Noise2Noise

Matteo Calafà, Paolo Motta, Thomas Rimbot
First Project for the Deep Learning (EE-559) course at EPFL Lausanne, Switzerland

***Abstract*—In this first part of the project, we analyse a possible denoising model aiming to restore images just by looking at corrupted examples, hence the name *Noise2Noise*. The based network architecture adopted is a U-Net, in which skip connections between the contracting and expansive paths are used.**

I. Introduction

In many applications such as image denoising, image compression, and, in some cases, even image data generation, there is the necessity to explicitly model a high-dimensional signal. To this end, the use of an autoencoder allows a neural network to learn an efficient representation of unlabeled data by training the network to ignore signal "noise". In the case of image denoising, the main idea is therefore to capture a small number of degrees of freedom that represent the physical context, and from these perform an efficient reconstruction. The auto-encoder therefore has two main building blocks. The first is an encoder, whose objective is to learn a lower-dimensional representation (encoding) for a higher-dimensional data. This is typically used for dimensionality reduction, and is achieved by training the network to capture the most important parts of the input image. The second is a decoder, whose objective is instead to reconstruct the image from its lower-dimensional representation.

The idea of the auto-encoder underlies the structure of the U-Net. This structure introduces *skip connections* between encoding and decoding layers, concatenating the states. This allows the U-Nets to use fine-grained details learned in the encoder part to reconstruct an image in the decoder part. Although the structure of the U-Net is mainly used for image segmentation tasks, in this paper we would like to show how the same structure can be adapted for an image denoising task, as already shown in [5]. As also stated in this paper, most image denoising models are trained using large numbers of pairs $(\hat{x_i}, y_i)$ of noisy inputs $\hat{x_i}$ and clean reference images y_i, but here the aim is instead to show how satisfactory results can be obtained even without clean samples, if the noise is additive and unbiased.

II. Dataset And Data Augmentation

A. First Analysis of the Dataset

The dataset consists of 50'000 examples for the training set and 1000 examples for the test one. Each example is represented by a noisy pair of RGB images that are downsampled and pixelated. All images are 32×32 pixels in size. The networks proposed have the goal to reduce the effects of downsampling on unseen images. In Figure 1 we can observe two examples from the test set that we will reuse later in order to visually observe the results of our model.

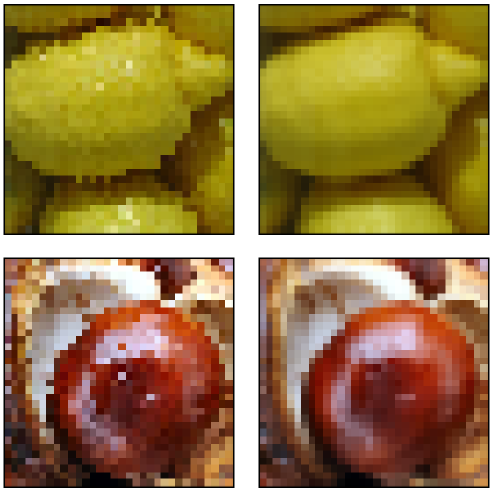

Fig. 1: Noisy starting image (left) and target image (right)

B. Data Augmentation

A data augmentation technique is used to extend the dataset by adding transformed copies of already existing images. In doing so, we can increase the generalizability and robustness of our model, avoiding overfitting as well. In particular, we decided to add a horizontally and a vertically flipped version to our dataset. The motivation for this choice lies in the fact that our model must be able to reconstruct the main features of the image regardless of its possible orientation.

Other transformations, both geometric and colour-based, can be performed on the images, but in most cases they do not bring any real improvement other than a deterioration in performance when colours are changed.

