Predicting Kinetochore Localization Using Pix2Pix Image Translation of Spindle Pole Body Foci

By

Deep Upadhyay

Senior Honors Thesis

Department of Biology

University of North Carolina at Chapel Hill

May 3rd, 2021

Dr. Elaine Yeh, Thesis Advisor

Dr. Jeff Sekelsky, Reader

Dr. Brian Taylor, Reader

Abstract

Recent advances in machine learning promise to revolutionize biological investigations by creating new methods for image analysis and generation. Deep learning models known as generative adversarial networks (GANs) have shown enormous promise in scientific applications, such as through super-resolution GANs that can artificially upscale image quality. However, GANs have not been applied extensively to predictive image modeling of cellular structures, such as the kinetochore. The kinetochore is the protein complex where spindle microtubules attach during cell division and is crucial in ensuring proper chromosome segregation. Here, we examine whether the Pix2Pix architecture, a type of GAN designed for paired-image translation, can be repurposed to generate predictive fluorescent microscopy images that can localize kinetochore proteins in the budding yeast *S. cerevisiae*. As a proof-of-concept, this architecture's ability to create generated images that matched the ground truth based only on synthetic input images of a different microscopy channel was evaluated. Subsequently, this architecture's robustness was evaluated through its performance on synthetic images with added noise, and then on actual microscopy images of the kinetochore. By comparing generated images with their ground truth targets on various metrics, we find that this architecture is well-suited to generative image modeling of the kinetochore, despite the loss of some fine detail mapping in later tests. Further refinements can improve the network's accuracy and extend its applicability to other kinetochore protein complexes and during different phases of cell division. The development of predictive image modeling architectures such as the Pix2Pix can elucidate the spatial and temporal localization of pathological errors in cell division that result from kinetochore dysfunction, informing further investigations into the causes of human diseases such as cancer.

Introduction

Machine learning is a relatively recent development in computer science that promises to revolutionize image analysis across disciplines. There are a vast range of different architectures and use-cases for machine learning networks, and this versatility has led to their use in biological imaging (Jones, 2020). Generative networks are one such type of architecture that utilize the core principles of machine learning by "training" themselves, learning mathematical patterns from a provided dataset (Goodfellow et al., 2014). Generative networks differ from other architectures in that they allow the generation of novel images based on a given input. Previous studies have demonstrated the applicability of generative networks to biological imaging, and architectures such as generative adversarial networks (GANs) have been used to augment fluorescent microscopy image analysis through super-resolution GANs that can artificially upscale image quality and increase image signal-to-noise ratio (Wang et al., 2018; Zhang et al., 2019).

GANs are a combination algorithm that utilize two competing networks known as the discriminator and the generator. These two networks are trained vis-à-vis each other until the desired level of performance is achieved. The discriminator is first trained independently on a training dataset, which makes it proficient at discerning between a "fake" generated image and the ground truth associated output image for any given input image. The generative network then creates simulated images, while the discriminative network evaluates them for accuracy compared to the ground truth target. Based on how close the generated image was to the ground truth according to the discriminator, the generator refines its mathematical weights and parses over a different batch from the training dataset to improve its predictions. The end goal of the network is to increase the error rate of the discriminator - essentially, to train a generator that can

create images of such high quality that the discriminator cannot distinguish between generated images and the ground truth image for any given input image (Goodfellow et al., 2014).

Despite their aforementioned use in other biological investigations, GANs have not yet been applied to the imaging and localization of the kinetochore, a complex of proteins that associates with duplicated chromatids and is the attachment site for spindle microtubules during cell division. The kinetochore plays a central role in many of the steps required for proper cell division, as it anchors chromosomes to microtubules, helps generate the force required for chromosome motility, and mediates the spindle checkpoint between metaphase and anaphase (Hauf & Watanabe, 2004). As a result, kinetochore protein dysfunction can result in aneuploidy in daughter cells, which is a major risk factor for the development of diseases such as cancer (Cimini, 2008). Fully characterizing the temporal and spatial localization of kinetochore proteins during cell division remains the end goal of these computational approaches, as this information can be a vital step in uncovering the specific mechanisms by which kinetochore dysfunction and subsequent pathogenesis occur.

The location of kinetochores deep within the cell and the fact that they are composed of over 100 distinct proteins in humans can make it challenging to image and track kinetochores over the course of cell division (Kuhn & Dumont, 2014). Computational approaches can significantly reduce the difficulty of localizing kinetochore proteins, and generative image networks are especially applicable to this task as they could allow monitoring of the entire kinetochore complex over time by labeling only a few select proteins. This would grant access to unprecedented information on how the co-localization of various kinetochore proteins changes over the course of cell division and help uncover the mechanisms that underlie kinetochore dysfunction as it relates to human disease. Here, we outline a method to train and test the

performance of the Pix2Pix GAN architecture in generating predictive images that can localize kinetochore protein complexes. The Pix2Pix architecture is a type of GAN that specializes in paired image-to-image transition, making it well-suited as a test case for the effectiveness of generative networks in predictive modeling of the kinetochore using paired microscopy images (Isola et al., 2017). The images of the kinetochore used to train and test the architecture come from *S. cerevisiae*, a simple eukaryote that is useful as a model organism due to its unicellularity and rapid generation time. *S. cerevisiae* is particularly applicable to the study of the kinetochore as its kinetochore proteins are evolutionarily conserved with more complex eukaryotes such as humans (Kitagawa & Hieter, 2001). Analyzing the effectiveness of this architecture on kinetochore protein localization in *S. cerevisiae* is an important initial step in developing robust and refined deep learning approaches that can vastly simplify investigations into both the kinetochore itself and other biological structures.

The Pix2Pix GAN was first modified from a deep learning architecture originally meant to convert sketches of building facades to their paired realistic photos (Isola et al., 2017). The GAN was then trained and tested through paired synthetic images of the kinetochore viewed in green-and red-channel modes, with the green channel representing the localization of kinetochore protein complex Cse4 and the red-channel representing the localization of kinetochore protein complex Nuf2. These proteins were chosen due to their significant positions within the overall structure of the kinetochore, as Cse4 is part of the inner kinetochore and binds to chromatin while Nuf2 is part of the outer kinetochore and binds to microtubules (Yamagishi et al., 2014). The ability to accurately predict the location of the Nuf2 microtubule-binding end of the kinetochore based only on the image data of the Cse4 chromatin-associated end would be invaluable in further characterizing the localization of kinetochore components throughout cell

division. These simulated red-channel and green-channel image pairs were created by the KineticButShakeless program and were free of signal noise, making them a good initial test case for the performance of the GAN (Lawrimore et al., 2019). The aim in this task was for the GAN to be provided with only the green-channel image input and output a predicted image that matched the ground truth paired red-channel image as closely as possible, essentially allowing it to predict Nuf2 localization from Cse4 image data. To test the effectiveness of the GAN in a more realistic scenario with a lower signal-to-noise ratio, the GAN was re-trained and tested on this synthetic image dataset, but with computationally added random noise. Lastly, the ability of the GAN to output images that localized the kinetochore proteins Cse4 and Nuf2 using real microscopy images was tested. In this instance, the GAN was provided with input images of the spindle pole body, which is the functional equivalent of the centrosome in *S. cerevisiae* (Kilmartin, 2014).

