De-noising and Representation Learning for Matching

Weijia Zhang
Imperial College London
wz4515@ic.ac.uk 01077024

Abstract

Many computer vision tasks require the extraction of local descriptors from images in the solution pipelines and very often the images are noise-corrupted. In this work, we investigate denoising algorithms followed by learning-based extractors for local feature descriptors, both based on Convolutional Neural Networks (CNNs). The performance of the pipeline is evaluated with rigorous metrics measuring the quality of the extracted descriptors on verification, matching and retrieval tasks.

1. Problem formulation

Pipeline The pipeline consists of patches de-noising followed by descriptors extracting from denoised patches, both implemented with deep learning approaches.

Denoising can be formulated as the minimisation of the distance d between a clean image and its noisy counterpart. The network takes as input a noise-corrupted image $x \in \mathbb{R}^{32\times32}$ and outputs its denoised version $y \in \mathbb{R}^{32\times32}$. The baseline uses a shallow version of U-Net[9] and the mean absolute error as the loss, that is the average of magnitude of pixel-wise difference between the clean and noisy images.

Descriptor extraction is formulated as the minimisation of distance between patches from the same class whilst maximising from different classes. Patches are represented by their learned descriptors $f \in \mathbb{R}^{128}$. f (also the output of the descriptor network). The baseline implements the L2-Net[10] with triplet loss[2]: in a set of three samples a, p, n where a, p, n denote anchor, positive and negative respectively, we define $\delta_+ = \mathbb{E}(f(a) - f(p)^2$ and $\delta_- = \mathbb{E}(f(a) - f(n))^2$ (i.e. the mean square error). The triplet loss is formulated as $L(\delta_+, \delta_-) = \max(0, \delta_+ - \delta_- + \alpha)$ where α is an arbitrary margin (1 by default).

Dataset: The N-HPatches dataset contains the HPatches dataset[1] and its noisy counterparts. We use split 'a' to divide the dataset into training set of 76 sequences and test set of 40 sequences (20 viewpoint and 20 illumination). The projected patches are perturbed with three levels of noise - EASY, HARD and TOUGH to simulate realistic settings.

Evaluation metrics: Given the unbalanced dataset, we use the mean average precision (mAP) to evaluate our descriptors on three complementary tasks: Patch Verification, Image Matching and Patch Retrieval[1]. The three metrics are further averaged to produce a final score.

2. Experimental results

We selected a subset of 10 training and 10 testing sequences to lower experimentation time. However, we will present results on the full dataset for the final report.

Optimiser Starting with SGD optimizer with baseline parameters, we fine-tune its learning rate in a systematic way in search for better performance. Following similar procedures we also experiment alternative optimisers (See Fig.1,2,3,6,9). In addition, we also change the loss function from *mae* to *mse* for the denoising network but did not achieve significant improvements in mAP metrics.

Training methodologies By varying the batch size we understood that the optimal batch size is optimiser-dependent and thus should be determined case-by-case. Varying the batch size (small to large) and learning rate (high to low) on the fly both improved results.

Architecture For the denoising part, we extend the shallow U-Net to a full architecture adapted from [9]. We showed that significantly improvements are obtained at the cost of longer training time(See Fig.5).

Batch Normalisation We introduce batch normalisaiton (BN), in both Shallow and full U-Net, for better convergence and generalisation[3] (See Fig.5).

3. Future Work

We will continue training and modifying the full U-NET(See Fig.8 for details). Other promising directions include residual networks which have proven effective in denoising tasks[11], and an alternative loss as in [6].We also propose to post-process the descriptors with ZCA whitening and normalisation as they empirically boost the results[1]. We also plan to implement data augmentation with synthetic noise for denoising networks, and with rotation, scaling and translation as in [1] for descriptor networks.

References

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors, 2017.

[2] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks, 2016.

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

[4] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima, 2016.

[5] D. Masters and C. Luschi. Revisiting small batch training for deep neural networks, 2018.

[6] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss, 2017.

[7] D. Mishkin, N. Sergievskiy, and J. Matas. Systematic evaluation of cnn advances on the imagenet, 2016.

[8] P. M. Radiuk. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets, 2017.

[9] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[10] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space, 2017.

[11] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, 2016.

4. Appendix

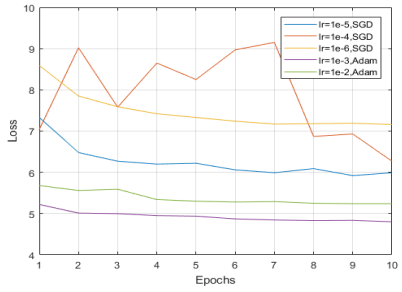


Figure 1: The learning curves of the shallow U-Net using SGD and Adam optimisers with different learning rates.

