0.1 GANs proof of concept

To test the adversarial setting versus a more traditional loss: Mean Square Error (MSE) (equation 0.1), a "toy example" is carried out[1]: train a generator model to fit a normal distribution with mean 0 and variance 1 (p_{data}).

The generator samples a vector of length 5 from a uniform ditribution (range -0.5-0.5) (p_z) and passes it through 3 fully connected layers[2] with LeakyReLU activation function and 5 units on each layer except for the final layer which only has 1 unit (it outputs a scalar) and no activation function (represented in figure 0.1).

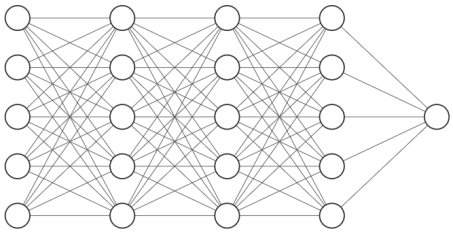


Figure 0.1: Generator architecture used for both the MSE and Generative Adversarial Networks (GANs) setting

The target data comes from a sample of 100,000 i.i.d normal random variables. The models are trained for 100 epochs with a batch size of 100 using the Adam optimization algorithm presented in section **??** with a learning rate of 5×10^{-4} and default momentum values.

The results are compared by computing the Kolmogorov-Smirnov test of normality[3].

0.1.1 Results for MSE loss

As to be expected, the Artificial Neural Networks (ANNs) simply ignores the source of randomness (set first layer's weights close to zero) and produces an almost deterministic output close to the distribution's expected value: this is in fact the optimal solution for

[1] Code available at `https://github.com/gio8tisu/GAN-PoC`

[2] A fully connected layer outputs the matrix product between its input and a weights matrix and adds a bias vector, a node/neuron represents a row in the weight matrix plus the corresponding element in the bias vector: $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$, where $\mathbf{x}$ is the input represented as a column vector.

[3] The test statistic provides a measurement of the divergence of your sample distribution from the normal distribution. The higher the value of D, the less probable the data is normally distributed. The p-value quantifies this probability, with a low probability indicating that the sample diverges from a normal distribution to an extent unlikely to arise merely by chance. It is computed using the `https://www.socscistatistics.com/tests/kolmogorov` online tool.

the MSE objective function.

$$L(G_\theta(\mathbf{x}), \mathbf{y}) = \|\mathbf{y} - G_\theta(\mathbf{x})\|_2^2 \tag{0.1}$$

Example of generated samples: `-0.05046 -0.05155 -0.05082 -0.05044 ...`

The value of the K-S test statistic for 300 samples generated by this model is 0.1023. The p-value is 0.01819. This provides good evidence that the data is not normally distributed.

0.1.2 Results for adversarial loss

Rather than training G to minimize $\log(1-D(G(\mathbf{z})))$, G is trained to maximize $\log D(G(\mathbf{z}))$. This is a common practice as equation (**??**) may not provide sufficient gradient for G to learn well early in training. It follows the same principle: try to fool the disctiminator.

$$L(G_\theta(\mathbf{x}), \mathbf{y}) = -\log D(G(\mathbf{z})) \tag{0.2}$$

The discriminator network is composed of 3 hidden layers with LeakyReLU as well, with a sigmoid activation function in the output layer.

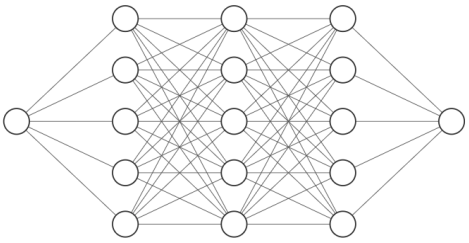


Figure 0.2: Discriminator architecture

In the case of the generator network traned in an adversarial manner, it produces samples with variability that are close to normally disributed (see figure 0.3 for a visual comparison with a true normal distribution).

The value of the K-S test statistic for 300 samples generated by this model is 0.03383. The p-value is 0.87045. This provides good evidence that the data does not differ significantly from that which is normally distributed.