Performance in the synthetic image predictions was quantified by comparing the kinetochore-kinetochore distance and full-width half maximum pixel intensity distribution values of the generated predictive images with the ground truth targets. These parameters determine the distance between the brightest intensity foci and pixel intensity spread, respectively. In the real images, the structural similarity index of generated images, which quantifies image similarity based on the parameters of contrast, luminance, and structure, was compared across different categories to further characterize image parameters that may have influenced GAN performance. These comparison categories included the specific protein type used as the target for predictions or the number of kinetochore foci in the target image. Spatial resolution and detail mapping were found to be of high quality in the noiseless synthetic images. In the follow-ups on noisy synthetic images and real images, detail mapping was less robust, but spatial relationships were generally

preserved. The performance of the network shows promise for the applicability and versatility of Pix2Pix/GAN architectures in predictive localization of the kinetochore and other biological structures.

Methods

Synthetic Image Dataset Creation and Noise Addition

The simulated kinetochore image generation program KineticButShakeless (Lawrimore et al., 2019), written in MATLAB, was used to create a dataset of 10,000 40 x 40-pixel paired red-channel and green-channel image sets of the kinetochore. Default two-color settings were used, and each synthetic set of images was grouped into 11 z-axis slice stacks. Synthetic image inputs can be seen in Figure 3. Images were read into Google Colab as arrays using the Python library NumPy, and pixel intensity values were normalized from [-1,1] by dividing the arrays by 32,767.5 and then subtracting 1 due to the 16-bit datatype (Bisong, 2019; Harris et al., 2020). The dataset was generated using the TensorFlow function data.Dataset.from_tensor_slices(), and then resized to 64 x 64 pixels using the TensorFlow bicubic image resize function before being passed into the architecture for training (Abadi et al., 2016). For the noise-added dataset, noise was computationally added by summing randomized intensity values to every pixel in the dataset array using the NumPy library (Harris et al, 2020). These values were drawn from a Gaussian distribution with a mean of 500 and a standard deviation equal to the mean intensity value of all input and target images divided by 4.

Real Image Acquisition and Processing

Real kinetochore microscopy images were acquired from an existing dataset generated in a prior in-house study, the methods of which are reported here verbatim (Lawrimore et al., 2019). Budding yeast strain YEF473A (MATa trp1Δ63, leu2Δ1, ura3-52, his3-. Δ200, lys2-801) was transformed with SPC29-RFP:HYGR to fluorescently label the spindle pole bodies (SPBs) to generate the strain KBY7999. Strain KBY7999 was transformed with GFP-NUF2:NATR to generate strain KBY8169. Budding yeast strain YEF 473A was transformed with pKK1 to fluorescently label Cse4 with GFP and the endogenous Cse4 was removed and replaced with HYGR to generate strain KBY2010. Strain KBY2010 was transformed with SPC29-RFP:KANR to fluorescently label the SPBs and generate strain DCY1196.1. Seven Z-plane image stacks of Spc29-RFP, N-terminal GFP-Nuf2 (KBY8169) and Spc29-RFP, Cse4-GFP (DCY1196.1) yeast strains were acquired with a Nikon Eclipse Ti TE2000-U inverted fluorescent microscope using a Nikon Apo 1.4 NA 100x objective, MetaMorph 7.8 software, Hamamatsu Orca Flash 4.0 LT camera, and LumenCor Aura Light Engine. The cells in the images were segmented using a MATLAB code repository, CellStarSelect1, that utilizes the CellStar segmentation algorithm (Versari et al., 2017) for segmenting budding yeast buds from brightfield microscopy images. The MATLAB function spotDetection calls the CellStar program to segment yeast buds and uses the function advPointSourceDetection.m (Cicconet et al., 2017), which was based on code developed for Aguet et al. (2013), to detect kinetochore and SPB foci within the bud segment in the fluorescent image channels. If two kinetochore foci and two SPB foci are detected, the bud segment is analyzed and saved in a cell array. The function compileImages.m parses the cell array containing the segmented fluorescent images and creates a 50 × 50-pixel image around each bud that contained two kinetochore and SPB foci. The seven-step 50 × 50-pixel stack of

each bud was condensed into a single plane using a maximum projection approach and had their intensity values normalized and saved as 16-bit, RGB images. Duplicates of the images were generated by rotating the images to generate seven additional orientations of the initial to increase the size of the dataset. The images then underwent a background subtraction procedure and were denoised using a low-pass 2D Wiener filter.

Training of the Pix2Pix Generative Adversarial Network

The architecture used was a modified version of the Pix2Pix conditional GAN created by Isola et al. (2018), with the code base provided by Google's open-source TensorFlow Core (TensorFlow Tutorials, 2021). Google Colab, which allows for cloud computing using high-end hardware, was used for building and training the algorithm. The convolutional layers and architecture of the discriminator can be seen in Figure 1A while the architecture of the generator can be seen in Figure 1B. The training batch size was set to 1000 images for the synthetic image datasets and to 440 images for the real image datasets due to the differing dataset sizes. The loss functions of the network were monitored over time to observe training progress over epochs. Four total loss equations were used, defined as follows: discriminator loss, which measures how accurately the discriminator is classifying images as either generated or ground truth; generator adversarial loss, which measures how well images made by the generator can trick the discriminator; generator L1 loss, which measures specifically how well each generated image matches up to the ground truth; and total generator loss, which is a composite of the other two generator loss functions. Loss function definitions were unmodified from their equations in the original Pix2Pix architecture (Isola et al., 2017). Training in each case was terminated once the loss functions of the network appeared to have reached an asymptote.

Quantification and Analysis of GAN Performance

GAN performance on the synthetic images was analyzed by comparing the distributions of the kinetochore to kinetochore (KK) distance and the full-width at half maximum intensity (FWHM) values of the predicted image set to those of the ground truth target image set. KK distance was measured in nanometers through MATLAB functions that recorded the distance between the brightest pixels of each kinetochore foci in a given image (Lawrimore et al., 2019l). The FWHM values were found by collapsing intensity values to one dimension and then curve-fitting a one-dimensional Gaussian distribution to measure the overall width of the datasets. Rank-sum tests using the stats.rank-sum function from the Python library SciPy were used to compare the output KK and FWHM versus the target KK and FWHM for both the noiseless and noise-added conditions (Virtanen et al., 2020).

