

Computer Vision CS-GY 6643 - Final Project - Super Resolution Microscopy

Shambhavi Seth, ss17936@nyu.edu, Aryan Prasad, ap7949@nyu.edu, Roman Vakhrushev, rv1057@nyu.edu, Ishan Yadav, iy2159@nyu.edu

1 Introduction and background

Problem Statement: Enhance cell images using Super-resolution techniques.

Importance: Traditionally, in the field of microscopy, resolution is defined as the minimal distance at which two distinct points can be differentiated. While traditional microscopy is limited by the diffraction limit of the light, which is approximately 200-300 nm (also known as “diffraction barrier”), super-resolution microscopy (SRM) tries to bypass this limit and achieve the resolution of 20 nm or even smaller. This provides the researchers with an opportunity to study small biological structures such as proteins, small organelles, cellular membranes, neural connections, and even molecules (Breedijk et al. 2020). SRM techniques allow the visualization of subcellular organization with unprecedented detail, but also confront biologists with the challenge of selecting the best-suited approach for their particular research question.

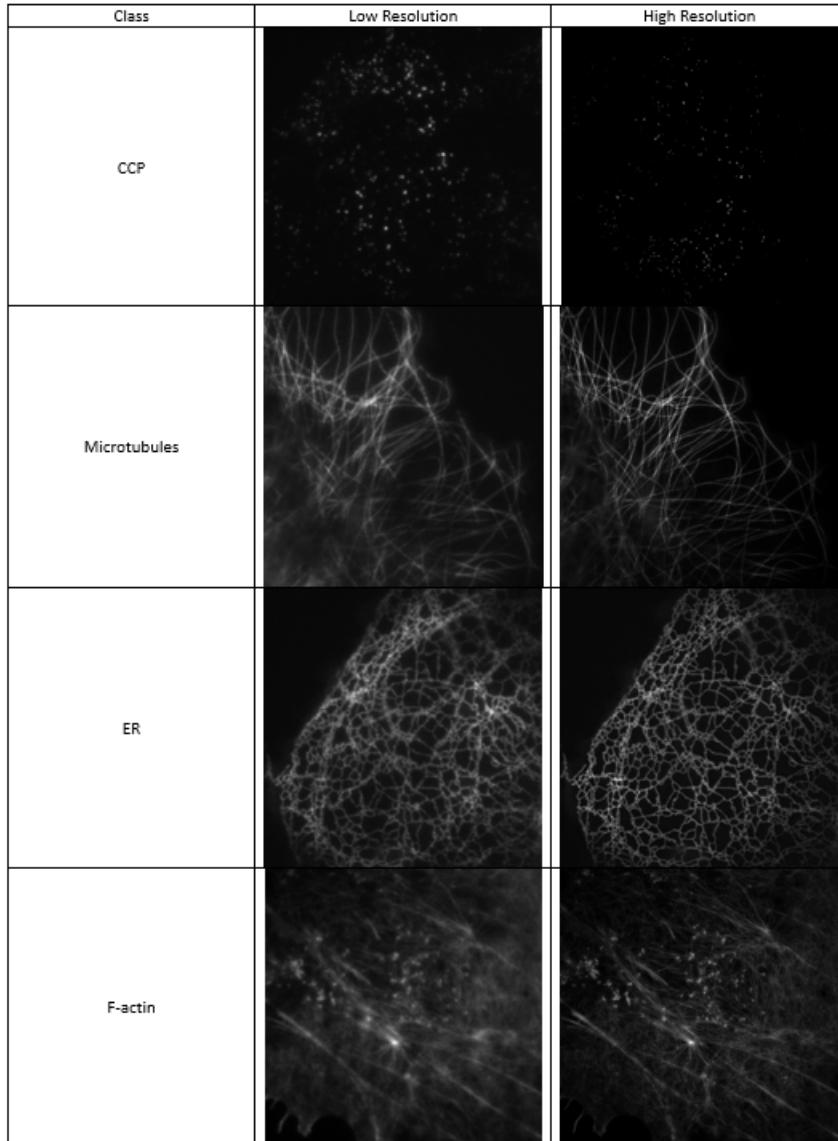
Background: There is a good amount of research already done, (and various in progress) on SRM. Few of the references we were able to find:

1. The ThunderSTORM (Ovesný et al. 2014) research paper is focused on single-molecule localization microscopy (SMLM) methods, such as PALM and STORM, which enable super-resolution imaging by pinpointing individual molecules with high precision. ThunderSTORM itself is an open-source ImageJ plugin designed to process and analyze data from these imaging methods. The software offers comprehensive tools for localization, data filtering, visualization, and even simulating realistic datasets for testing algorithms.
2. The DeepSTORM (Nehme et al. 2018) uses a encoder decoder convolutional model. It relies on CNN to reconstruct high-resolution images from stochastically-blinking emitters, such as fluorescent molecules. It stands out because it's fast, precise, and parameter-free, that means it doesn't require expertise to adjust settings for optimal results, making it accessible to a wider range of users.
3. We also came across a model that uses a transformer based approach (Verma et al. 2023) with four additional optimization techniques to improve the model reconstruction capability, namely Locality Sensitive Attention (LSA), Shifted Patch Tokenization (SPT), Rotary Position Embeddings (RoPE) and adversarial loss function inspired from Generative Adversarial Networks (GANs). LSA, SPT and RoPE enable the transformer to learn from the dataset more efficiently, while the adversarial loss function enhances the resolution of the reconstructed images.
4. The next paper dives into a recent trend of diffusion models, which have gained traction in generative AI, to generate super-resolution images (Saguy et al. 2024). The approach in the paper is to train these models on existing experimental datasets from sources like the Shareloc database. The trained models are then able to generate images that are visually comparable to those produced by actual experiments, potentially offering a new way to create high-resolution microscopy data without needing to perform costly experiments. This can be especially useful for training purposes and data augmentation in super-resolution microscopy.
5. Another short detailed overview of what is Super-resolution Microscopy (SRM) is detailed in this (Schermelleh et al. 2019) paper. As stated in the paper, super-resolution microscopy is a groundbreaking imaging technique that overcomes the diffraction limit of light microscopy, allowing scientists to view cellular structures in unprecedented detail. This limit was historically seen as an obstacle, preventing researchers from imaging structures smaller than 250 nanometers. SRM enables visualization at the nanoscale, revealing details such as protein interactions and cellular dynamics that are crucial for biological discoveries.

In the past, there were various methods that were used to overcome this obstacle, but now, with the invention of state-of-the-class models like diffusion models (which are used in generative AI), these models might be able to outperform their predecessors. So, we propose to use diffusion models for super-resolution microscopy.

2 Datasets

We plan to use BioSR, which is a biological image dataset for super-resolution microscopy, including more than 2200 pairs of low-and-high resolution images covering four biology structures (CCPs, ER, MTs, F-actin), nine signal levels (15-600 average photon count). The technique used to capture the images is called Structured illumination microscopy (SIM). It is based on the excitation of the sample with a known spatially structured pattern of light and relies on the generation of interference patterns known as Moiré effect. Different images are acquired, and by mathematically deconvolving the interference signal, a super-resolution image is obtained.



Sample Images from the BioSR dataset

The dataset includes the following biological structures-

- **Clathrin-Coated Pits (CCPs):** Specialized regions of the cell membrane involved in endocytosis, where clathrin proteins form a lattice-like coating.
- **Endoplasmic Reticulum (ER):** A network of membranous tubules within cells responsible for protein and lipid synthesis.
- **Microtubules:** Cylindrical structures made of tubulin that form part of the cytoskeleton and are essential for cell shape, division, and intracellular transport.

- **F-actin:** Filamentous actin, a polymerized form of actin that contributes to the cell's structure and movement.
- **F-actin Nonlinear:** A variation of F-actin imaged using nonlinear SIM techniques, capturing higher-order structural details.

Data Format Conversion: The dataset initially provided images in .mrc format, which is commonly used in electron microscopy. However, these files are incompatible with most deep learning architectures. To address this, the images were converted into two formats:

- .png for training and inference with our models
- .tiff specifically used for evaluating the DFCAN baseline method

Dataset Organization and Processing: The dataset was organized and processed as follows:

- **ER:** Each low-resolution image (levels 1–6) had a corresponding ground truth high-resolution image, which were mapped one-to-one directly.
- **Other structures (CCPs, Microtubules, F-actin, and F-actin Nonlinear):** A single high-resolution image corresponded to multiple low-resolution images. These were paired accordingly for training.

Data Augmentation: Data augmentation techniques haven't been applied during training. However, we have considered several options:

- **Viable options:** Rotation and horizontal/vertical shifts, as they preserve the structure of the microscopy data.
- **Avoided techniques:** Cropping, extreme shifting, or distorting, as they might harm the model's ability to generalize to real microscopy images.

Additional Considerations

- **Resolution Levels:** The dataset provides multiple resolution levels for each structure, allowing for the study of super-resolution at various scales.
- **Noise Characteristics:** The low-resolution images contain realistic noise patterns typical of SIM imaging, providing a challenging and authentic scenario for super-resolution algorithms.
- **Metadata:** Each image is accompanied by metadata including imaging parameters, which could be utilized for more advanced model development or analysis.

