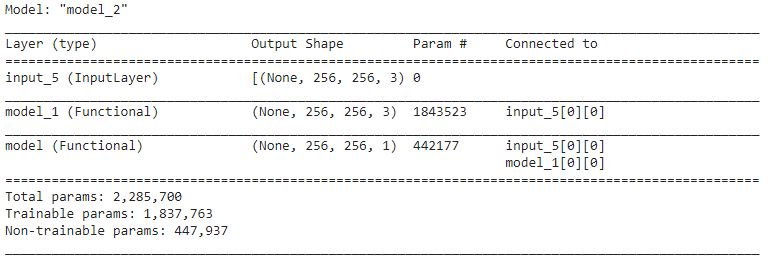
# Supplementary Information

## Technical parameters defined for U-Net & O-Net

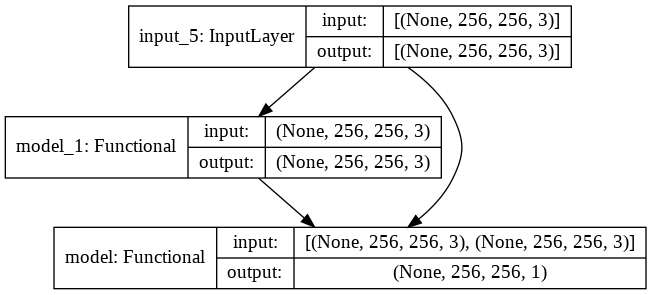
The architectures for both the U-Net & O-Net GANs, as well as the various parameters used for different aspects of these frameworks are illustrated in the diagrams below:

### U-Net Framework

The U-Net Pix2Pix GAN is depicted in the following diagram:

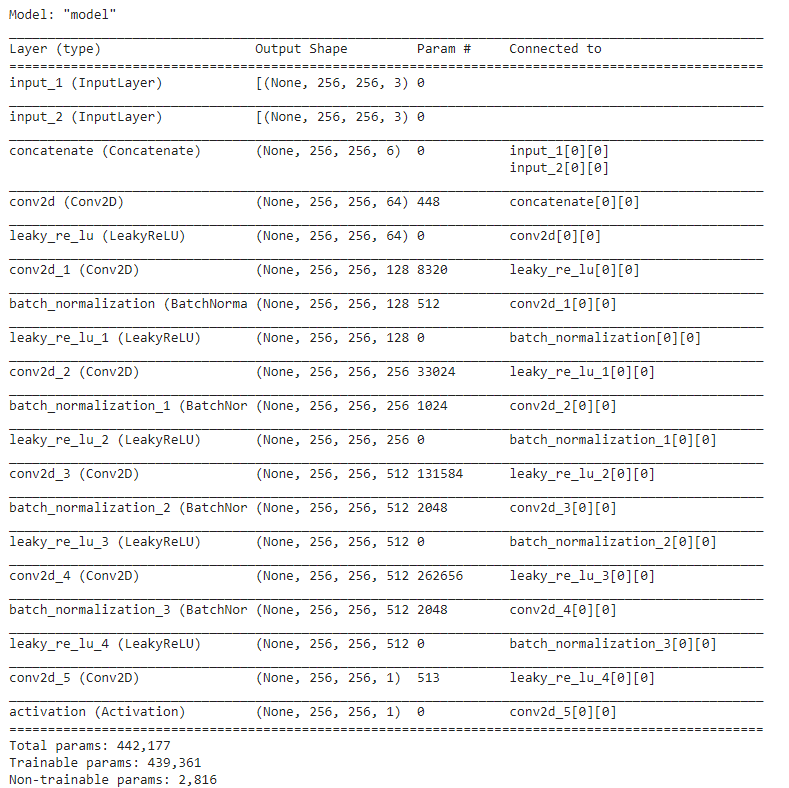


⮉ **Supplementary Table 1**: Model parameters for the U-Net GAN.