This model clearly beats the MSE one on producing realistic and varied samples, but the training was much more unstable and the problem of mode collapse was encountered. Several training iterations with different hyperparameters (like number of layers, activation functions, learning rate,...) were necessary.

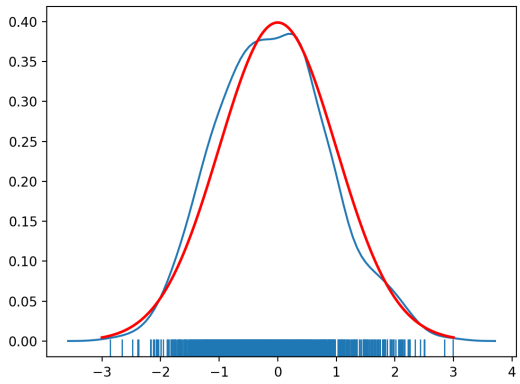


Figure 0.3: In blue a plot of the kernel density estimation using 1000 samples from the generator trained through the GANs framework. In red the probability density function of a normal random variable.

0.2 Despeckling network

The loss function used to define the objetive function the model will learn on is the MSE between Fluorescence Confocal Microscopy (FCM) 256×256 crops and the artificially contaminated version of it.

$$L(f_\theta(\mathbf{x}_{noisy}), \mathbf{x}_{clean}) = \|\mathbf{x}_{clean} - f_\theta(\mathbf{x}_{noisy})\|_2^2 \tag{0.3a}$$

$$\mathbf{x}_{noisy} = \mathbf{x}_{clean} \odot \mathbf{s} \tag{0.3b}$$

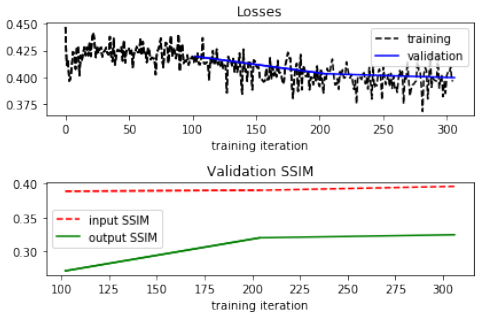
Where $\mathbf{s}$ is a realization of the speckle noise random variable model.

0.2.1 Model selection

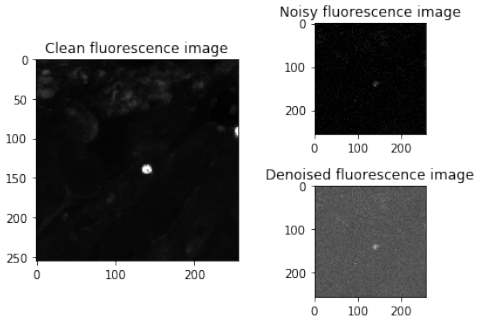
As explained in section **??**, three different network architectures are implemented. To evaluate the model performance, a comparison of the Structural SIMilarity (SSIM) (**Wang2004**) between noisy and clean (denoted $SSIM_{input}$); and denoised and clean (denoted $SSIM_{output}$) is used. Due to the lack of information about the confocal microcope used, the experiments are done with 2 different values $\{1, 5\}$ for the parameter L of the noise model (**??**).

The division skip-connection model is discarded early in the process because it fails to find any solution close to the desired.

In table 1 the mean SSIM values on the validation set for each of the final models are shown. The training process is done during 20 epochs using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 32 samples.



(a) Learning curve or the division skip-connection model. $SSIM_{input}$ is greater than $SSIM_{output}$ throughout the trianing process



(b) Example of denoised FCM

Figure 0.4: The division skip-connection model fails to converge.