3 Methods

3.1 Latent Diffusion Model

Super-resolution using Latent Diffusion Models (LDMs) is formulated as a conditional image generation problem. The objective is to learn a mapping $f : \mathcal{I}_{\text{low}} \rightarrow \mathcal{I}_{\text{high}}$, where \mathcal{I}_{low} represents the low-resolution image space, and $\mathcal{I}_{\text{high}}$ represents the high-resolution image space. The process involves three main steps: latent space encoding, the diffusion process, and decoding.

Mathematical Formulation and Architecture The low-resolution image $\mathbf{x}_{\text{low}} \in \mathcal{I}_{\text{low}}$ is encoded into the latent space $\mathbf{z}_{\text{low}} \in \mathcal{Z}$:

$$\mathbf{z}_{\text{low}} = E(\mathbf{x}_{\text{low}})$$

Here, $E(\cdot)$ is the encoder of a pre-trained Variational Autoencoder (VAE).

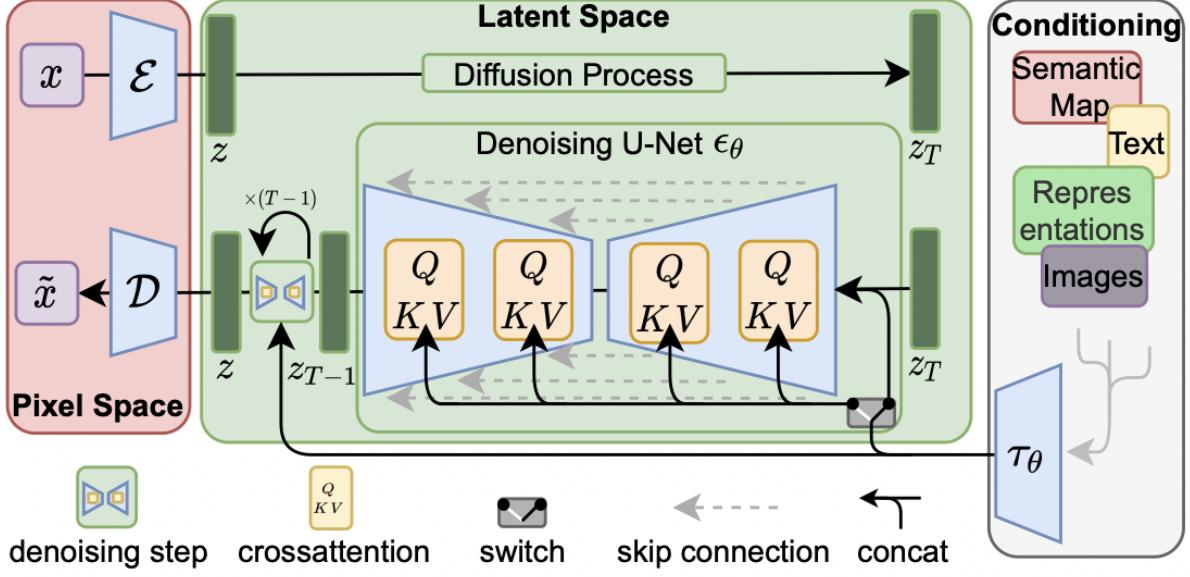


Figure 1: Latent Diffusion Model

Diffusion Process: The latent representation \mathbf{z} undergoes a forward diffusion process by progressively adding Gaussian noise at each timestep t :

$$q(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{\alpha_t} \mathbf{z}_{t-1}, (1 - \alpha_t) \mathbf{I})$$

During training, the reverse diffusion process is learned:

$$p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t, \mathbf{x}_{\text{low}}) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, \mathbf{x}_{\text{low}}, t), \sigma_t^2 \mathbf{I})$$

where μ_θ is the denoising network parameterized by θ .

Decoding: After reverse diffusion, the high-resolution latent representation \mathbf{z}_{high} is decoded into the high-resolution image:

$$\mathbf{x}_{\text{high}} = D(\mathbf{z}_{\text{high}})$$

Here, $D(\cdot)$ is the VAE decoder.

Model Architecture The model consists of the following components:

1. Variational Autoencoder (VAE):

- **Encoder:** Compresses the low-resolution input into a compact latent representation.
- **Decoder:** Reconstructs the high-resolution image from the latent space.

2. Diffusion Model in Latent Space:

- A UNet-based architecture is used as the denoising network.
- Cross-attention layers integrate features from the low-resolution image.

3.2 SR3: Image Super-Resolution via Iterative Refinement

SR3 (Super-Resolution via Repeated Refinement) (Chitwan Saharia 2021) is a novel approach to image super-resolution that adapts denoising diffusion probabilistic models to conditional image generation. This section outlines the key aspects of SR3.