For the real images localizing Cse4 and Nuf2 based on spindle pole body inputs, the structural similarity index (SSIM) was used. SSIM was computationally calculated using the TensorFlow image.ssim function as a number between -1 and 1 indicating how similar each predicted image was to its ground truth target based on the mathematical parameters of luminance, contrast, and structure (Yang et al., 2017). The mean SSIM values of the Nuf2 outputs were compared versus the Cse4 outputs. In addition, the mean SSIM for both Cse4 and Nuf2 predicted images with two foci in the target image was compared with the mean SSIM of predicted images with more or less than two foci in the target image. Rank-sum tests were conducted using the same method as in the synthetic datasets for these three comparisons to evaluate statistical significance.

Results

Network Performance Improved Over Training Time

The performance of the network in predicting the ground truth over training time is quantified as algorithm loss. Loss is a statistical measure of how well the algorithm is modeling the pixel intensity data when comparing predicted output images to the ground-truth target images. Network performance is improved by minimizing loss, and a lower loss for the generator implies a network better at its function. In order to confirm the network was improving during training and pick up on any indicators of the network's shortcomings based on trends in loss, the loss of the generator and discriminator over training time was monitored using TensorBoard (Abadi et al., 2016) (Figure 2, see methods for information on how loss functions were defined).

Discriminator loss, or its ability to accurately differentiate generated images from ground truth ones, initially decreased as it was trained, then slowly increased as the generator's images became closer to the ground truth (Figure 2A). Generator adversarial loss, which measures how well the generator's images could trick the discriminator, decreased in opposition to discriminator loss as the generator was trained (Figure 2B). Generator L1 loss also decreased, which is important in ensuring that the generator is in fact modeling the ground truth accurately and not trapped in a local minimum where it is creating images that can satisfy the discriminator. (Figure 2C). Predictably, total generator loss decreased as it is a composite of the other two generator loss functions (Figure 2D). Each loss function eventually reaches an asymptote, indicating the point at which additional training epochs are ineffective at improving network accuracy. Due to a technical error in Google Colab, loss graphs for the training runs on the other datasets (noise-added synthetic, Cse4, and Nuf2) could not be obtained. However, this is unlikely to have affected overall results or conclusions given the network's architecture itself was

conserved across each dataset, and the network's trends in loss over training time confirm that training was proceeding as expected.

Generated Images Accurately Predict Broad Features and Boundaries

After training was complete, a set of sample images were created by the synthetic image generator and the synthetic noise-added image generator to demonstrate the qualitative visual efficacy of the network in predicting the ground truth of a given input image (Figure 3A,3B). The comparison of the ground truth and generated images from the synthetic generator demonstrates that the generative algorithm was able to capture the correct spatial orientation of the ground truth images, and additionally was able to replicate the finer detail near the foci of each image. The noise-added generator shows a decreased ability to capture minute details, especially around edges and boundaries, but is overall still capable of accurately capturing the correct spatial orientation and inner features of the ground truth in its generated images. The same process was followed for the real image cases. In these comparisons, one generator had been trained to localize Cse4 based on spindle pole body as an input, while the other had been trained to localize Nuf2 based on the spindle pole body as an input. Although general spatial relationships are preserved, there is a noticeable decline in fine detail mapping in both the Cse4 and Nuf2 generators (Figure 4A,4B). The network is unable to precisely model boundaries between high and low intensity regions in this scenario, but accuracy improves near the high intensity foci. Qualitatively, the localization of the Nuf2 outputs appears to more closely match the ground truth compared to the boundaries of the Cse4 outputs.

Lower Signal-to-Noise Ratio Causes a Decline in Network Performance

The practical imaging of cellular structures often presents with the issue of visual signal noise in captured images. To determine the effect of a lower signal-to-noise ratio more reflective of typical experimental conditions, the GAN was retrained on the synthetic image dataset but with random signal noise added. The performance of the two generators trained on the noiseless and noise-added datasets was quantified by comparing the kinetochore-to-kinetochore (KK) distance and full-width at half maximum (FWHM) intensity values of the generated images to the ground truth target images they were aiming to mimic. A rank-sum test was utilized to measure whether there was a statistically significant difference in the KK and FWHM distributions for the predicted images versus the ground truth images, for both the noiseless and noise-added generators.

The addition of noise worsens both KK distance and FWHM modeling by the network, although the KK distance is better replicated than the FWHM even in the noiseless condition. This can be observed visually in the KK distance and FWHM distributions, where the KK distribution in the noiseless dataset matches up closely to the ground truth but the FWHM distribution is visually distinct (Figure 5A). In the noise-added dataset, the output KK distribution is not as close a match to the ground truth KK but is still qualitatively the same shape, while the FWHM distribution appears to be significantly altered compared to the ground truth FWHM (Figure 5B). In the noise-added generator rank-sum comparison, the KK distance was again found not to differ significantly, though at a borderline value ($P = 0.0498$), while the FWHM distribution still demonstrated a statistically significant difference ($P < 0.001$). Although both metrics appear to decline in the noise-added generator, the KK distribution is still modeled with a greater accuracy than the FWHM is based on both the qualitative and quantitative comparisons.

Protein Selection and Image Foci Count Affect Network Performance

In order to quantify the performance of the network at predicting protein localization from real images of the spindle pole body, the structural similarity index (SSIM) of the output images was compared in three different cases for the real image-trained generators. The SSIM ranges from -1 to 1 and is a measure of how closely an output image matches up to its ground truth, with 1 indicating a perfect recreation based on the parameters of luminance, contrast, and structure recreation. These comparisons were meant to determine if certain proteins were better localized by the network than others, and whether the number of intensity foci in the target images affected performance.

The average SSIM was measured between Cse4 target and output images and then between Nuf2 target and output images to determine whether the protein selected as the target can alter the performance of the network. For the second comparison, the average SSIM between Cse4 images with two foci and those with more or less than two foci. The third comparison was similar to the second but compared SSIM values between Nuf2 images with two foci and those with more or less than two foci. The last two comparisons serve to test whether the presence of two foci causes an improvement or decline in GAN performance. A rank-sum test was then conducted for all three distribution comparisons.

The mean SSIM value was found to be about 11.5% higher for Nuf2 than Cse4, with a statistically significant difference between the SSIM distributions from the rank-sum test ($P < 0.001$), indicating this network was better at localizing Nuf2 than Cse4 from spindle pole body inputs (Figure 6). The ability of the network to localize Cse4 in images with more or less than two foci was marginally better with the mean SSIM for these images being about 1.53% higher

than the SSIM for two foci images (Figure 6). The difference between the distributions from the rank-sum test for the second comparison was also highly significant ($P < 0.001$). The difference between SSIM distributions for Nuf2 with two foci versus more or less than two foci was not found to be statistically significant ($P = 0.3627$).

