Inductive Visual Localisation: Factorised Training for Superior Generalisation

Supplementary Material

Ankush Gupta
Andrea Vedaldi
Andrew Zisserman
{ankush,vedaldi,az}@robots.ox.ac.uk

Visual Geometry Group
Department of Engineering Science
University of Oxford

In section 1 we first give detailed architecture of the image-encoder Ψ used for text-recognition in multiple lines (section 4.1 in paper), and for visual object counting (section 4.2). Next, in section 2 we present results on a synthetic shapes dataset for the recognition task, similar to the one used counting; this was excluded from the paper due to lack of space.

1 Image Encoder Architecture (Ψ)

Our image encoder is based on the Dilated Residual Network [1]. We give details of the architecture of the encoder used for text-recognition and counting respectively.

1.1 Text Recognition Encoder

The image encoder is based on the DRN-C-26 network of [1]. The network if fully-convolutional, downsamples the input by a factor of 8, and has a stride of 32. The layer-level details are as following (top is first layer):

Conv−5×5−F16−D1
Res−3×3−F16−D1−S2
Res−3×3−F32−D1−S2
Res−3×3−F64−D1
Res−3×3−F64−D1−S2
Res−3×3−F128−D1
Res−3×3−F128−D1
Res−3×3−F256−D2
Res−3×3−F256−D2
Res−3×3−F512−D4
Res−3×3−F512−D4
Conv−3×3−F512−D2
Conv−3×3−F512−D2
Conv−3×3−F512−D1
Conv−3×3−F512−D1

Where,

- 'Conv' stands for a convolutional layer, with ReLU activation [2] and batch-normalisation [3]; 'Res' stands for the Pre-activation Residual Unit of He *et al*. [4].
- second term is dimensions of the the filters
- 'Fn' means n filters
- 'Dr' gives the dilation rate of the filters [5].
- if present, 'S2' means a filter stride of 2, otherwise the stride is 1.

1.2 Visual Object Counting Encoder

The image encoder employed for counting is much simpler, and employs six residual [4] layers. The image is not downsampled. Layer-wise details, using the naming scheme given above, are as following:

Res−5×5−F32−D1−S1
Res−5×5−F32−D1−S1
Res−5×5−F32−D2−S1
Res−5×5−F32−D2−S1
Res−5×5−F32−D4−S1
Res−5×5−F32−D4−S1

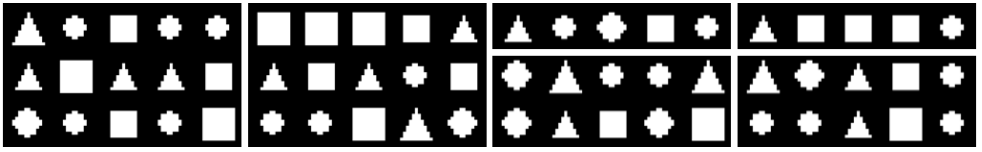
2 Recognising Multiple Lines of Text: Toy Example

We evaluate generation ability of sequence recognition models to multiple lines on a synthetic Shapes dataset similar to the Coloured Shapes dataset used for counting. We experimented with the toy task of recognising sequences of shapes organised in multiple lines in an image (see fig. 1). Two models are evaluated: first, our inductive block parser, which is trained at the line-level. The other, a conventional soft-attention encoder/decoder RNN trained at the block level (no factorisation in terms of lines).

Dataset. Binary valued images containing three types of shapes — square, triangle, and circle — each in two different sizes, and organised into lines, are synthetically generated. Each line consists of a sequence of five randomly sampled shapes. The training set consists of 2000 such images all containing *three* lines. The test set consists of 12 different subsets with varying number of lines — {1 to 10, 15, 20}, each containing 200 samples.

Evaluation. We evaluate the models on images containing differing number of lines {1–10,15,20}, to test for generalisation. We report the normalised edit-distance, computed as the total edit-distance between the predicted block-string, and the ground-truth block-string, normalised by the length of the ground-truth string.

Model Architecture. The two models, our inductive block parser and a conventional RNN model have the identical architectures: the image-encoder is a stack of six convolutional layers+ReLU (with 4×16, 2×32 filters, two 2×2 max-pooling after the second and the



↓ Model \ Lines→	1	2	3	4	5	6	7	8	9	10	15	20
end-to-end	241.80	54.59	0	28.41	42.66	52.44	59.17	64.20	68.17	71.28	80.9	85.71
inductive	0	0	0	0	0	0	0	0	0	0	0	0.02

Figure 1: (top) Samples with different number of text lines from the Toy Shapes test set. (bottom) Normalised edit-distance rates (%) for the task of recognizing shapes organised in blocks, comparing the generalisation capabilities of a conventional soft-attention RNN trained end-to-end, and with our inductive factorisation. The models were trained on blocks containing three lines, and tested for generalisation on a varying number of lines. Error rates of more than 100% are due to the model always predicting exactly three lines.

fourth-layers); the decoder is a soft-attention LSTM-RNN with 128 hidden units; attention embedding is also 128 dimensional. The memory updates $\Delta\mathbf{m}_t$ are regressed using two convolutional+ReLU layers (32 filters each).

Discussion. The results are shown in fig. 1. Both the models achieve perfect recognition accuracy on the test set containing three lines (the same number of lines as in the training set), but only the inductive line parser is able to generalise to different number of lines. The conventional RNN trained end-to-end to produce block-level predictions, always predicts three lines regardless of the number of lines in the test image.

References

[1] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. CVPR*, 2017.

[2] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, 2010.

[3] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*, 2016.

[5] F. Y. and V. K., "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016.