Mathematical Formulation: SR3 adapts the denoising diffusion probabilistic model (DDPM) framework to conditional image generation. The key components of the mathematical formulation are:

1. **Conditional Distribution:** Given a dataset of input-output image pairs (x, y) , SR3 aims to learn a parametric approximation of the conditional distribution $p(y|x)$.
2. **Iterative Refinement Process:** The model generates a target image y in T refinement steps, starting from pure noise y_T and iteratively refining it to y_0 :

$$p_\theta(y_0|x) = \int p_\theta(y_{0:T}|x) dy_{1:T}$$

3. **Conditional Transition Distributions:** The refinement process is governed by learned conditional transition distributions:

$$p_\theta(y_{t-1}|y_t, x) = \mathcal{N}(y_{t-1}; \mu_\theta(y_t, t, x), \sigma_t^2 I)$$

4. **Training Objective:** The model is trained to minimize the variational lower bound:

$$L = \mathbb{E}_{q(y_{0:T}|y_0)} \left[-\log p(y_T) - \sum_{t=1}^T \log \frac{p_\theta(y_{t-1}|y_t, x)}{q(y_t|y_{t-1})} \right]$$

where $q(y_t|y_{t-1})$ is a forward process that gradually adds Gaussian noise to the target image.

Architecture: SR3 utilizes a modified U-Net architecture, which is trained to iteratively refine noisy images:

- The model starts with pure Gaussian noise and progressively denoises it to produce the final high-resolution image.
- The U-Net incorporates residual blocks inspired by BigGAN, replacing the original DDPM residual blocks.
- Skip connections in the U-Net are re-scaled by $\frac{1}{\sqrt{2}}$ to improve stability.
- The number of residual blocks and channel multipliers at different resolutions are increased compared to the original DDPM architecture.

To condition the model on the low-resolution input the input image is upsampled to the target resolution using bicubic interpolation and then the upsampled image is concatenated with the noisy image along the channel dimension.

4 Baseline Method: DFCAN/DFGAN (Chang Qiao 2021)

- DFCAN:

The architecture of the Deep Fourier Channel Attention Network (DFCAN) begins with a convolutional layer, followed by a Gaussian Error Linear Unit (GELU) activation function. The GELU activation function is defined as:

$$\text{GELU}(x) = 0.5x(1 + \text{erf}(x/\sqrt{2})) \quad (1)$$

where $\text{erf}(x)$ is the error function, formulated as:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (2)$$

The output of the GELU activation is fed into four identical residual groups (RGs). Each residual group consists of four Fourier Channel Attention Blocks (FCABs) and a skip connection. The operation of an RG is represented as:

$$\text{RG}(x) = x + \text{FCAB}^{(4)}(x) \quad (3)$$

The operation of the FCAB is defined recursively. For an input feature map x , the FCAB operation is given by:

$$\text{FCAB}^{(n)}(x) = \underbrace{\text{FCAB}(\text{FCAB}(\dots \text{FCAB}(x)))}_{n \times \text{FCAB}} \quad (4)$$

In each FCAB, feature maps are rescaled in a channel-wise manner as follows:

$$\text{FCAB}(x) = x + y \times f(W_U \delta(W_D \varphi(y))) \quad (5)$$

where:

$$y = \text{GELU}[\text{Conv}\{\text{GELU}[\text{Conv}\{x\}]\}] \quad (6)$$

$$\varphi(y) = \text{Pooling}_{\text{global}}\left(\text{ReLU}\left[\text{Conv}\left\{\text{abs}(\text{FFT}(y))^{\gamma}\right\}\right]\right) \quad (7)$$