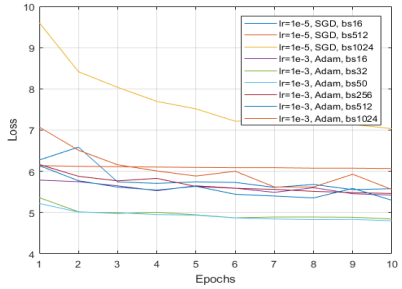


Figure 2: Learning curves of the shallow U-Net using SGD and Adam optimisers. Unless specified, all learning curves in this work plots the validation loss rather than the training loss. We experiment power-of-2 batch sizes as empirically they improve runtime efficiency and deliver better results[8]. Extremely large batch size such as 512 and 1024 degrades the network performance, which is supported by [5][4][7]. Similarly, overly small batch size also leads to poor results. Empirically the optimal batch size observed are **32** and **50** and Adam outperforms SGD.

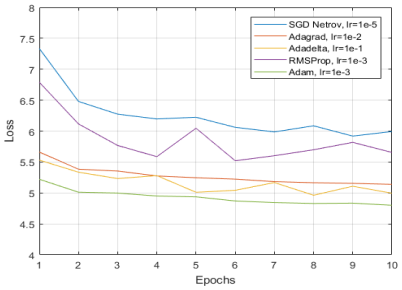
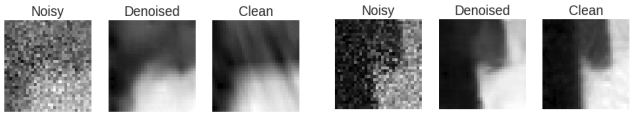


Figure 3: Learning curves of the baseline denoising network with different optimisers, trained with different batch sizes. The presented curves correspond to the optimal optimiser parameters determined empirically. Batch size is 50 for all experiments.



(a) (b)

Figure 4: Visualisation of noisy, denoised and the ground-truth clean patches. Left: denoised by Adadelta after 10 epochs (yellow curve in Fig.3). Right: denoised by Adam after 10 epochs (green curve in Fig.3).

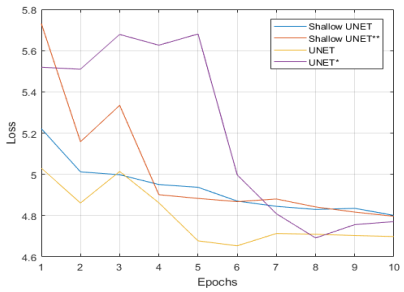


Figure 5: Learning curve of different network architectures under empirically optimal parameter settings. It is evident that the more complicated full U-Net architecture surpasses its shallow counterpart, while the addition of BN layers does not bring about convincing improvements. The validation performance can be potentially improved with dropout, which is not used in all shallow and full U-Net experiments.

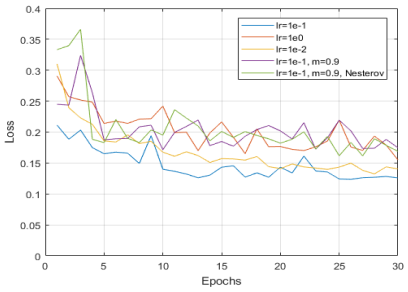


Figure 6: Learning curves of the baseline descriptor extraction network using variants of SGD. It is seen that **vanilla SGD** with learning rate **0.1** produces the best results followed by SGD with learning rate 0.01. Momentum and Nesterov did not optimise the learning process.

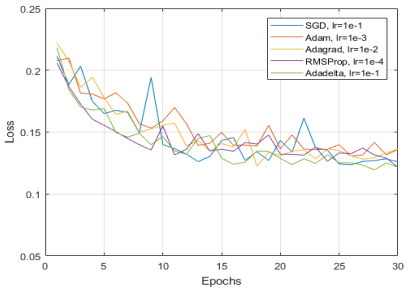


Figure 7: Learning curves of the baseline descriptor extraction network using different optimisers. The presented curves correspond to the optimal optimisers parameters determined empirically. We only show up to epoch 30 as beyond this saturation in validation loss starts to arise. Poor generalisation is observed in all of the 6 experiments where training loss got significantly lower than the validation loss, albeit the deceasing trends for both, motivating us to introduce more dropout in the L2-Net. According to [10], fixing the weighting and bias terms in BN layers to 1 and 0 respectively helped to resolve the issue of poor distribution of output feature maps and descriptors. However, in our experiment it resulted in worse generalisation and is therefore not presented. Moreover, [10] used Local Response Normalisation layer at the output to obtain normalised descriptors which improved results. Our implementation showed no tangible improvement and is is therefore not presented.