Model	M	K	N	mean $SSIM_{output}$
Multiply ($L = 1$)	3	32	5	0.877
Multiply ($L = 1$)	5	64	5	0.728
Multiply ($L = 5$)	5	64	5	0.947
Log-Add ($L = 1$)	3	32	5	0.960
Log-Add ($L = 1$)	5	64	5	0.806
Log-Add ($L = 5$)	5	64	5	0.965

Table 1: Models comparison with different number of layers M, number of filters K and filter size N. The validation set mean $SSIM_{input}$ is 0.414 for $L = 1$ and 0.723 for $L = 5$

0.3 Staining network

0.3.1 Baseline

The baseline presented in **??** is trained using the Adam optimizer with a batch size of 8, learning rate of 5×10^{-4}, momentums of 0.5 and 0.999 for β_1 and β_2 respectively. This hyperparameter values provide the best looking results, as with other parameters the training quickly destabilizes.

To provide an initialization closer to the optimal one, the weights are initializated with the weights defined by **Gareau2009** transformation.

The final results are very similar to the ones of the initial state, (see figure 0.5) so the model makes no real progress to make a better looking stain version.

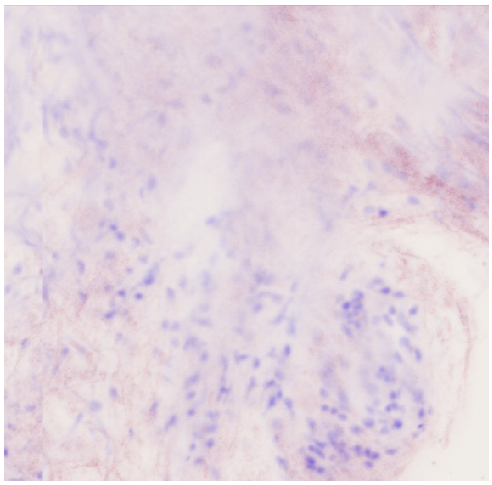


Figure 0.5: Example of a crop stained by the baseline model

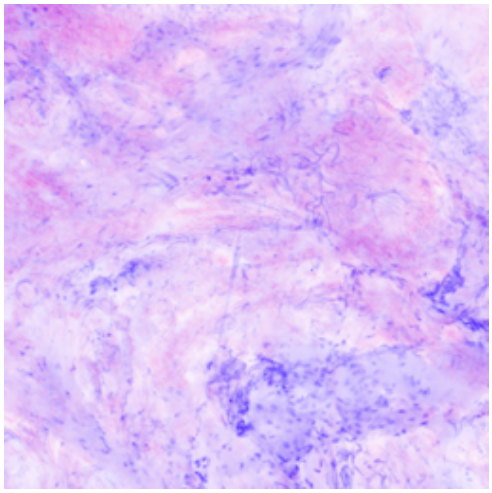


Figure 0.6

hyperparameter	value
λ_{cycle}	10
$\lambda_{identity}$	5
learning rate	2^{-4}
β_1	0.5
β_2	0.9
epochs	200
#layers D	3
residual blocks (residual model)	9
# down-sampling layers (UNet model)	7

Table 2: Table of hyperparameters used in staining network

0.3.2 Advanced Deep Neural Network (DNN)

The two architectures are trained using the hyperparameters described in table 2 for tand instead of optimizing the original GAN objective the so-called LSGAN —use the L2 distance instead of the cross-entropy— is used which is empirically shown to provide a more stable training (**lsgan**).

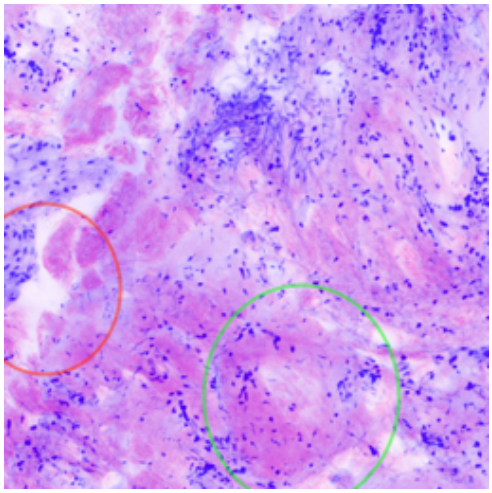
In theory the UNet-like model shoud mantain the structure and be less prone to "hallucinate", in practice this generally holds but still some structures are made up by the model; the residual model on the other hand sometimes eliminates nuclei present in the source image. Both cases can be seen in figure 0.7.