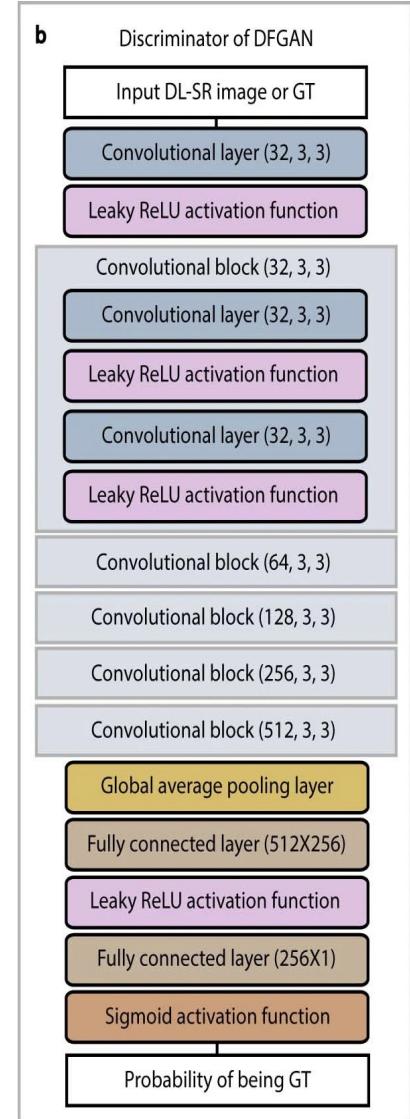
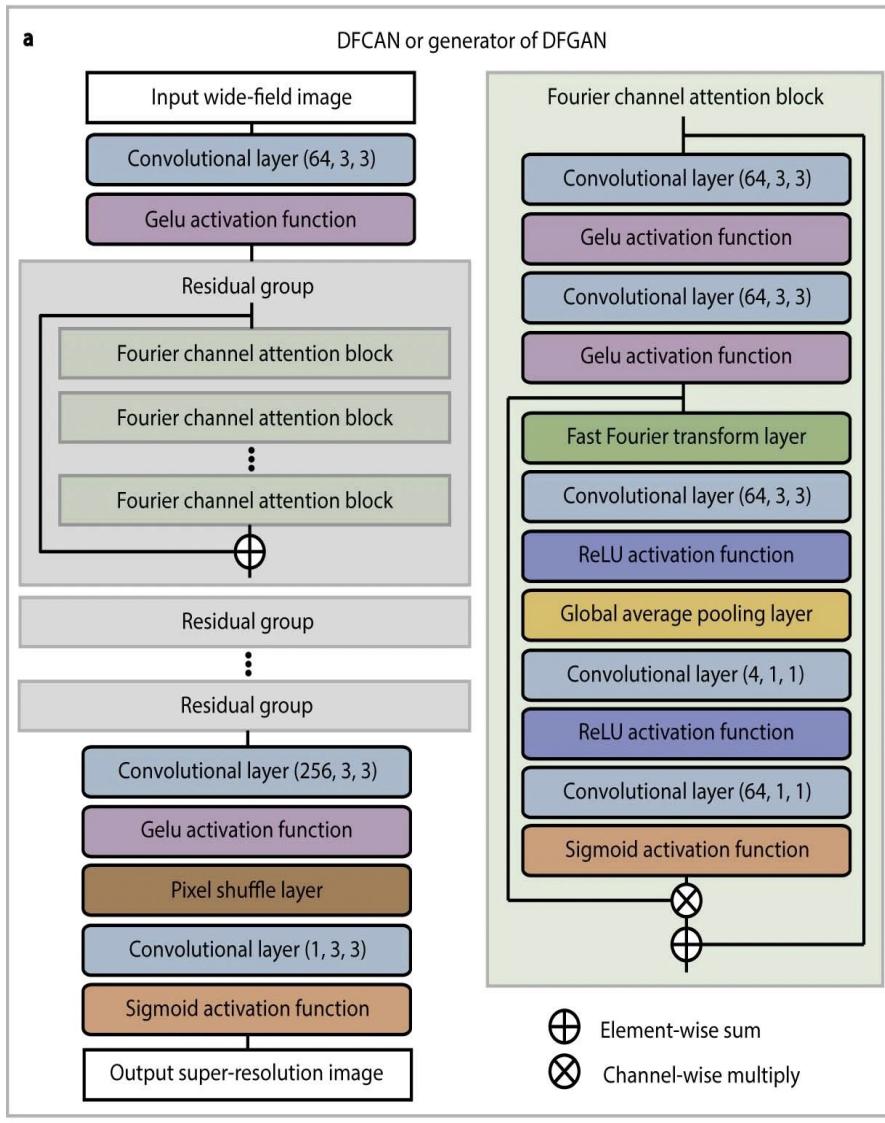
Here, $\text{FFT}(\cdot)$ is the Fast Fourier Transform applied to enhance the high-frequency components. γ is a scaling factor that enhances these contributions, and global average pooling (denoted as $\text{Pooling}_{\text{global}}$) is applied to compute a representative value for each feature map. W_D and W_U are the downscaling and upscaling weights, respectively, implemented via 1×1 convolutional layers. The functions $f(\cdot)$ and $\delta(\cdot)$ represent the sigmoid activation and ReLU activation functions, respectively, forming a gating mechanism to calculate the final rescaling factors adaptively.

After passing through the residual groups, the output is processed by a convolutional layer followed by a GELU activation. A pixel shuffle layer, a convolutional layer, and a sigmoid activation layer are then used to upscale the image to the same size as the ground truth (GT) image. The final output is a monochrome super-resolved (SR) image.