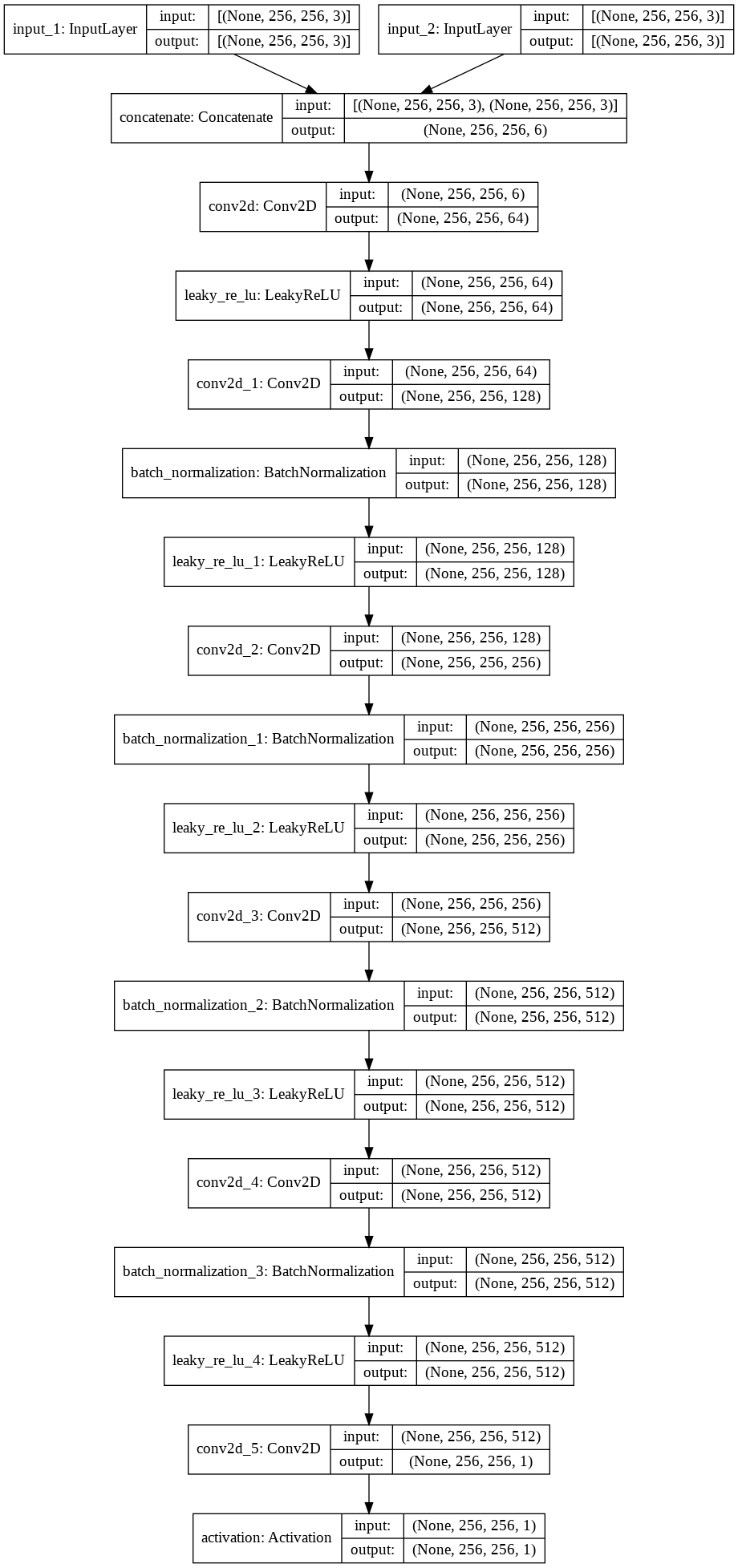


⮉ **Supplementary Figure 1**: Diagram illustrating the model parameters for the U-Net GAN outlined in **Supplementary Table 1** above.

A schematic of the discriminator architecture is provided as follows:

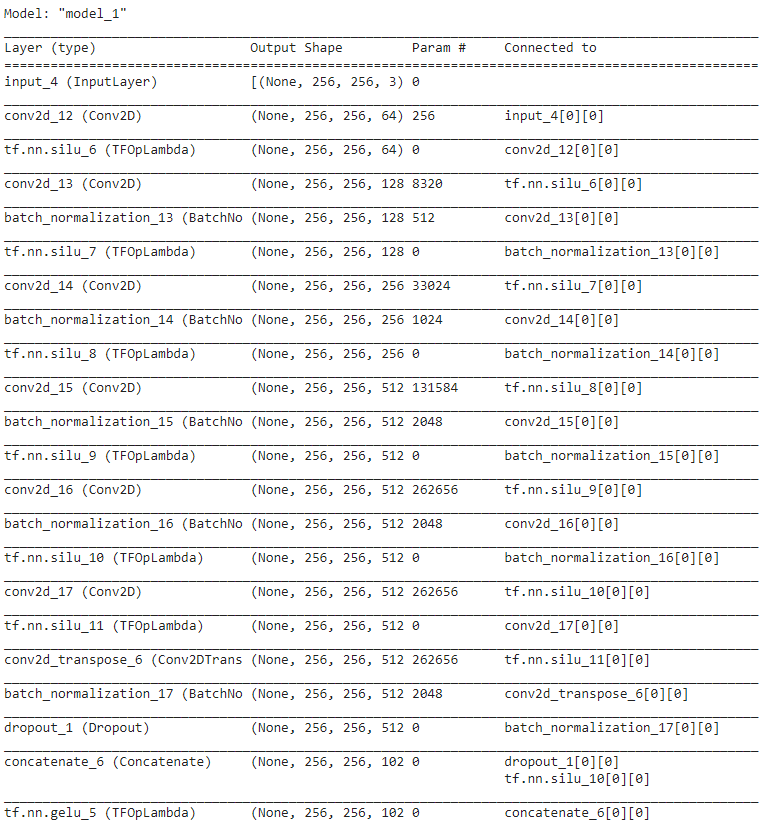


⮉ **Supplementary Table 2**: Model parameters for the discriminator of the U-Net GAN.

