P. Bodesheim, A. Freytag, E. Rodner, and J. Denzler: Local Novelty Detection in Multi-class Recognition Problems. – Supplemental Material – , Proc. WACV 2015, (accepted for publication).

Local Novelty Detection in Multi-class Recognition Problems – Supplemental Material –

Paul Bodesheim, Alexander Freytag, Erik Rodner, Joachim Denzler
Computer Vision Group, Friedrich Schiller University Jena, Germany
`{firstname.lastname}@uni-jena.de | www.inf-cv.uni-jena.de`

Abstract

This document contains additional evaluations for the methods presented in the paper Local Novelty Detection in Multi-class Recognition Problems *at WACV 2015. However, it is not necessary to understand the main paper. We show further comparisons regarding computation times of the local KNFST models compared to the full KNFST baseline [5]. Please refer to the main paper for explanations of the different methods (Sect. 2.2 and 3) as well as the experimental setup and further results (Sect. 4).*

S1. Further evaluations of computation times

Additional results for the computation times of local KNFST models are shown in Table S1 and Figure S2. The experiments are performed with the ImageNet dataset [8] and measured runtimes are averaged over 20 different training sets and in the test phase additionally over the number of test samples (= twice the number of training samples). We have used a 64-bit machine with AMD Opteron processor and 2.8 GHz. It can be seen that average test times of the local KNFST models increase with an increasing size of the neighborhood (parameter k). However, they are still below 1 second for a single test sample with $k \leq 700$ (see Figure S2). Note that measured runtimes involve computing similarities using the kernel function during both training and testing. Therefore, also local models spend some time during training (see Table S1). Nevertheless, this amount of time is independent of k and also included in the training time of the full KNFST model.

Precomputing the kernel matrix containing pairwise similarities of all training samples can be omitted for the local models, leading to zero computation time during training. However, in this case the computation time during testing would increase since local models require the kernel matrix of the nearest neighbors. While precomputing the kernel matrix is done for all training samples resulting in runtimes listed in Table S1, computing similarities only for the nearest neighbors is faster and depends on the size of the neighborhood k. Computation times for calculating the kernel matrix with various choices of k are shown in Figure S1. We have also fit a second order polynomial since computing the kernel matrix takes time quadratic in the number of samples that are involved. Note that we have used the histogram intersection kernel [3] and bag-of-visual-words histograms of dimension $D = 1{,}000$ in the experiments. Considering a single test sample, the runtimes in Figure S1 have to be added to the test times in Table S1 and Figure S2 in case the kernel matrix has not been precomputed. When calculating novelty scores for a set of test samples, they are worst-case runtimes because similarities between the same neighbors can be shared among test samples and only need to be computed once. Thus, the total overhead of computing the kernel matrix of training samples in the test step is not larger than the training times of local models in Table S1 when considering the total test time for *all* test samples. Perhaps some training samples will never be chosen as neighbors and this upper bound is not reached for a specific test set. All in all, we recommend to precompute the kernel matrix in order to save these computational costs in the test phase.

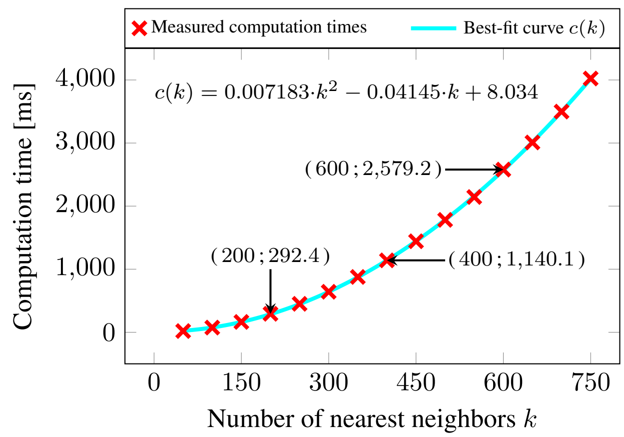

Figure S1. Computation time that is spent to calculate a $k \times k$ kernel matrix from k training samples that are determined as the nearest neighbors of a test sample. See text for further details.

	Full KNFST [5]	Local KNFST models (this paper) k=200	k=400	k=600
average train time (1,000 samples)	$11.0 \cdot 10^3$ ms	($7.5 \cdot 10^3$ ms)	($7.5 \cdot 10^3$ ms)	($7.5 \cdot 10^3$ ms)
average test time per sample	7.6 ms	50.8 ms	213.6 ms	632.1 ms
average train time (3,000 samples)	$137.8 \cdot 10^3$ ms	($70.8 \cdot 10^3$ ms)	($70.8 \cdot 10^3$ ms)	($70.8 \cdot 10^3$ ms)
average test time per sample	24.4 ms	65.5 ms	242.0 ms	665.0 ms
average train time (5,000 samples)	$548.6 \cdot 10^3$ ms	($196.2 \cdot 10^3$ ms)	($196.2 \cdot 10^3$ ms)	($196.2 \cdot 10^3$ ms)
average test time per sample	40.1 ms	80.9 ms	258.1 ms	676.7 ms