Discussion

The Pix2Pix Network as a Method for Kinetochore Image Translation and Generation

Here, we outline a deep learning method and performance evaluation for the Pix2Pix generative architecture in determining the localization of kinetochore proteins based on input images of other structures. The network was able to effectively predict kinetochore foci separation in the synthetic images, with fine detail mapping and internal features recreated with high accuracy (Figure 3). Despite worse fine detail mapping in the real images, the spatial relationships between image foci were generally well preserved (Figure 4).

Declines in network performance in other conditions were generally expected. The addition of noise affected KK distance accuracy, but to a lesser degree than FWHM accuracy (Figure 5). The decline in FWHM accuracy in conditions with more noise is expected as FWHM is a measure of how wide the intensity spread around image foci is, and the generator cannot recreate this spread as the unique noise profile of the output is not predictable based on the noise profile of the input. In the case of the real images, the largest factors causing the decline in detail mapping may be the small sample size and lower predictive association between the spindle pole body and Cse4/Nuf2 leading to overfitting. Overfitting can often be the main impediment in training GANs and refers to the algorithm becoming “trapped” in a local loss minimum. In these cases, the generator has settled on a loss value with suboptimal weights as they allow it to

consistently trick the discriminator with its outputs, meaning the generator will not adjust its weights and improve to its peak capability (Yazici et al., 2020). This overfitting may be the cause of the real image generator modeling the foci correctly to minimize the loss by convincing the discriminator, but not improving its boundary modeling. This is evident in the Cse4 and Nuf2 image comparisons, where the foci locations and nearby features are modeled correctly but boundaries and details further away are less accurately recreated (Figure 4).

The significant drop in performance when spindle pole bodies are used as an input to predict protein localization rather than a second kinetochore protein indicates the importance of selecting inputs that are in fact predictive of the desired target's localization (Figure 4). Despite the sensitivity of GAN performance to noise and input/target selection, the high structural similarity index values demonstrate that even in less-than-ideal conditions, the GAN models protein localization with a high level of accuracy (Table 2). The comparison of the SSIM values further informs the performance on the real images. The improved performance when modeling Nuf2 versus Cse4 is expected, given the fact that Cse4 associates with chromatin and has a more variable distribution (Figure 6) (Haase et al., 2013). The increase in performance on Cse4 images with more or less than two foci may have occurred because of the low predictability of the spindle pole body input combined with the high variability of Cse4, as the network modeled details near foci much more accurately than features further away. With these adverse conditions, the generator has likely settled into a local loss minimum where it manages to overcome the discriminator by learning the associations of image foci rather than modeling all relevant image boundaries, and increased numbers of foci augment this overfit generator's predictions. This increase in SSIM on images with more or less than two foci is absent in Nuf2,

which likely did not suffer as much from overfitting during training due to its more predictable localization.

Future Directions

The initial performance and findings demonstrated by the Pix2Pix GAN architecture in this study can be used as a basis to further refine this network for the purposes of predictive image modeling of the kinetochore. Expanding the sample size of real images could make up for some of the disparity in performance between the synthetic image generators versus the real image generators, as discounting rotational variants the synthetic image datasets had almost 20 times the number of images. Using a more predictive input in real images than spindle pole bodies can also provide the network with more training information to learn localization patterns, reducing the chance of overfitting.

The evaluation of the architecture described in this study is the first step in the creation of networks that can predict the localization of other proteins or structures based on static image data alone. This would allow the construction of temporally sensitive models of the kinetochore while circumventing the complications and challenges of live-cell imaging. The interpolative ability of these networks allows the construction of models of processes such as cell division with greater speed and precision than with traditional imaging techniques, simplifying investigations into various biological mechanisms. Beyond further refining network performance, the immediate next steps will be the extension of the network to volumetric 3-D imaging, either through plane-to-plane or full-volume modeling. This can then be combined with time-labeling based on cell-cycle stage, as this information is already encoded in images through the amount of separation between kinetochore foci. These two techniques together would allow

the creation of an artificial time-lapse in video form of kinetochore protein localization over the course of cell division.

The extensible nature and broad applicability of GANs is readily apparent from these findings, as shown here through the modification of an architecture originally meant to translate sketches into photos of building facades into one that can accurately generate predictive images of kinetochore protein complexes. The architecture described here can be adopted for the study of numerous other biological structures with just a few modifications, allowing for cooperative refinement and application of this model among biological investigators. These findings validate the enormous promise of GANs as versatile tools for modeling cell division and other biological processes to uncover novel causes of pathogenesis.

Acknowledgements

I would like to express my gratitude to UNC's Office of Undergraduate Research and Summer Academic Programs for their financial support through the 2020 Summer Award for Research Intensive Courses, which helped cover tuition costs during the initial stages of this project.

Thank you to Dr. Amy Maddox for her teaching and informative advice on writing and refining this thesis. I am immensely grateful for the extensive mentoring provided by Dr. Joshua Lawrimore, a postdoctoral researcher in the Bloom Lab. Thank you to Dr. Elaine Yeh for her guidance and serving as an advisor for the honors thesis and in prior research courses. Lastly, thank you to both Dr. Elaine Yeh and Dr. Kerry Bloom for their professional support and giving me the opportunity to learn and work in the Bloom Lab.

References

Abadi, Martin, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI16) (pp. 265–283).

Aguet, F., Antonescu, C. N., Mettlen, M., Schmid, S. L., & Danuser, G. (2013). Advances in Analysis of Low Signal-to-Noise Images Link Dynamin and AP2 to the Functions of an Endocytic Checkpoint. *Developmental Cell*, *26*(3), 279–291. https://doi.org/10.1016/j.devcel.2013.06.019

Bisong, E. (2019). Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 59–64. https://doi.org/10.1007/978-1-4842-4470-8_7

Cicconet, M., Hochbaum, D. R., Richmond, D., & Sabatini, B. L. (2017). Bots for Software-Assisted Analysis of Image-Based Transcriptomics. *Proceedings of TheIEEE International Conference on Computer Vision Workshop (ICCVW)*. https://doi.org/10.1101/172296

Cimini, D. (2008). Merotelic kinetochore orientation, aneuploidy, and cancer. *Biochimica Et Biophysica Acta (BBA) - Reviews on Cancer*, *1786*(1), 32–40. https://doi.org/10.1016/j.bbcan.2008.05.003