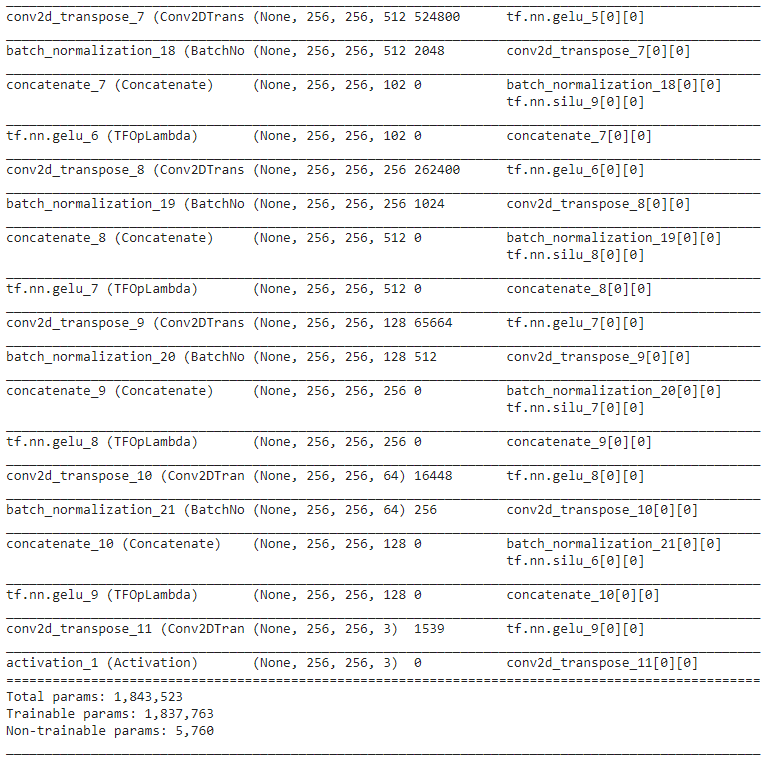


⮉ **Supplementary Figure 2**: Diagram illustrating the model parameters for the U-Net GAN discriminator outlined in **Supplementary Table 2** above.

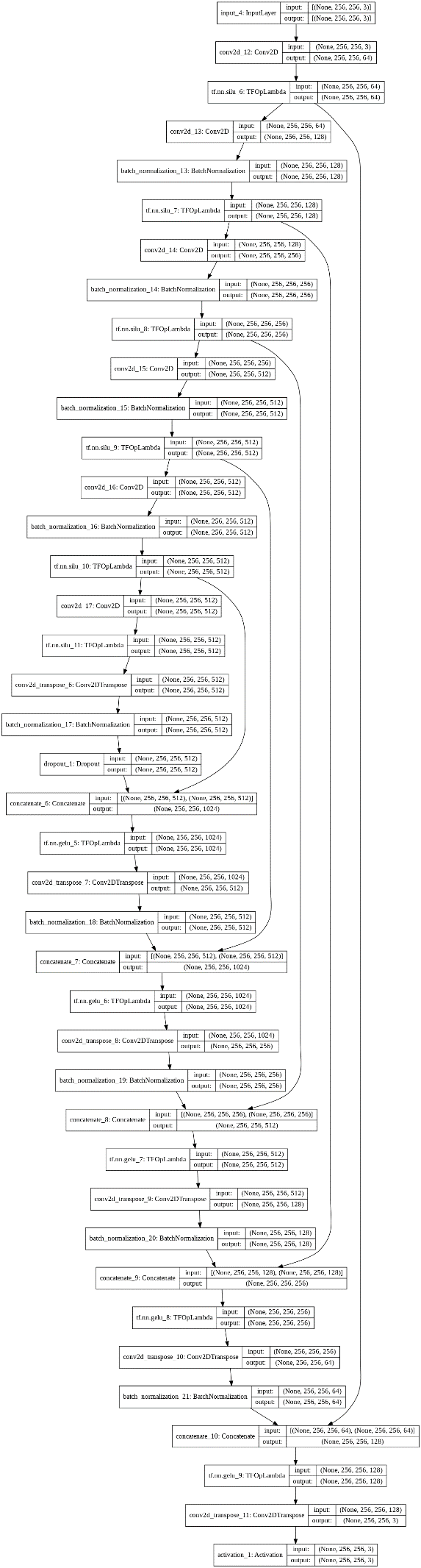
The generator architecture (employing U-Net) is also indicated below:



------------ Continued on the following page ------------------------



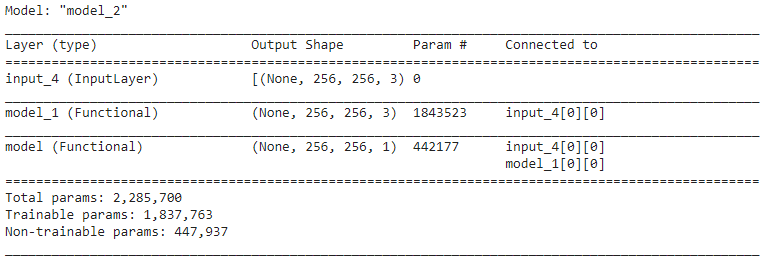
⮉ **Supplementary Table 3**: Model parameters for the generator of the U-Net GAN.