III. Models and Methods

A. U-Net Architecture

In the original structure of the U-Net [6], the input and output images have different sizes. To be able to evaluate denoising quality with the same size, we reworked the model by changing the parameters of the convolution function. As mentioned earlier, the U-Net network can be divided into two parts. The first is the contracting path which uses a typical CNN architecture. Each block in the contracting path consists of two successive 3×3 *convolutions with* *padding* followed by a *Leaky ReLU* activation unit and a *downsampling* layer. However, despite the fact that the downsampling layer is typically implemented through a max-pooling operation, we decided to replace this layer with a convolution with a larger stride. Indeed, max-pooling (or any kind of pooling) is a fixed operation and replacing it with a strided convolution can also be seen as learning the pooling operation, which increases the model's expressiveness ability [8]. This structure is repeated several times. The main characteristic of U-Net comes in the second part, called the expansive path, in which each stage upsamples the feature map using 3×3 *transposed convolution* with a larger stride. Then, the feature map from the corresponding layer in the contracting path is *concatenated* onto the upsampled feature map. This is followed by two successive 3×3 convolutions with padding and a Leaky ReLU activation. At the final stage, an additional sequence of convolutions 3×3 is applied to reduce the feature map to the required number of channels and produce the denoised image. The overall result is a network with a U-shape and, more importantly, it propagates contextual information along the network, which allows it to properly reconstruct the context. Figure 4 illustrates the overall U-net architecture.

B. Residual U-Net Architecture

Along with the U-Net presented above, we also decided to implement a different structure of our network, combining the benefits of the U-Net structure with that of a Res-Net [2]. Instead of directly predicting the denoised image, the model predicts the residual noise of the corrupted image. Such a structure also makes it possible to use a network with many more layers without running into the vanishing gradient problem. The resulting final structure is called *Residual U-Net*, and although this architecture is also mainly used for image segmentation [9], we decided to adapt it for our denoising task.

The structure of the network is similar to that described for the U-Net, with the main differences being that each block is implemented as a residual block. Since within the residual block the input must be added to the output of the block, if the two quantities have discordant channel sizes, a 1×1 convolution is adopted to scale the number of input channels. Furthermore, it has been proven in the literature that *Batch Normalization* makes the training process smoother. However, it requires a sufficiently large batch size, and our choice of $\texttt{batch\_size} = 30$ may be too restrictive. For this reason, we also tried to make use of *Group Normalization* which, unlike batch normalization, does not require a very large number of batches, as it divides the channels into groups and normalizes the features within each group.

After experimenting without normalization, with batch normalization and with group normalization, we observed that the latter yielded much better results and therefore decided to adopt it for our final model. Figure 2 illustrates the overall Residual U-Net architecture.

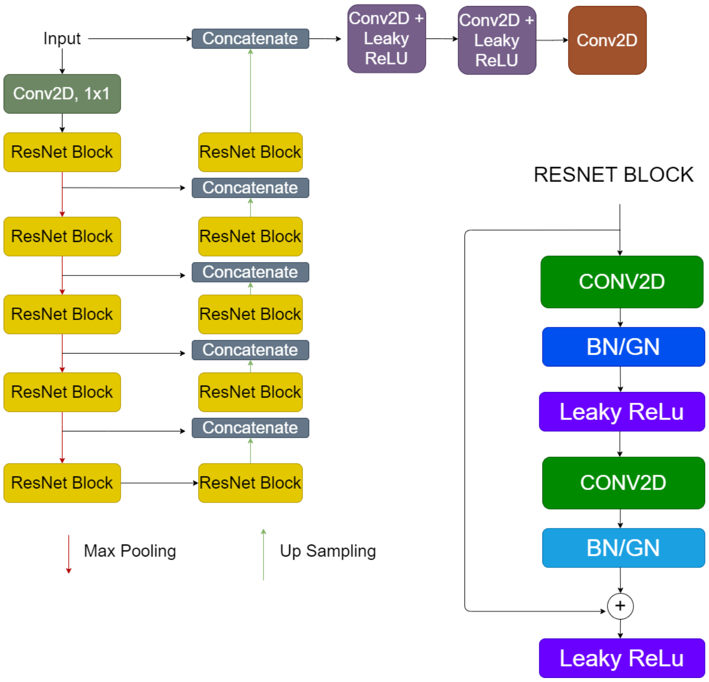

Fig. 2: Residual U-Net Architecture

C. Weight Initialization

In order to prevent layer activation outputs from exploding or vanishing gradients during training, an appropriate initialization of the weights is necessary and allows to achieve better performance. Among the most widely used are the Xavier normalization and He-et-al normalization when dealing with convolutional layers [3]. We experimented with both inizializations for our networks and the former seems to lead to slightly better performances than the latter. We therefore sticked to this one in both models' implementations.