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Haase, J., Mishra, P. K., Stephens, A., Haggerty, R., Quammen, C., Taylor, R. M., … Bloom, K. (2013). A 3D Map of the Yeast Kinetochore Reveals the Presence of Core and Accessory Centromere-Specific Histone. *Current Biology*, *23*(19), 1939–1944. https://doi.org/10.1016/j.cub.2013.07.083

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hauf, S., & Watanabe, Y. (2004). Kinetochore Orientation in Mitosis and Meiosis. *Cell*, *119*(3), 317–327. https://doi.org/10.1016/j.cell.2004.10.014

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.632

Jones, D. T. (2019). Setting the standards for machine learning in biology. *Nature Reviews Molecular Cell Biology*, *20*(11), 659–660. https://doi.org/10.1038/s41580-019-0176-5

Kilmartin, J. V. (2014). Lessons from yeast: the spindle pole body and the centrosome. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *369*(1650), 20130456. https://doi.org/10.1098/rstb.2013.0456

Kitagawa, K., & Hieter, P. (2001). Evolutionary conservation between budding yeast and human kinetochores. *Nature Reviews Molecular Cell Biology*, *2*(9), 678–687. https://doi.org/10.1038/35089568

Kuhn, J., & Dumont, S. (2014). Imaging and physically probing kinetochores in live dividing cells. *Methods in Cell Biology*, 467–487. https://doi.org/10.1016/b978-0-12-420138-5.00025-2

Lawrimore, J., Doshi, A., Walker, B., & Bloom, K. (2019). AI-Assisted Forward Modeling of Biological Structures. *Frontiers in Cell and Developmental Biology*, *7*. https://doi.org/10.3389/fcell.2019.00279

Pix2Pix : TensorFlow Core. TensorFlow Tutorials. (2021). https://www.tensorflow.org/tutorials/generative/pix2pix.

Versari, C., Stoma, S., Batmanov, K., Llamosi, A., Mroz, F., Kaczmarek, A., … Batt, G. (2017). Long-term tracking of budding yeast cells in brightfield microscopy: CellStar and the Evaluation Platform. *Journal of The Royal Society Interface*, *14*(127), 20160705. https://doi.org/10.1098/rsif.2016.0705

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wang, H., Rivenson, Y., Jin, Y., Wei, Z., Gao, R., Günaydın, H., … Ozcan, A. (2018, December 17). *Deep learning enables cross-modality super-resolution in fluorescence microscopy*. https://www.nature.com/articles/s41592-018-0239-0.

Yamagishi, Y., Sakuno, T., Goto, Y., & Watanabe, Y. (2014). Kinetochore composition and its function: lessons from yeasts. *FEMS Microbiology Reviews*, *38*(2), 185–200. https://doi.org/10.1111/1574-6976.12049

Yang, J., Lin, Y., Ou, B., & Zhao, X. (2016). Image decomposition-based structural similarity index for image quality assessment. *EURASIP Journal on Image and Video Processing*, *2016*(1). https://doi.org/10.1186/s13640-016-0134-5

Yazici, Y., Foo, C.-S., Winkler, S., Yap, K.-H., & Chandrasekhar, V. (2020). Empirical Analysis Of Overfitting And Mode Drop In Gan Training. *2020 IEEE International Conference on Image Processing (ICIP)*. https://doi.org/10.1109/icip40778.2020.9191083

Zhang, H., Fang, C., Xie, X., Yang, Y., Mei, W., Jin, D., & Fei, P. (2019). High-throughput, high-resolution deep learning microscopy based on registration-free generative adversarial network. *Biomedical Optics Express*, *10*(3), 1044. https://doi.org/10.1364/boe.10.001044