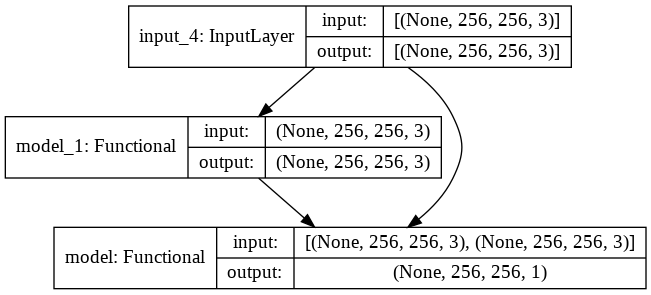
⮉ **Supplementary Figure 3**: Diagram illustrating the model parameters for the U-Net GAN generator outlined in **Supplementary Table 3** above.

### O-Net Framework

The O-Net Pix2Pix GAN is as described in the subsequent diagram:

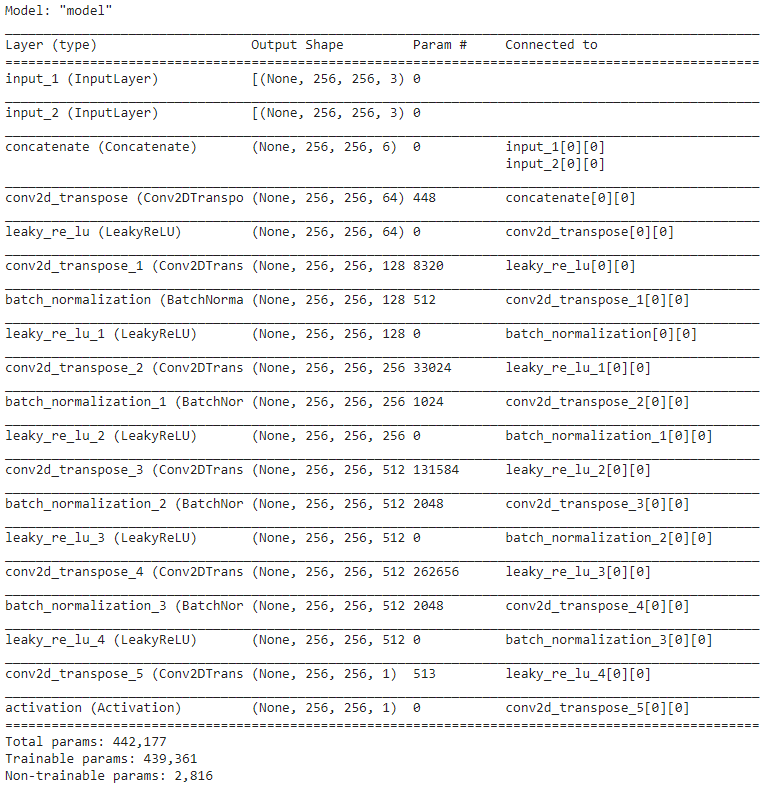


⮉ **Supplementary Table 4**: Model parameters for the O-Net GAN.

