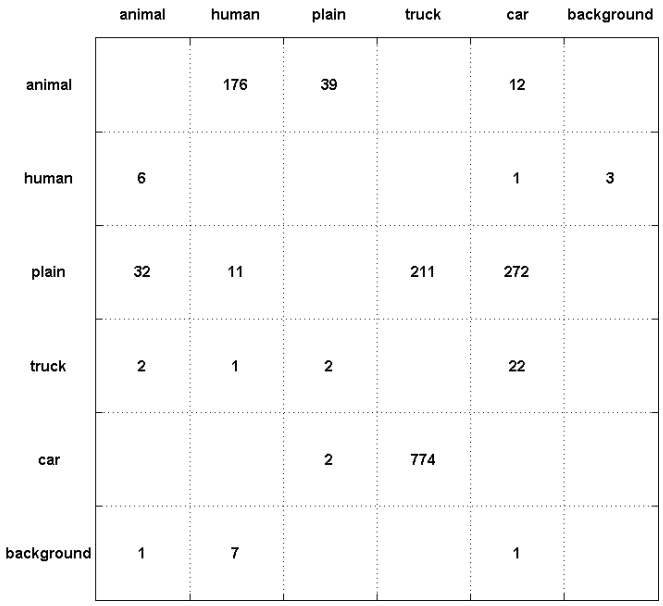

Figure 8. Confusion matrix for the NORB: correct labels on vertical axis; detected labels on horizontal axis.

References

[1] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010. 3

[2] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN'11)*, 2011. 3