Figures

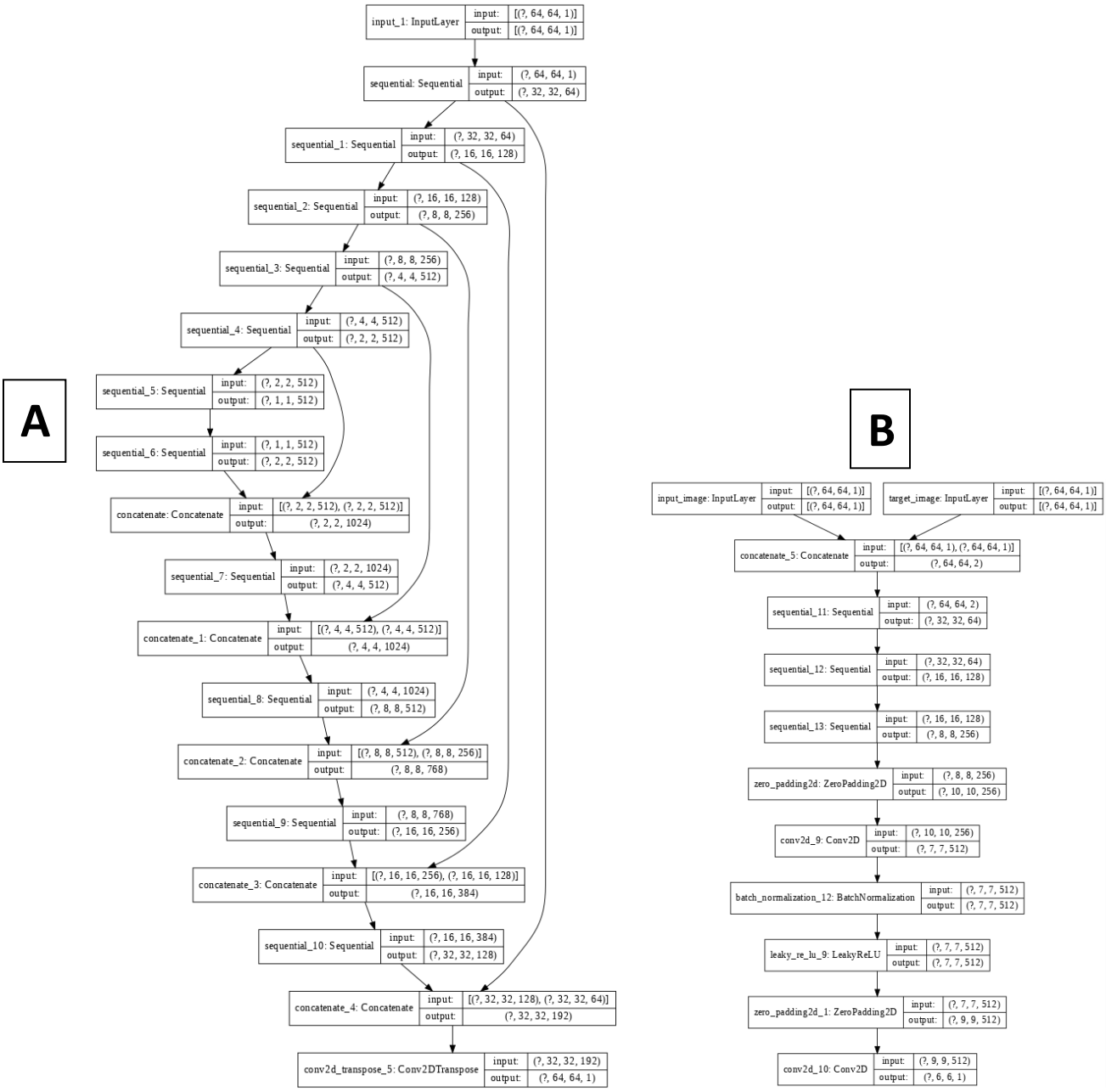

FIGURE 1 | Graphical representations of the architectures used in the GAN. Arrows represent computational steps between convolutional layers. Each convolutional layer shows the shape of the data after a linear operation in which the values in the layer before it are multiplied by a set of weights, with the weights adjusted after each training epoch as the network improves. **(A)** Pix2Pix generator architecture used to create predictive images. **(B)** Pix2Pix discriminator architecture used to judge whether images created by the generator were the ground truth or predictions.

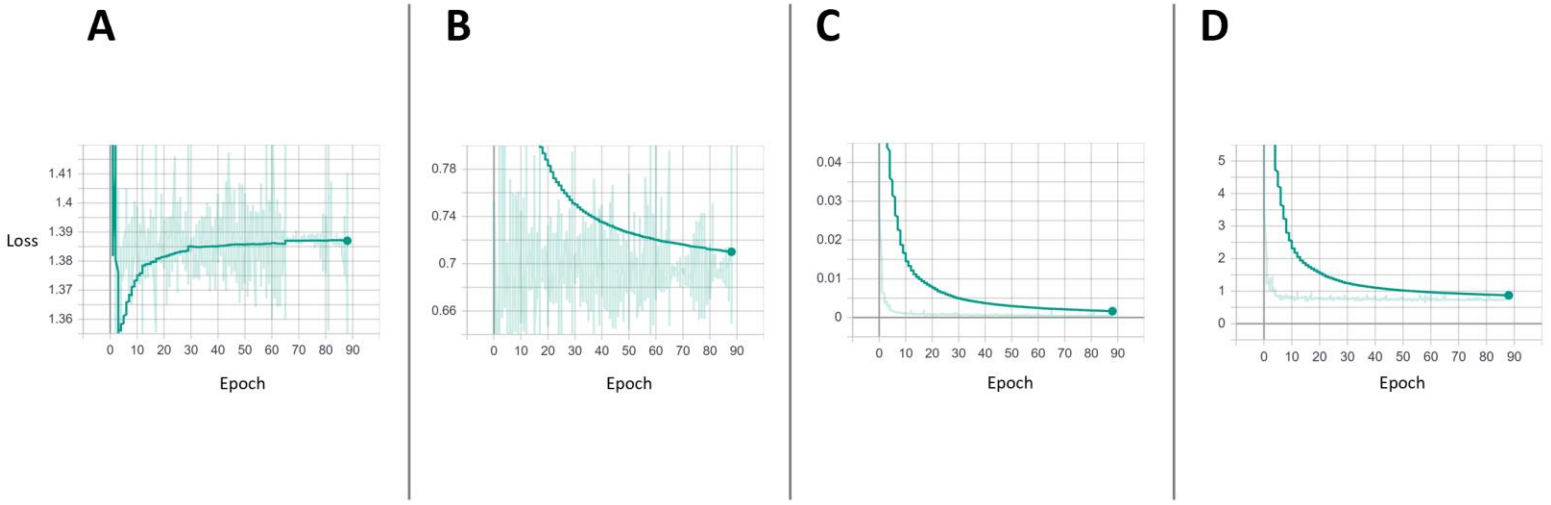

FIGURE 2 | Loss over training epochs. Light green background values represent the raw data, while the solid darker line shows a smoothed line obtained from an exponentially weighted moving average to improve readability. **(A)** Discriminator loss. **(B)** Generator adversarial loss, a measure of how well the discriminator could trick the discriminator. **(C)** Generator L1 loss, a measure of how well the generator's inputs matched the target image. **(D)** Total generator loss, a composite of the adversarial and L1 losses.

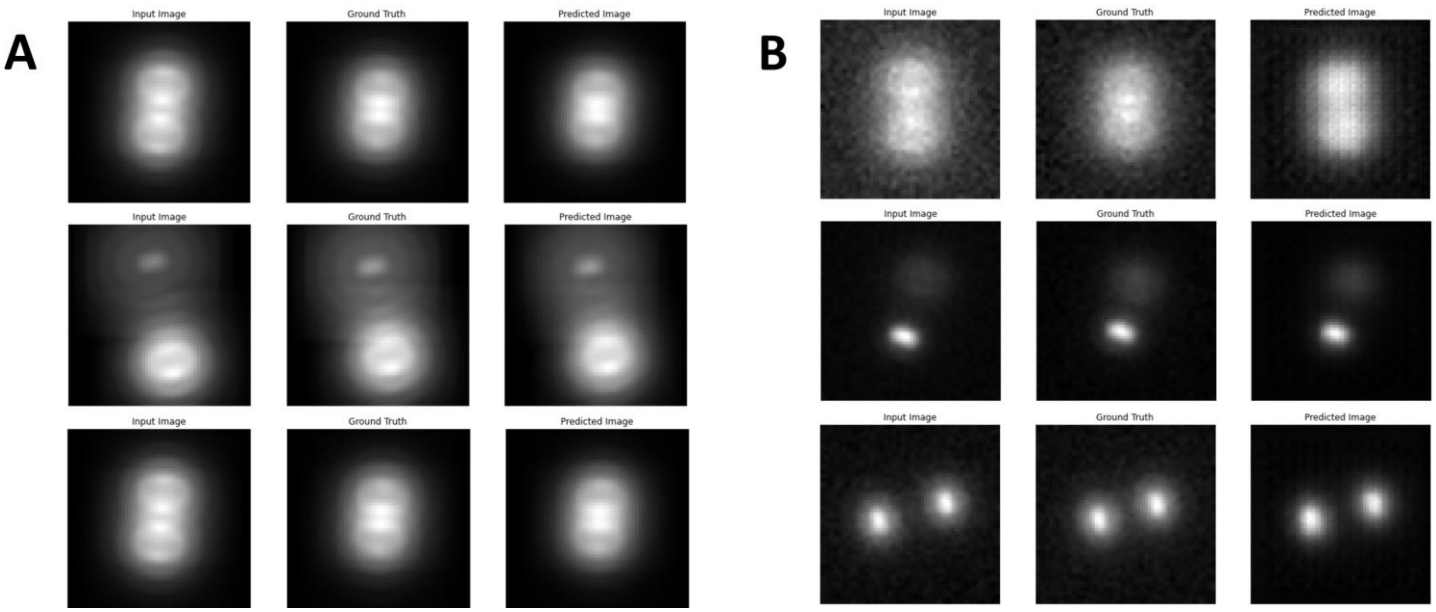

FIGURE 3 | Representative images generated by the GAN trained on the synthetic dataset. The input image (leftmost) was provided to the network, which it used to generate a predicted output (rightmost) that matched the target (middle) without having knowledge of what the present target looked like. **(A)** Images generated by the network trained on the noiseless dataset. **(B)** Images generated by the network trained on the noise-added dataset.