Loss Function: It is a combination of Mean Squared Error (MSE) loss and Structural Similarity Index (SSIM) loss. The MSE loss ensures pixel-wise accuracy, while SSIM improves the structural similarity of the output. The objective function is formulated as:

$$L_{\text{DFCAN}}(\hat{Y}, Y) = \frac{1}{w \times h} \sum_{i=1}^{w \times h} (\hat{Y}_i - Y_i)^2 + \lambda \times [1 - \text{SSIM}(\hat{Y}, Y)] \quad (8)$$

where \hat{Y} is the predicted SR image, Y is the ground truth image, and (w, h) are the pixel dimensions of the image. The scalar λ is a weight to balance the contributions of MSE and SSIM, typically set to 0.1.



(a) The architecture of DFCAN or the generator of DFGAN.

(b) The architecture of the discriminator of DFGAN.

- DFGAN: The DFGAN is constructed based on the conditional generative adversarial network (cGAN) framework, which consists of two models: one is the generative model G that learns the data distribution and performs the image transformation, the other is the discriminative model D that distinguishes if an image came from training data or was generated by the generator G. G takes low-resolution fluorescence images as input, and its output is an upscaled SR image. The discriminator D takes the output image from G or a GT SR image as input, and provides a score that reflects the probability of the input image being the GT. In the DFGAN framework, a deeper DFCAN acts as G, and D is constructed based on conventional CNN architecture. D consists of 12 convolutional layers, and the output of each convolutional layer is activated by a LeakyReLU activation function with a leaky factor of $\alpha=0.1$, which can be formulated as

$$\text{LeakyReLU}(x, \alpha) = \max(0, x) - \alpha \max(0, -x) \quad (9)$$

Then the output of the last LeakyReLU activation function is fed into a global pooling layer and two fully connected layers followed by a sigmoid activation function. After that, D outputs the estimated probability.

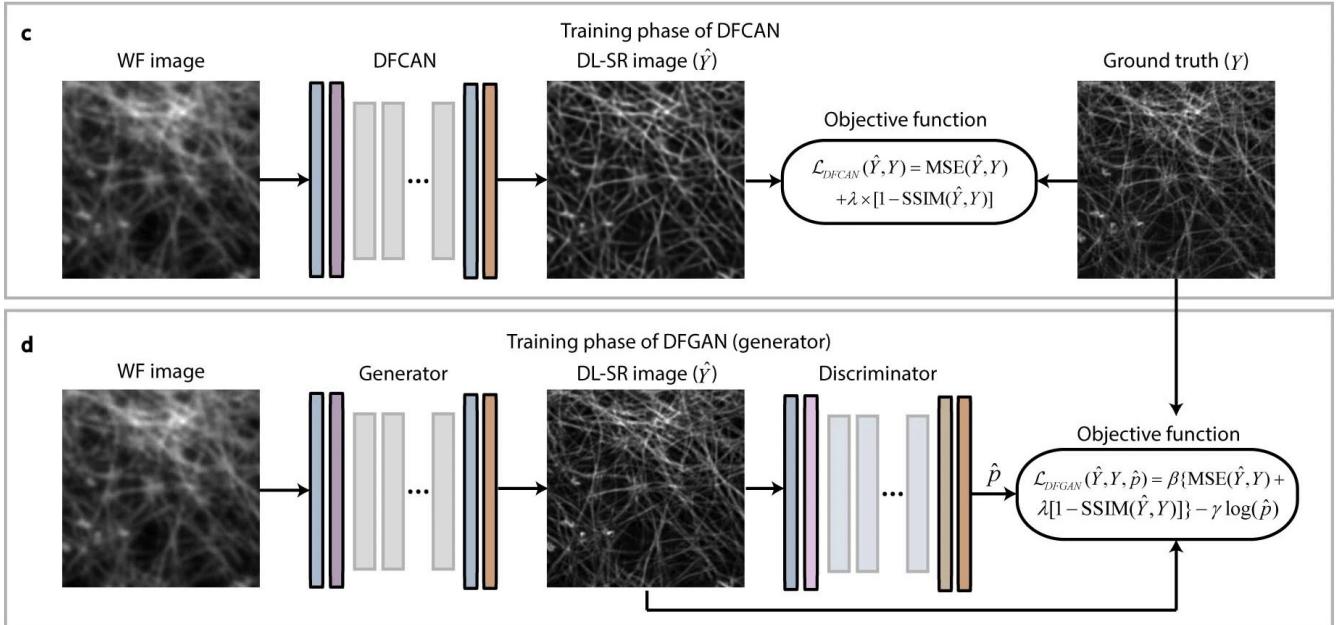


Figure 3: c. Schematic of the training process of DFCAN. d. Schematic of the training process of the generator of DFGAN. The generator and the discriminator were trained alternatively. The discriminator's parameters remained unchanged during each training step of the generator.