IV. Results

In this section we report the results obtained with our best model given by the U-Net architecture, together with the final performance of the Residual U-Net. The model's performance is calculated through the PSNR between our prediction (denoised image I) and the target image T, given by:

$$\mathrm{PSNR} := 20 * \log_{10}\left(\frac{\mathrm{MAX\{I\}}}{\sqrt{\mathrm{MSE(I,T)}}}\right)$$

where $\mathrm{MAX\{I\}}$ is the maximum possible pixel value of the image and $\mathrm{MSE(I,T)}$ is the mean square error between the images:

$$\mathrm{MSE(I,T)} := \frac{1}{\mathrm{MN}}\sum_{i}^{M-1}\sum_{j}^{N-1}||I(i,j) - T(i,j)||^2$$

Figure 3 illustrates the trend of the training loss and the PSNR on the test set during training, using 30 epochs.

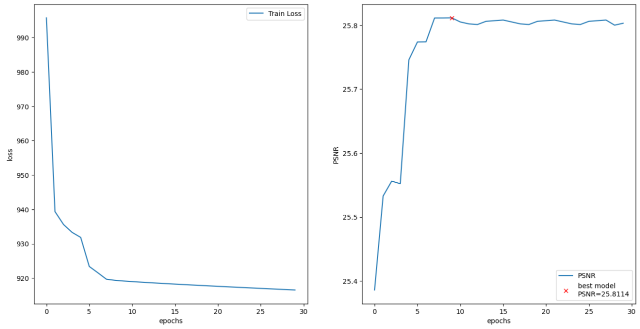

Fig. 3: Evolution of the loss (left) and PSNR (right)

The PSNR obtained on the test set with the best models are summarized in the following table.

Network Architecture	PSNR
U-Net	25.811
Residual U-Net	25.803

Finally, we report in Figure 5 the results of our best model on the two pairs of test images previously presented.

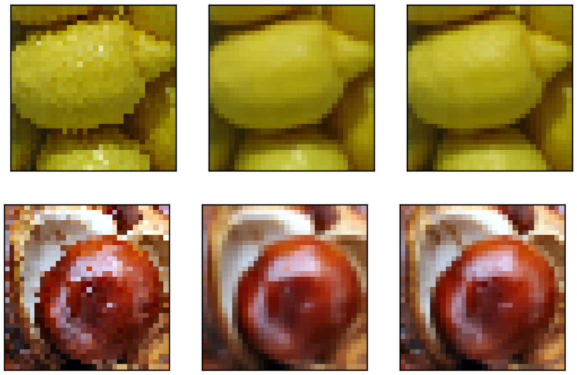

Fig. 5: Noisy input image (left), denoised output image (center) and target image (right)

V. Conclusion

The obtained results confirm that the architecture of a U-Net can also be adapted for a denoising task. However, using the network on images with such a small resolution (32×32) does not allow to fully exploit its potential. The structure of the network also allows its use on larger images, on which its effectiveness could be tested.

Finally, several variants have been built on the same basic architecture and can therefore be subsequently used and tested within this task [7].