Table S1. Runtime comparison of selected local KNFST models and the full KNFST baseline [5]. Numbers in parentheses indicate time spent for computing the kernel matrix only, because local models have no training step and thus no additional computation costs. Training of local models is done in the test step leading to a larger amount of time that is needed during testing. If the kernel matrix for training samples is not precomputed, the train times of local models will be equal to zero but test times will increase depending on the number of neighbors retrieved (parameter k). However, test times would *not* increase by the train times mentioned in this table, because for each sample only the kernel matrix of the nearest neighbors has to be computed (see Figure S1 for those additional computation times).

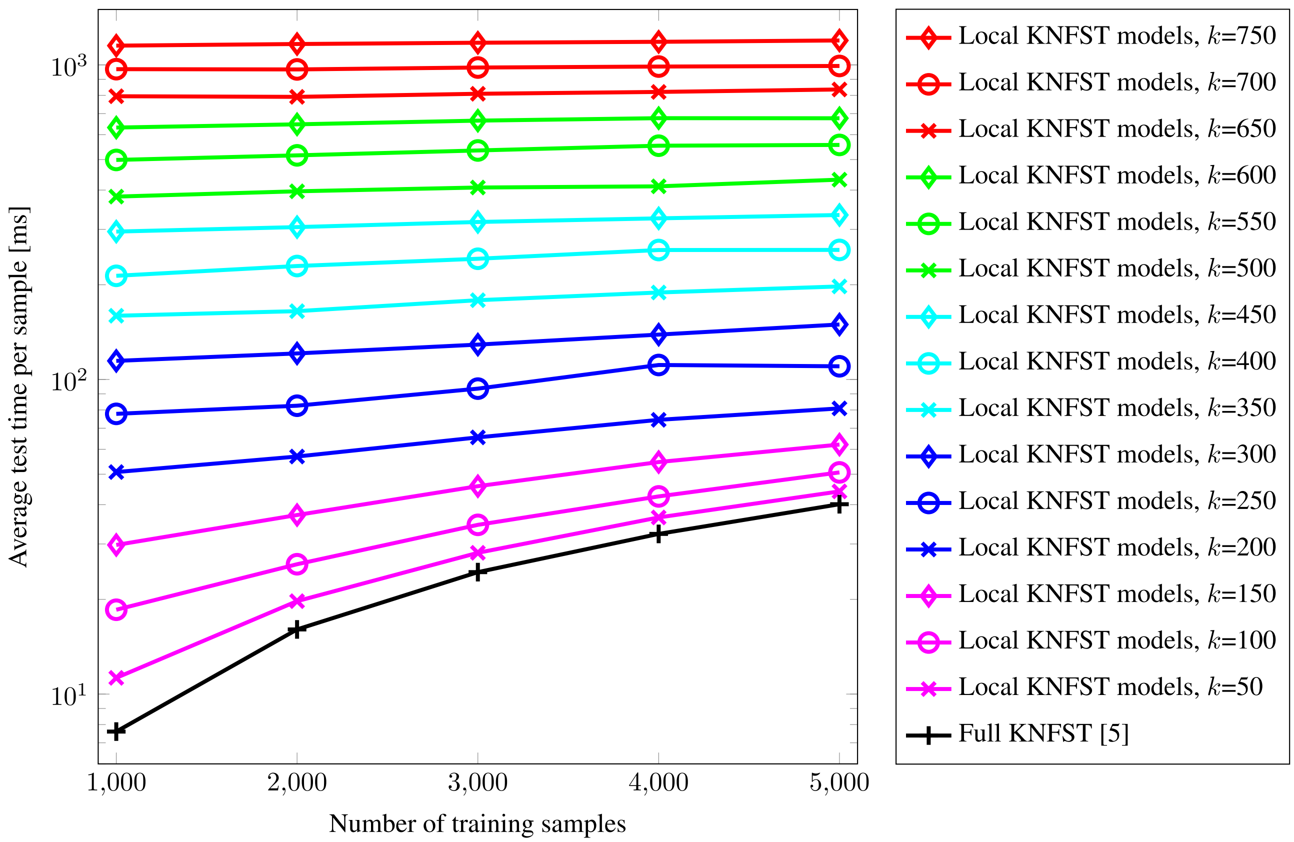

Figure S2. Test time of KNFST models depending on the number of training samples. We vary the size of the neighborhood and compare with the baseline of applying a full KNFST model [5]. Note the logarithmic scale of the y-axis.

References

[3] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *Proc. ICIP*, pages 513–516. IEEE, 2003.

[5] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler. Kernel null space methods for novelty detection. In *Proc. CVPR*, pages 3374–3381. IEEE, 2013.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255. IEEE, 2009.