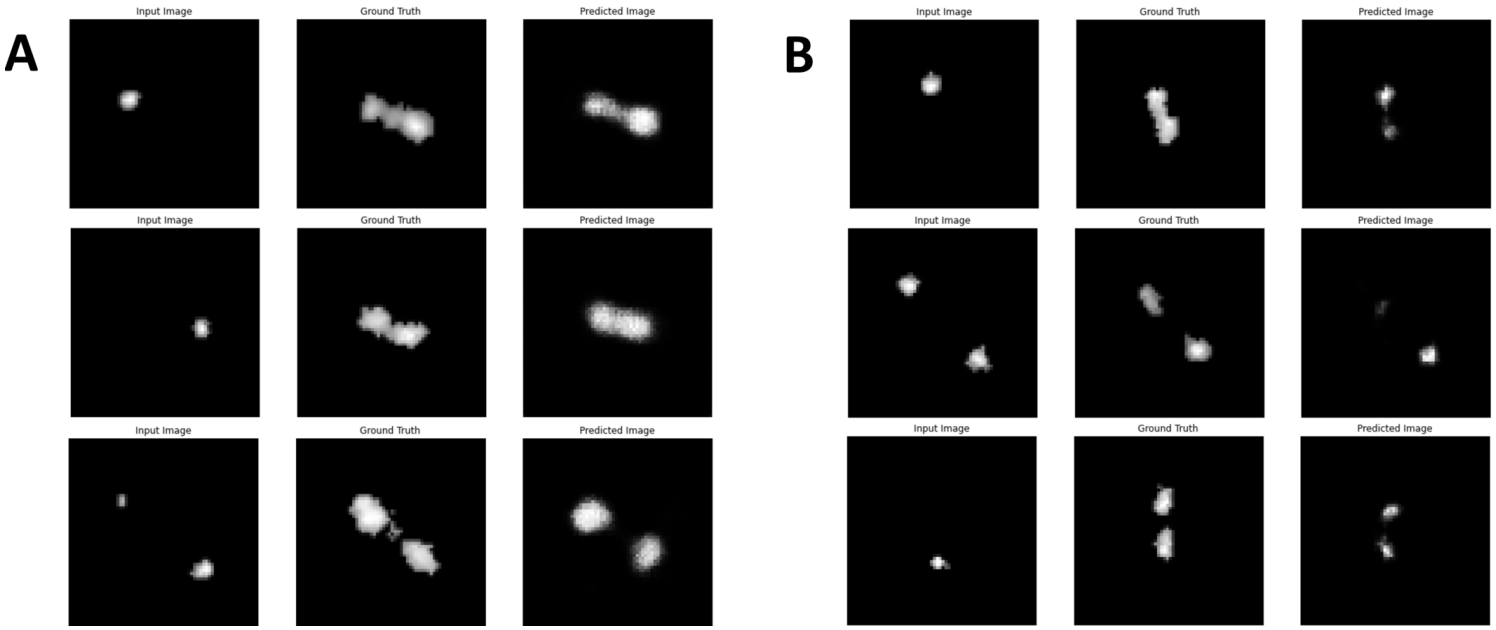

FIGURE 4 | Representative images generated by the GAN trained on the real image dataset. Image maximum intensity projections were denoised using a Wiener filter. As with the synthetic dataset, the input image (leftmost) was provided to the network, which it used to generate a predicted output (rightmost) that matched the target (middle) without having knowledge of what the present target looked like. **(A)** Images generated by the network trained on the Cse4 dataset. **(B)** Images generated by the network trained on the Nuf2 dataset.