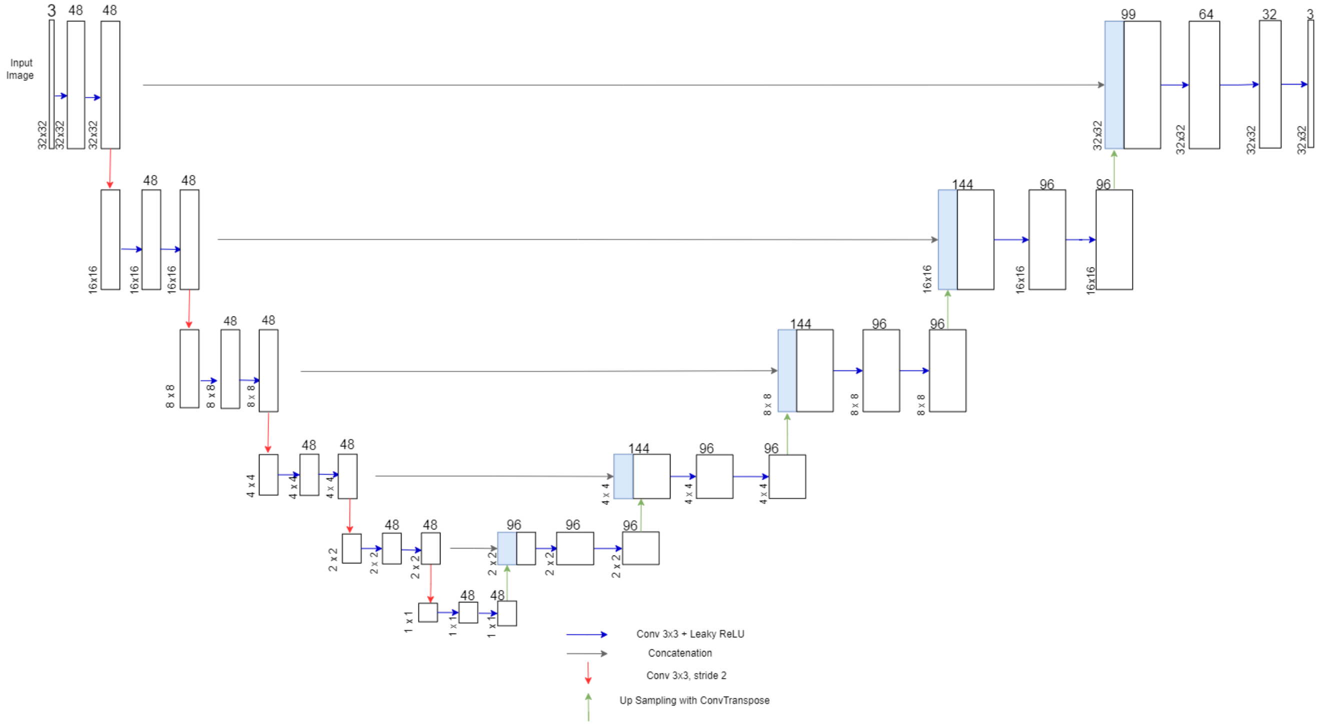

Fig. 4: U-Net Architecture

References

[1] Foivos I. Diakogiannis et al. "ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data". In: *CoRR* abs/1904.00592 (2019). arXiv: 1904.00592. URL: http://arxiv.org/abs/1904.00592.

[2] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[3] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *CoRR* abs/1502.01852 (2015). arXiv: 1502.01852. URL: http://arxiv.org/abs/1502.01852.

[4] Rina Komatsu and Tad Gonsalves. "Comparing U-Net Based Models for Denoising Color Images". In: *AI* 1.4 (2020), pp. 465–486. ISSN: 2673-2688. DOI: 10.3390/ai1040029. URL: https://www.mdpi.com/2673-2688/1/4/29.

[5] Jaakko Lehtinen et al. "Noise2Noise: Learning Image Restoration without Clean Data". In: *CoRR* abs/1803.04189 (2018). arXiv: 1803.04189. URL: http://arxiv.org/abs/1803.04189.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[7] Nahian Siddique et al. "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications". In: *IEEE Access* 9 (2021), pp. 82031–82057. DOI: 10.1109/ACCESS.2021.3086020.

[8] Jost Tobias Springenberg et al. "Striving for Simplicity: The All Convolutional Net". In: *CoRR* abs/1412.6806 (2015).

[9] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. "Road Extraction by Deep Residual U-Net". In: *CoRR* abs/1711.10684 (2017). arXiv: 1711.10684. URL: http://arxiv.org/abs/1711.10684.