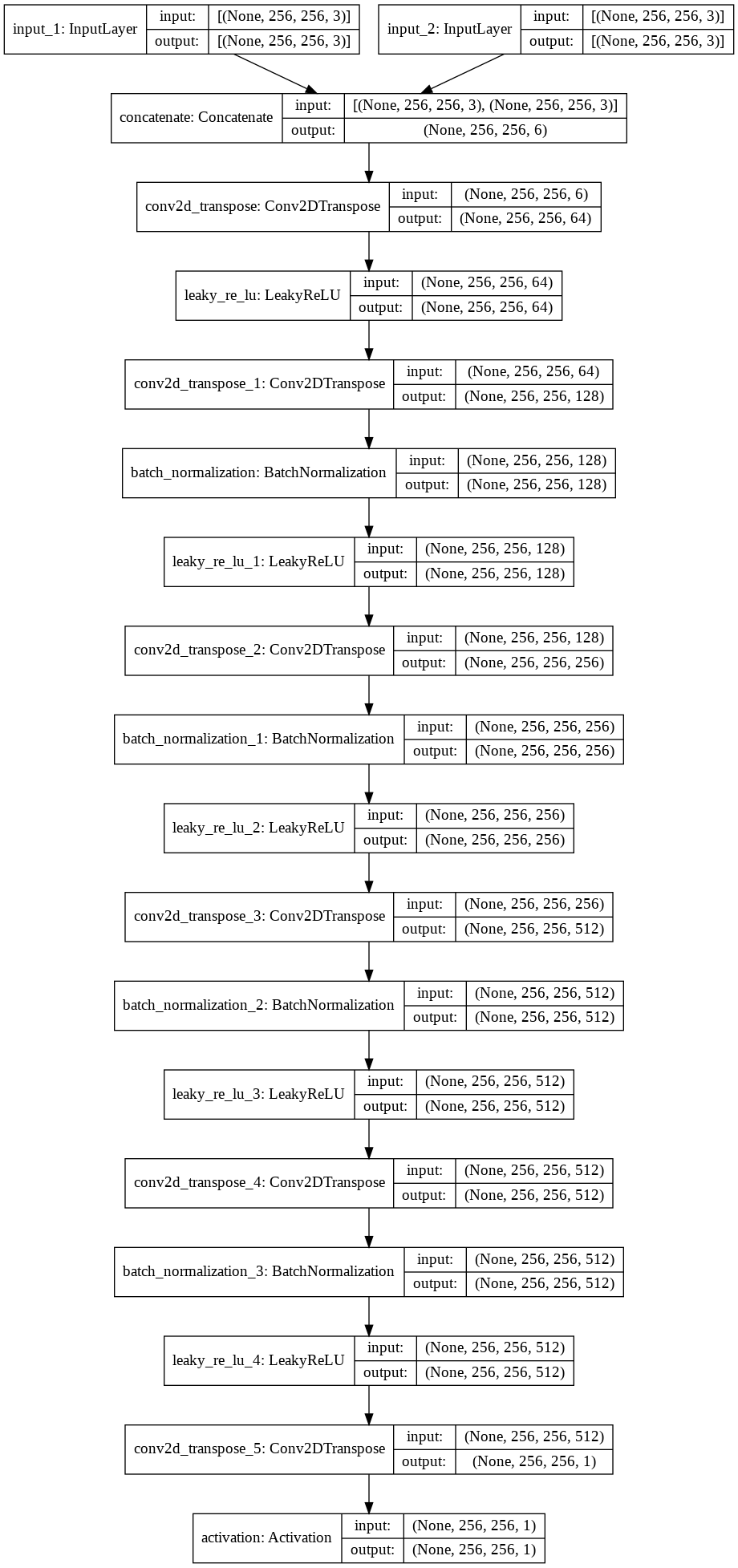


⮉ **Supplementary Figure 4**: Diagram illustrating the model parameters for the O-Net GAN outlined in **Supplementary Table 4** above.

As previously, the discriminator schematic is shown for reference:

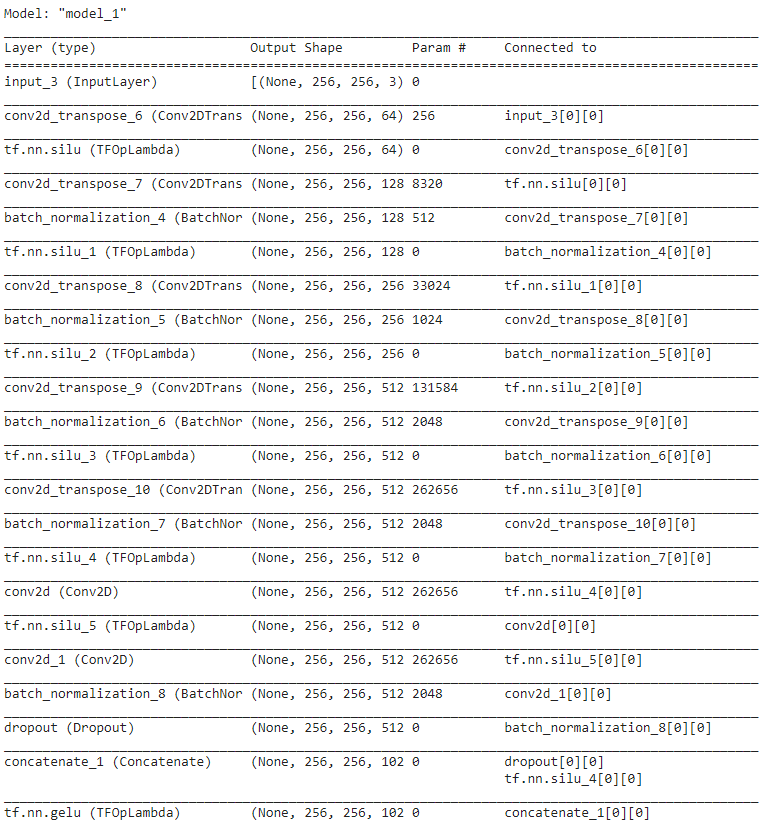


⮉ **Supplementary Table 5**: Model parameters for the discriminator of the O-Net GAN.