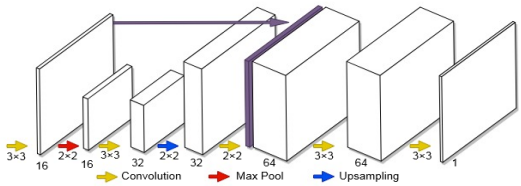


Figure 9: Architecture of the shallow U-NET used as the baseline denoising network. For shallow U-NET*, BN layers are added after each convolutional layer and before the activation function.

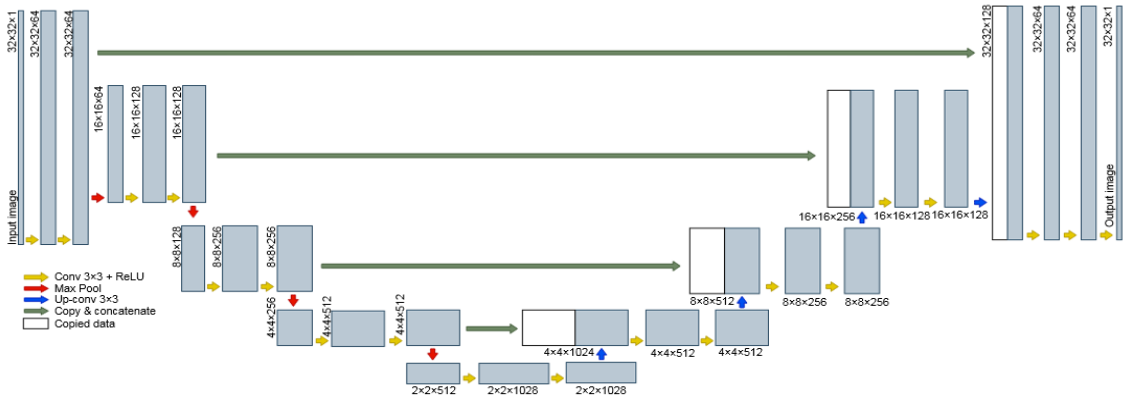


Figure 8: Architecture of the original U-NET. For U-NET*, BN layers are added after each convolutional layer and before the activation function. As the original U-NET architecture is used for input of dimension 572×572[9], we plan to reduce the number of down-sampling operations by at least 1 for our much smaller images (32×32) to see if it improves results. The intuition is that the smallest feature maps at the bottleneck link is only 2×2 in our case, and useful local features may not be effectively learned from such a small region. Furthermore, we plan to analyse the effect of batch size in training U-NET as whether a large or small[5][4][7] batch size leads to optimal results are largely heuristic and problem- and scenario-dependent. Also, as noted in Fig.5 BN in fact worsened the results. We plan to lower the momentum parameter from 0.99 to 0.6 in an attempt to stabilise the validation performance as pointed by other researchers.

Method	Architecture	Optimiser	#Epoch	others	Veri.	Match.	Retriev.	Avg.
Baseline	(Shallow u-net,L2 net)	(sgd,sgd)	(1,1)		0.695881	0.092932	0.338946	0.375920
Expt1	(Shallow u-net, L2 net)	(adam,sgd)	(10,1)		0.696606	0.114471	0.367083	0.392720
Expt2	(Shallow u-net, L2 net)	(adam,sgd)	(10,30)		0.813049	0.218617	0.521891	0.517852
Expt3	(Shallow u-net, L2 net)	(adam,sgd)	(10,30)	LRN	0.825039	0.230471	0.537963	0.531158
Expt4	(Shallow u-net, L2 net)	(adam,adam)	(10,30)		0.812408	0.210021	0.505527	0.509319
Expt5	(Shallow u-net, L2 net)	(adadelta,sgd)	(10,30)		0.821504	0.233744	0.530141	0.528463
Expt6	(Shallow u-net, L2 net)	(adadelta,sgd)	(10,50)		0.831328	0.247479	0.550802	0.543203
Expt7	(U-net, L2 net)	(adam,sgd)	(10,50)		**0.838425**	**0.249742**	**0.562377**	**0.550181**

Table 1: Method set-ups, mAP evaluation metrics on verification, matching and retrieval tasks, and their average as the final score for selected experiments. Elements in a bracket denote the set-ups for the denoising and the descriptor networks respectively.