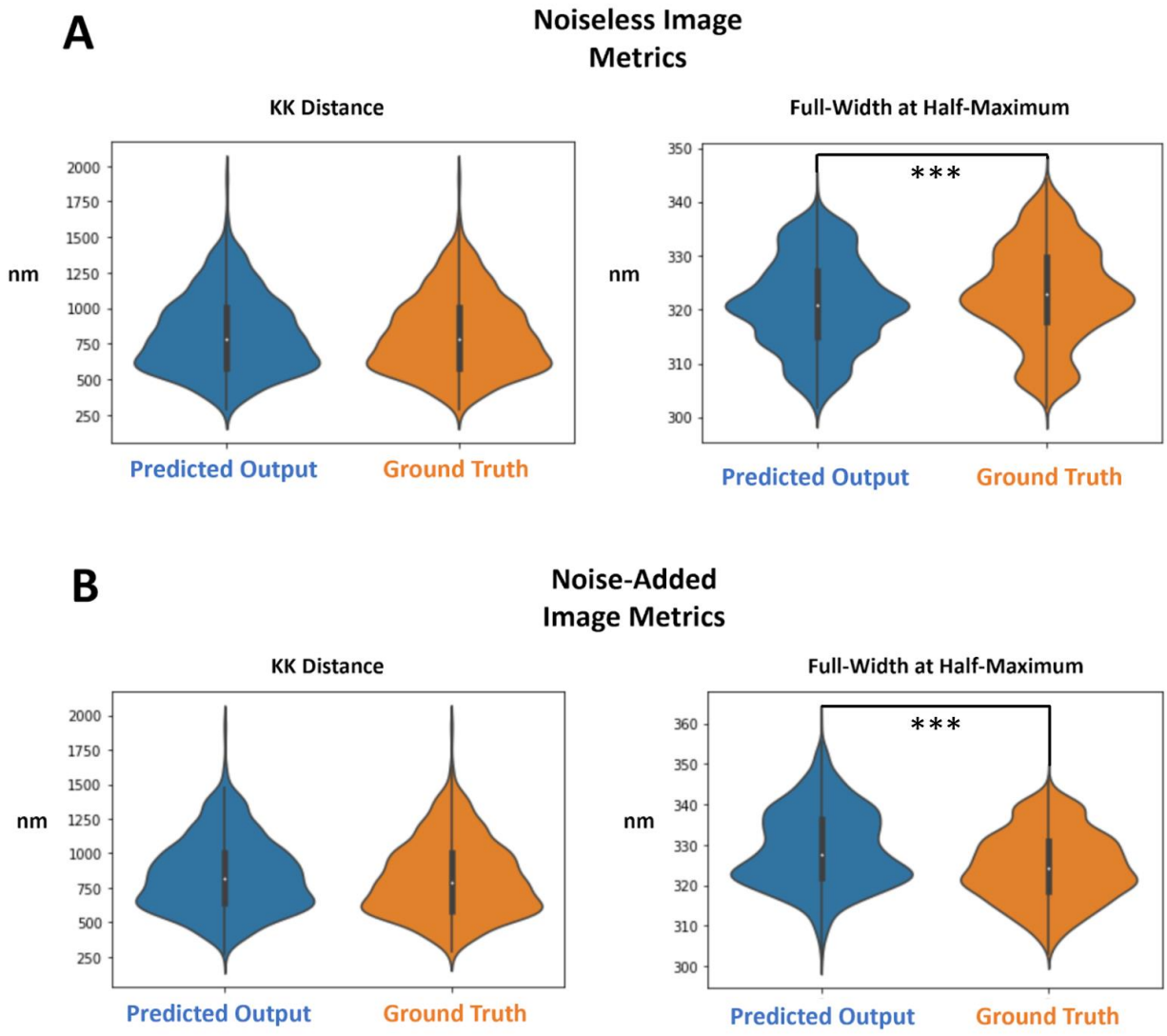

FIGURE 5 | Comparison of image metric distributions between the ground truth target images and the generated predictions. Blue represents predicted output metrics while orange represents ground truth metrics. Three asterisks indicate statistical significance ($P < 0.001$) in a rank-sum test between distributions. **(A)** Kinetochore-to-kinetochore (KK) distance distribution comparison ($P = 0.9635$) and full-width at half maximum (FWHM) distribution comparison ($P < 0.001$) for the noiseless generated images. **(B)** KK distance distribution comparison ($P = 0.0498$) and FWHM distribution comparison ($P < 0.001$) for the noise-added generated images.

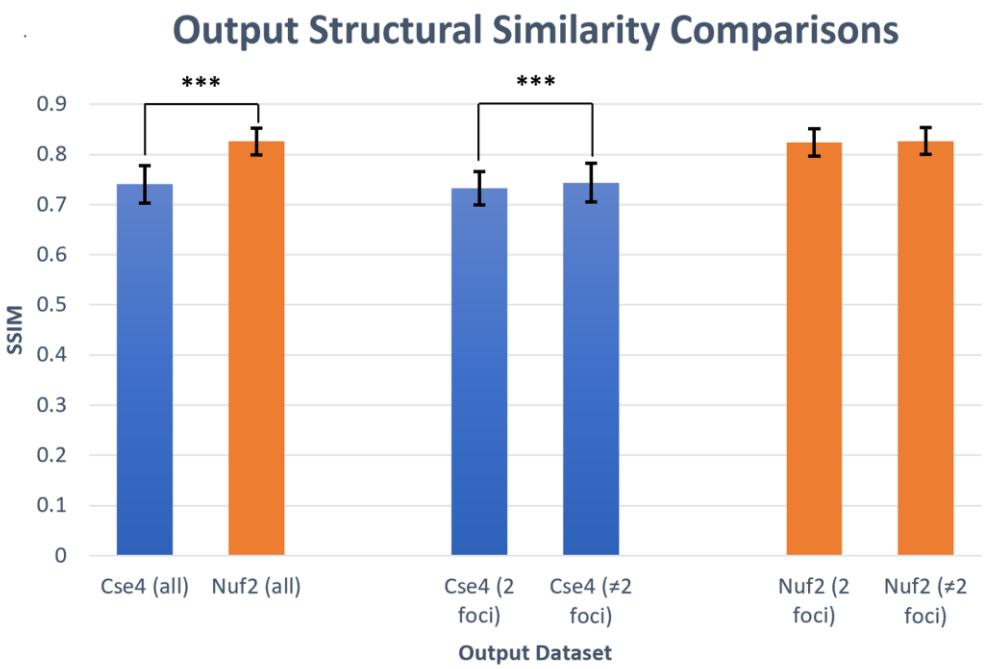

FIGURE 6 | Comparison of the average structural similarity index (SSIM) over all images generated by the GAN trained on the real image dataset. A higher SSIM indicates the generated images more closely match the ground truth dataset. Blue bars represent Cse4 output datasets, while orange bars represent Nuf2 output datasets. Error bars indicate the standard deviation of each dataset's SSIM. Three asterisks indicate statistical significance ($P < 0.001$) in a rank-sum test of SSIM distribution between indicated datasets. The leftmost group shows a comparison between the SSIM mean over all Cse4 generated images versus all Nuf2 generated images (rank-sum test $P = 0.000$). The center group shows a comparison between the SSIM mean between Cse4 images with two spindle pole body foci in the target image versus those with more or less than two spindle pole body foci in the target image (rank-sum test $P = 5.1979 * 10^{-17}$). The rightmost group shows the same comparison as the center group, but for Nuf2 instead (rank-sum $P = 0.3627$).

Tables

Compared Datasets	Parameter	Rank-sum Test P-value
Noiseless Output vs. Noiseless Target	KK	0.9635
Noiseless Output vs. Noiseless Target	FWHM	$3.2570 * 10^{-14}$
Noise-Added Output vs. Noiseless Target	KK	0.0498
Noise-Added Output vs. Noiseless Target	FWHM	$1.7174 * 10^{-42}$
Cse4 (all) Output vs. Nuf2 (all) Output	SSIM	0.0000
Cse4 (2 foci) Output vs. Cse4 (≠2 foci) Output	SSIM	$5.1979 * 10^{-17}$
Nuf2 (2 foci) Output vs. Nuf2 (≠2 foci)	SSIM	0.3627

TABLE 1 | Statistical rank-sum test P-values for each compared dataset. Kinetochore-kinetochore distance and full-width at half maximum distributions were compared between output images from a generator and the target ground-truth images. SSIM was compared across two separate outputs from different generators.

Output Dataset	Mean SSIM	Standard Deviation of SSIM
Cse4 (all)	0.7403	0.0741
Cse4 (2 foci)	0.7322	0.0658
Cse4 (≠2 foci)	0.7434	0.0769
Nuf2 (all)	0.8258	0.0535
Nuf2 (2 foci)	0.8238	0.0543
Nuf2 (≠2 foci)	0.8267	0.0531

TABLE 2 | Structural similarity index mean and standard deviation for each real-image output category, used to compare the effectiveness of the real-image generators for differing image characteristics.

`