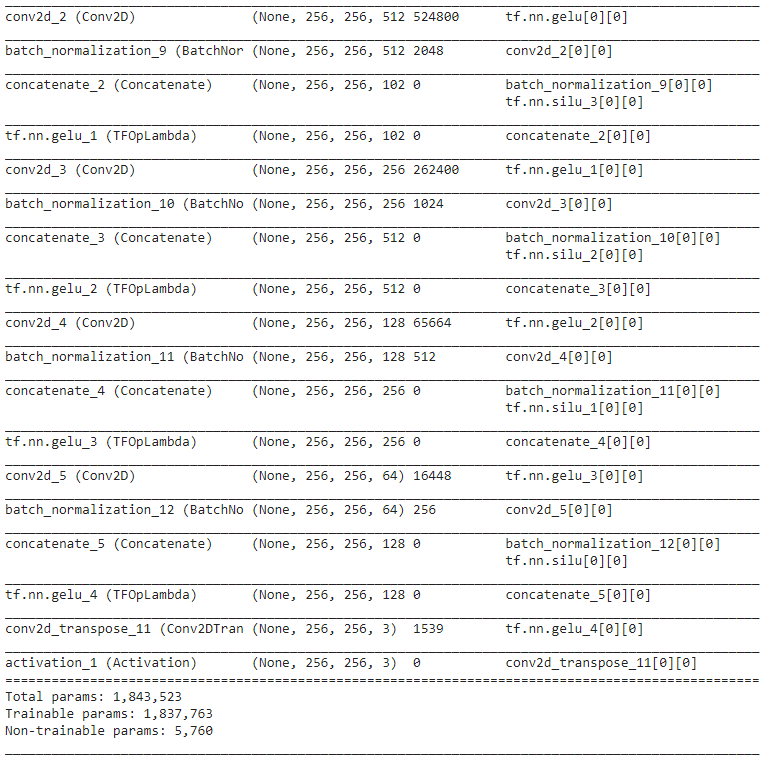


⮉ **Supplementary Figure 5**: Diagram illustrating the model parameters for the O-Net GAN discriminator outlined in **Supplementary Table 5** above.

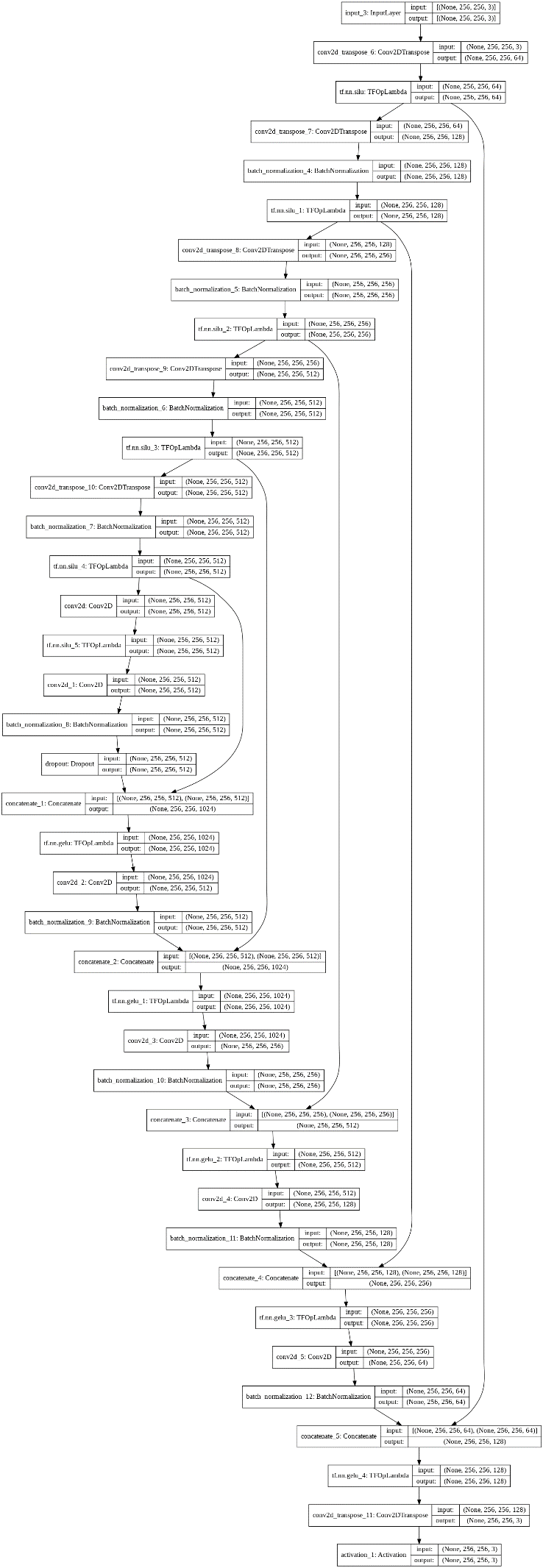
The generator architecture (founded on O-Net) is portrayed below:



------------ Continued on the following page ------------------------



⮉ **Supplementary Table 6**: Model parameters for the generator of the O-Net GAN.



⮉ **Supplementary Figure 6**: Diagram illustrating the model parameters for the O-Net GAN generator outlined in **Supplementary Table 6** above.