The mean value of the Local Binary Patterns (LBP) histogram distance for the validation set is 0.0332 for the residual model and 0.0183 for the unet.

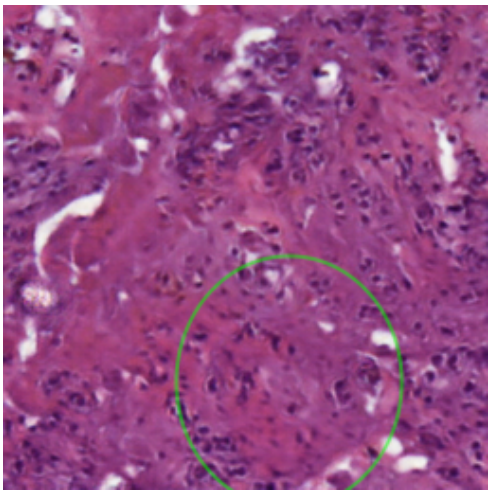
0.4 Inference method

The 2 of th inference methods described in section **??** are compared by computing the mean chi-squared distance of 512×512 patches in 7 large slides, i.e.: from a Confocal Microscopy (CM) large mosaic, the digitally stained version by **Gareau2009** is computed (DSCM), the selected inference method is used to produce the Hematoxylin and Eosin stain (H&E) version using the U-Net model and the DSCM as input, then these are divided with a grid of 512×512 cells and the LBP histogram is computed on each cell, once the histograms are computed the chi-squared distance is used to measure how similar each cell is.

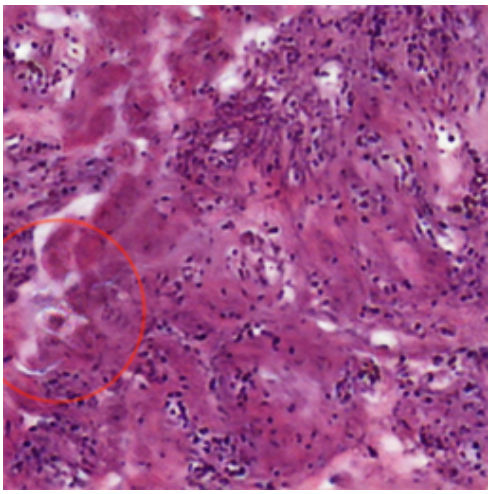
The plot in figure 0.8 shows how the method using 50% overlap and the weight metrix defined in 0.6 has a lower distribution of the metric for 7 different samples.



(a) Real



(b) Residual network



(c) UNet-like

Figure 0.7: (a) is the models' input, the second row show the outputs for the respective models. On (b) some nuclei that are in the input have been erased by the network —marked in green—. On (c) some structures are generated that are not present in the input —marked in red—.

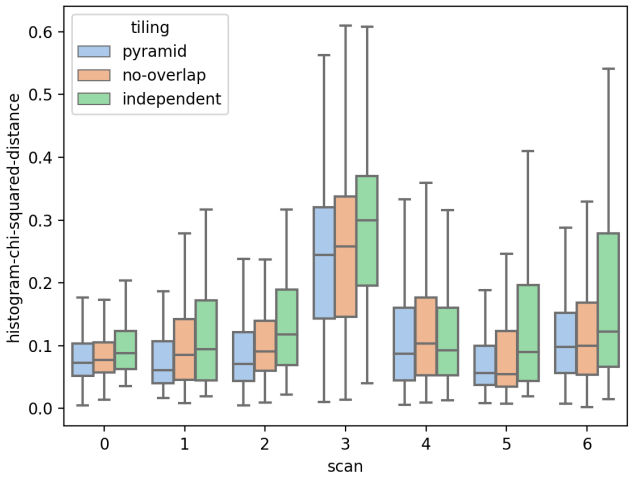


Figure 0.8: Independent tiling refers to inference made tile-by-tile with no overlap whatsoever. No-overlap refers to the technique where the inputs have overlap but the output is cropped so that they do not overlap. Pyramid is the WSI inference technique from **Bel2019**.