Deep Lenaring Interim Report

Vasil Zlatanov (01120518)
vz215@ic.ac.uk

***Abstract*—This is my abstract**

I. Probelm Definion

This coursework's goal is to develop an image representation for measuring similarity between patches from the `HPatches` dataset. The `HPatches` dataset contains pacthes sampled from image sequences, where each sequence contains images of the same scenes. Patches are separeted into `i_X` patches which ahve undergone illumintation changes and `v_X` patches which have undergone viewpoint changes. For each image sequence there is a reference image with corresponding reference patches, and two more flie `eX.png` and `hX.png` containing corresponding pacthes from the images in the sequence with altered illumintation or viewpoint. Corresponding patches are extracted by adding geometric noise, easy `e_X` have a small amount of jitter while `h_X` patches have more[1]. The patches as processed by our networks are monochrome 32 by 32 images.

A. Tasks

The task is to train a network, which given a patch is able to produce a descriptor vector with a dimension of 128. The descriptors are evaluated based on there performance across three tasks:

- Retrieval: Use a given image's descriptor to find similar images in a large gallery
- Matching: Use a given image's descriptor to find similar in a small gallery with difficult distractors
- Verificaiton: Given two images, use the descriptors to determine their similarity

II. Baseline Model

The baseline model provided in the given IPython notebook, approaches the problem by using two networks for the task.

A. Shallow U-Net

A shallow version of the U-Net network is used to denoise the noisy patches. The shallow U-Net network has the same output size as the input size, , is fed a noisy image and has loss computed as the euclidean distance with a clean reference patch. This effectively teaches the U-Net autoencoder to perform a denoising operation on the input images.

Efficient training can performed with TPU acceleartion, a batch size of 4096 and the Adam optimizer with learning rate of 0.001 and is shown on figure V. Training and validation was performed with **all** available data. The network is able to achieve a mean average error of 5.3 after 19 epochs. With gradient descent we observed a loss of 5.5 after the same number of epochs. We do not observe evidence of overfitting with the shallow net, something which may be expected with a such a shallow network. An example of denoising as performed by the network is visible in figure V.

Quick experimentation with a deeper version of U-Net shows it is possible to achieve validation loss of below 5.0 after training for 10 epochs, and a equivalent to the shallow loss of 5.3 is achievable aftery only 3 epochs.

B. L2 Net

The network used to output the 128 dimension descritors is a L2-network with triplet loss as defined in CVPR 17 [2]. L2-Net was specifically for descriptor output of patches and is a very suitable choice for this task. L2-Net is robust architecture which has been developed with the HPatches dataset.

Training of the L2-Net can be done on the noisy images, but it is beneficial to use the denoise images from the U-Net to improve performance. Training the L2-Net with denoised yields training curves shown in

1) Triplet Loss: The loss used to train the siamese L2 Network: $ \mathcal{L] = max(d(a,p) - d(a,n) + margin, 0)$

There is an intrinsic problem that occurs when loss approaches 0, training becomes more difficult as we are throwing away loss data which prevents the network from progerssing significantly past that point. Solutions may involve increase the margin α or addopting a non linear loss which is able to avoid the loss truncation.

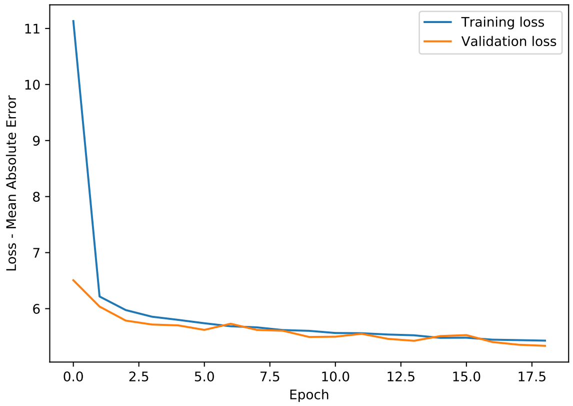
III. Peformance & Evaluation

Training speed was found to be greatly improvable by utilising Google's dedicated TPU and increasing batch size. With the increase in batch size, it becomes beneficial to increase learning rate. Particularly we found an increase of batch size to 4096 to allow an increase in learning rate of a factor of 10 over the baseline which offered around 10 training time speedup, together with faster convergence of the loss for the denosie U-Net.

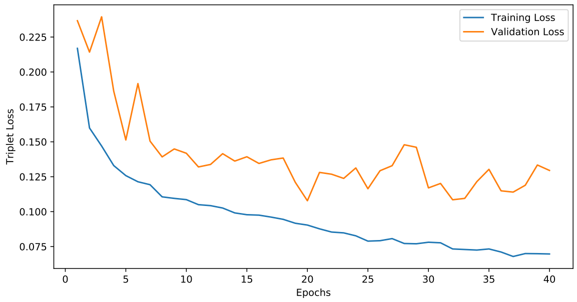
We evaluate the baseline accross the retrieval, matching and verification tasks:

IV. Planned Work

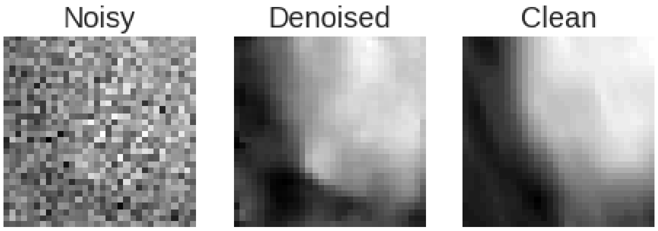
V. Appendix

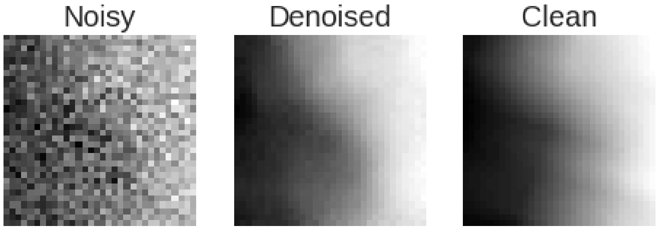


{}



{





{

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *CVPR*, 2017.

[2] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *The ieee conference on computer vision and pattern recognition (cvpr)*, 2017.