## Equations considered for the activation functions

A total of 4 different activation functions were utilized for both the U-Net & O-Net discriminator and generator networks in the present study. These activation functions are as follows:

|  |  |  |
| --- | --- | --- |
| Activation Function | Equation | Source |
| Sigmoid | ------------------------------ (SE1) | (Layer Activation Functions, n.d.) |
| Tanh (Hyperbolic tangent) | ------------ (SE2) | (Layer Activation Functions, n.d.) |
| Swish | ------------------------------ (SE3) | (Ramachandran, Zoph, & Le, 2017) |
| GELU (Gaussian ELU) | , where is the Gaussian CDF of -------------------- (SE4) | (Hendrycks & Gimpel, 2020) |

## Equations considered for the loss functions

The frameworks utilized in our current study are based on a pix2pix generative adversarial network (GAN), incorporating a generator and a discriminator. The loss function utilized for the discriminator losses – *dR* (for real samples) & *dG* (for generated samples) – is the **binary cross-entropy loss** (or log loss) ∈, which may be defined in Equation (SE5) as follows [adapted from Murphy (2012)]:

∈ ----------------------------- (SE5)

where *bi* is the label and represents the probability of *b*i = 1 [derived from Murphy (2012) & (Godoy, 2018)].

According to Isola et al. (2017), a weight of 0.5 is introduced to the discriminator losses to reduce its update rate (as compared to the generator during training).

In addition to binary cross-entropy loss, the final GAN model also incorporates a mean absolute error (MAE) / l1 loss, defined as follows [and adapted from (Shukla, 2015)]:

MAE = ------------------------- (SE6)

where *n* is the number of pixels in the image, *ai* refers to the target value & *bi* refers to the estimated value of the assayed parameter (e.g. pixel RGB (or HSL) intensities at pixel *i*.

The binary cross-entropy loss drives the generator to create more accurate images, while the MAE / l1 loss is used to compare the generated image against the desired image (ground truth) (Isola et al., 2017). Contextually, the overall objective of the GAN may thus be described in SE7 as follows (Isola et al., 2017):

------------------------------------- (SE7)

where L cGAN(*G*, *D*) = 𝔼x,y[log *D*(*x*, *y*)] + 𝔼x,z[log (1 – *D*(*x*, *G*(*x*, *z*))] (Isola et al., 2017).

## Formulae for the image quality metrics

The image quality metrics used in the current study are namely the peak signal-noise ratio (PSNR), signal noise ratio (SNR), the image mean square error (IMSE) & structural similarity index (SSIM). These metrics were applied in MATLAB R2020a (© 1984-2020, The MathWorks, Inc), with each of these metrics being defined as follows:

### Peak signal-noise ratio (PSNR)

PSNR is often used to determine the relative level of resolvable signal *vs* noise in an assayed waveform. Here, the following formula for PSNR is implemented [adapted from Peak signal-to-noise ratio (PSNR) (n.d.)] & assimilated for the purposes of the present study:

PSNR = 10 *lg*() ---------- (SE8) [Source: Peak signal-to-noise ratio (PSNR) (n.d.)]

where PV is the peak (max) value of a pixel in an image (e.g. 255 for an 8-bit image), MSE is the mean squared error and *lg*(*x*) = log10(*x*) (for some constant *x*). In most instances, a PSNR >20 denotes a relatively good (noise-free) image, while PSNR values ranging from 18-20 indicate an acceptable standard of image quality (when factoring the level of background noise).

### Signal-noise ratio (SNR)

SNR is another popular image quality metric to assess the image quality (when considering noise). Bearing similarity to the PSNR metric as described previously, SNR may be computed as follows:

SNR = 10 *lg*() ------------------- (SE9)

where S and N are the respective strengths of the signal and noise (in volts) (Solutions, n.d.). In the current study, the MATLAB function **psnr(*A*, *ref*)** also returns the SNR value of the noisy image *A* (with respect to *ref*) [Peak signal-to-noise ratio (PSNR) (n.d.)].

### Image mean square error (IMSE)

IMSE is often used as a quantitative measure for comparing 2 images, where one image signifies a noisy variant of the other. Figuratively, IMSE may be expressed by Equation (SE10) as follows:

IMSE(*A*, *ref*) = ------------------ (SE10)

where an image *A* is being compared against a standard *ref* of similar size (*n* is the total number of pixels in either *A* or *ref*). The MATLAB-implementation of IMSE using the in-built function **immse** (Mean-squared error, n.d.) is being employed in this context.

### Structural similarity index (SSIM)

SSIM represents an image quality metric which considers 3 primary factors of image analytics, namely (i) luminance l, (ii) contrast *c* and (iii) structure *s* (Structural similarity (SSIM) index for measuring image quality, n.d.). According to Wang et al. (2004), SSIM has been found to be generally superior to other metrics (such as MSE) for comparing different images experiencing varying levels of distortion. Mathematically, the SSIM of an image *A* with reference to a standard (herein denoted as *ref*) may be denoted as follows [adapted from Structural similarity (SSIM) index for measuring image quality (n.d.)]:

SSIM(*A*, *ref*) = [l(*A*, *ref*)]α • [*c*(*A*, *ref*)]β • [*s*(*A*, *ref*)]γ --------------------- (SE11)

where μ*A*, μ*ref*, σ*A*, σ*ref*, and σ*Aref* denote the local means, standard deviations, and cross-covariance for images *A, ref* respectively, and

l(*A*, *ref*) = , *c*(*A*, *ref*) = , *s*(*A*, *ref*) = ------------ (SE12) [Source: Structural similarity (SSIM) index for measuring image quality (n.d.)]

Considering the default MATLAB-assigned values of α = β = γ = 1 and C3 = C2 / 2, a simplified expression for the SSIM may be derived as follows, which is utilized in the current work:

SSIM(*A*, *ref*) = ---------------------- (SE13) [Source: Structural similarity (SSIM) index for measuring image quality (n.d.)]

## Local SSIM Maps & Parameters

The figures below illustrate the local SSIM maps for each of the images shown in Figures 5 & 9 in the manuscript:



⮉ **Supplementary Figure 7**: Local SSIM Maps portraying the differences between the U-Net (& O-Net)-generated images against the Expected (*ground truth*) images, acquired using DIC microscopy. Here, the ellipses highlight visual differences between the generated and the Expected images, which are not clearly identifiable from the local SSIM map. A similar set of images is exhibited in Figure 5 of the manuscript as well.

Further analysis of the parameters for each of the individual ROIs shown in **Supplementary Figure** **7** above are described in the following Table:



⮉ **Supplementary Table 7**: Image analysis parameters for the ROIs indicated in the **Supplementary Figure** **7**. Green-coloured values indicate the better-scoring image in a pair of similar ROIs (obtained from both U-Net and O-Net generated images).

For PCM images, the following figure demonstrates the local SSIM maps obtained from selected ROIs for further analysis:



⮉ **Supplementary Figure 8**: Local SSIM Maps portraying the differences between the U-Net (& O-Net)-generated images against the Expected (*ground truth*) images, acquired using phase contrast microscopy (PCM). Here too, the ellipses highlight visual differences between the generated and the Expected images, which are not clearly identifiable from the local SSIM map. A similar set of images is exhibited in Figure 9 of the manuscript.

Similarly, details on the image parameters for each of these ROIs are presented in the following Table:



⮉ **Supplementary Table 8**: Image analysis parameters for the ROIs indicated in the **Supplementary Figure** **8**. Green-coloured values indicate the better-scoring image in a pair of similar ROIs (obtained from both U-Net and O-Net generated images).

## Data, Codes & Figures Availability

To access and download the figures and code as used in the manuscript, please visit the link below & download the ZIP archive, then extract the folders & files in the archive.

**Download link:**

<https://doi.org/10.1017/S1431927622000782>

The codes are written in Python and are available in the files having **.py** as their extension. Files having the extension **.m** are to be opened (& executed) in MATLAB – these files contain the codes for computing the image quality indices (such as PSNR, SNR, IMSE & SSIM).

Please note that the figures contained in this archive are the raw & unaltered output of the codes used (to facilitate validation & verification of our proposed models, without introducing artifacts which may falsify the results obtained). For this reason, we recommend the reader of the manuscript to increase their screen brightness, contrast and/or gamma if some of the details in the images generated from these Figures are not as clear/apparent to them.

## References

Godoy, D. (22 11, 2018). *Understanding binary cross-entropy / log loss: a visual explanation*. Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

Hendrycks, D., & Gimpel, K. (8 7, 2020). Gaussian Error Linear Units (GELUs). *arXiv:1606.08415*.

Isola, P., Zhu, J.-Y., Zhu, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition*.

*Layer Activation Functions*. (n.d.). (Keras) Retrieved 30 7, 2021, from https://keras.io/api/layers/activations/

*Mean-squared error*. (n.d.). (© 1994-2021 The MathWorks, Inc.) Retrieved 29 7, 2021, from https://www.mathworks.com/help/images/ref/immse.html

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

*Peak signal-to-noise ratio (PSNR)*. (n.d.). (© 1994-2021 The MathWorks, Inc.) Retrieved 29 7, 2021, from https://www.mathworks.com/help/images/ref/psnr.html

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Swish: a Self-Gated Activation Function. *arXiv:1710.05941v1*.

Shukla, R. (28 7, 2015). *L1 vs. L2 Loss function*. Retrieved from http://rishy.github.io/ml/2015/07/28/l1-vs-l2-loss/

Solutions, C. P. (n.d.). *What is Signal to Noise Ratio and How to calculate it?* (© 2021 Cadence Design Systems, Inc.) Retrieved 29 7, 2021, from https://resources.pcb.cadence.com/blog/2020-what-is-signal-to-noise-ratio-and-how-to-calculate-it

*Structural similarity (SSIM) index for measuring image quality*. (n.d.). (© 1994-2021 The MathWorks, Inc.) Retrieved 29 7, 2021, from https://www.mathworks.com/help/images/ref/ssim.html

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612.