5 Results

5.1 Metrics

To evaluate and compare the results we used the following two metrics.

NMSE (Normalized Mean Squared Error): The Normalized Mean Squared Error (NMSE) is a measure of the difference between predicted and actual values. It is calculated as the mean squared error between the two sets of values, normalized by the variance of the actual values. NMSE is a widely used metric to evaluate the performance of predictive models, with lower values indicating better fit.

SSIM (Structural Similarity Index Measure): The Structural Similarity Index Measure (SSIM) is a perceptual metric that assesses the similarity between two images or signals. It considers luminance, contrast, and structural differences between the two, providing a more comprehensive evaluation of image quality than traditional metrics like PSNR. SSIM values range from -1 (worst) to 1 (best), with higher values indicating greater similarity between the two images or signals.

5.2 Baseline - DFCAN

Our analysis of the DFCAN model (**Baseline**) on the dataset produced the following results:

Class	NMSE	SSIM
Microtubules	13.560808225394004	0.22664751843245048
F-actin	14.469389400342473	0.22264938053330777

Table 1: NMSE and SSIM values for different classes on results obtained from DFCAN.

These metrics indicate the model's performance in terms of structural similarity index (SSIM), and normalized mean square error (NMSE). The results highlight the model's capability to enhance spatial resolution effectively.

F-actin prediction:

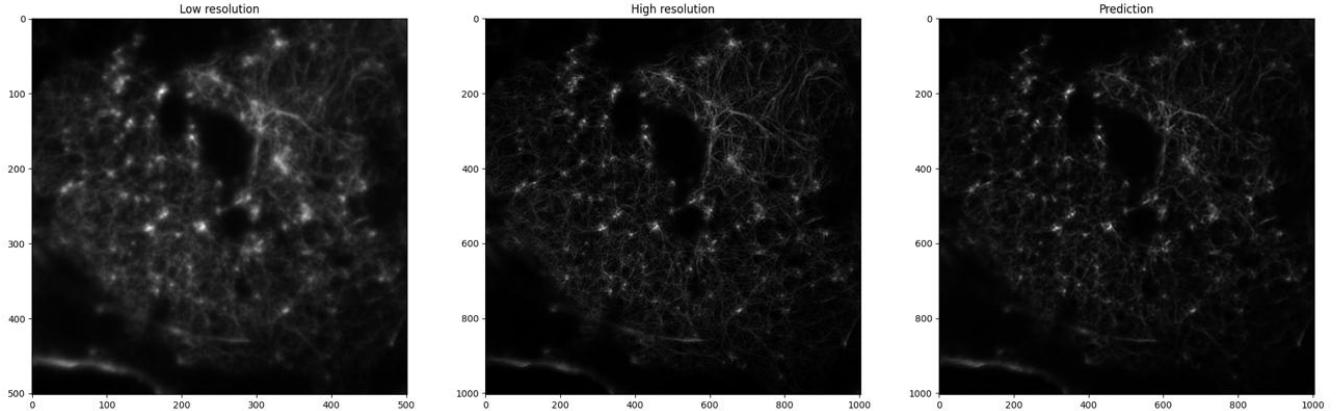


Figure 4: Sample image prediction using DFCAN - F-actin

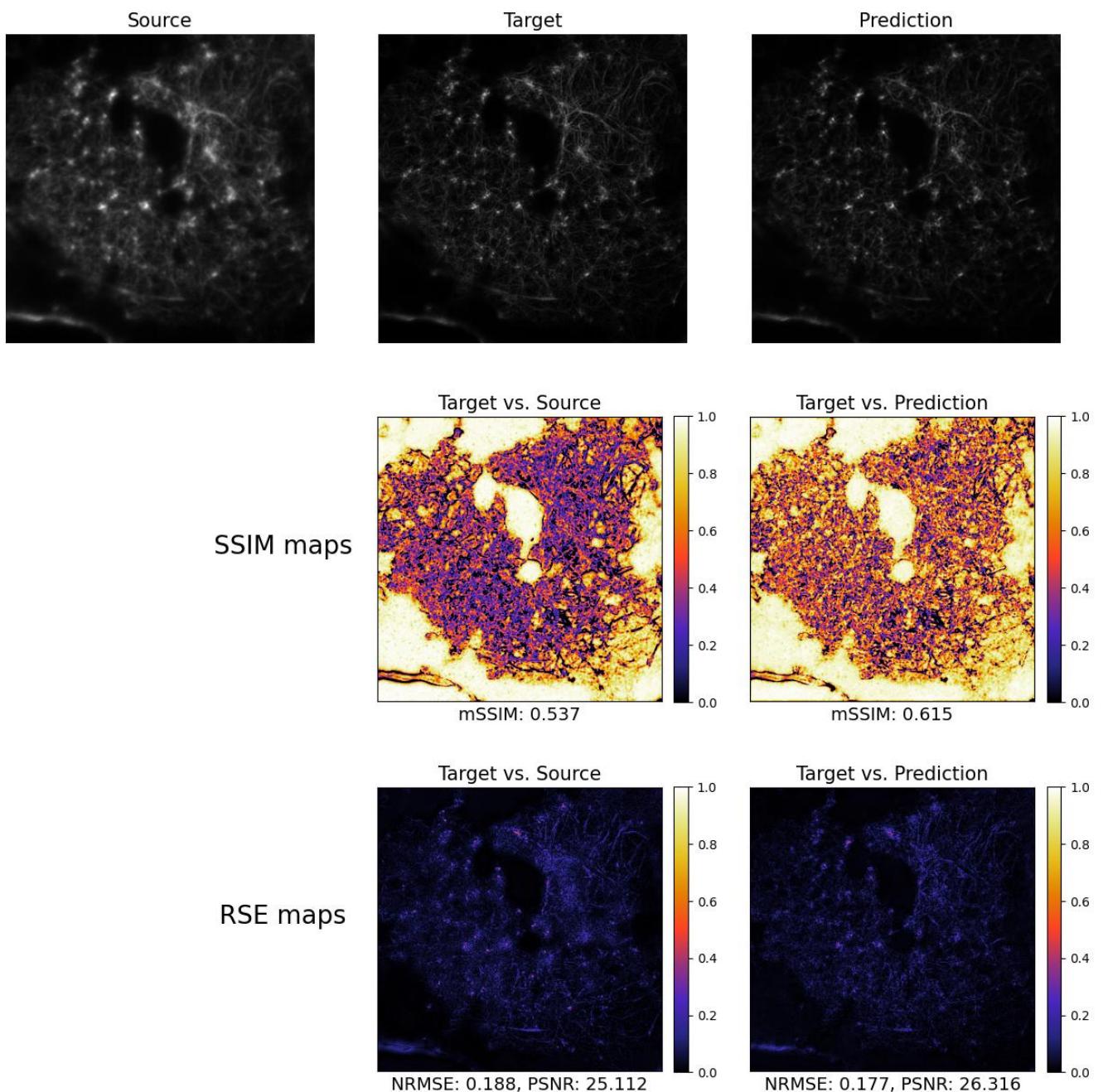


Figure 5: SSIM and RSE Map

Microtubule prediction:

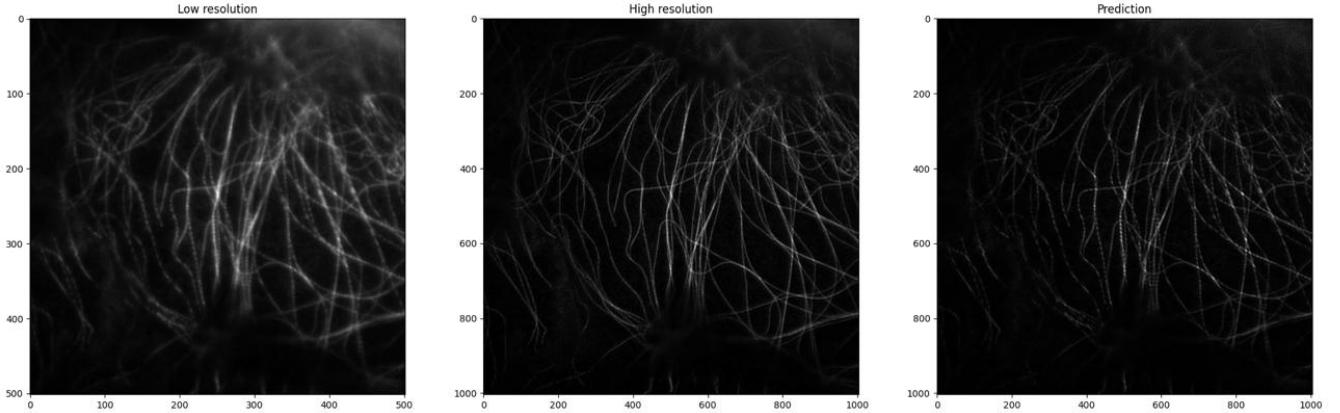


Figure 6: Sample image prediction using DFCAN for microtubules

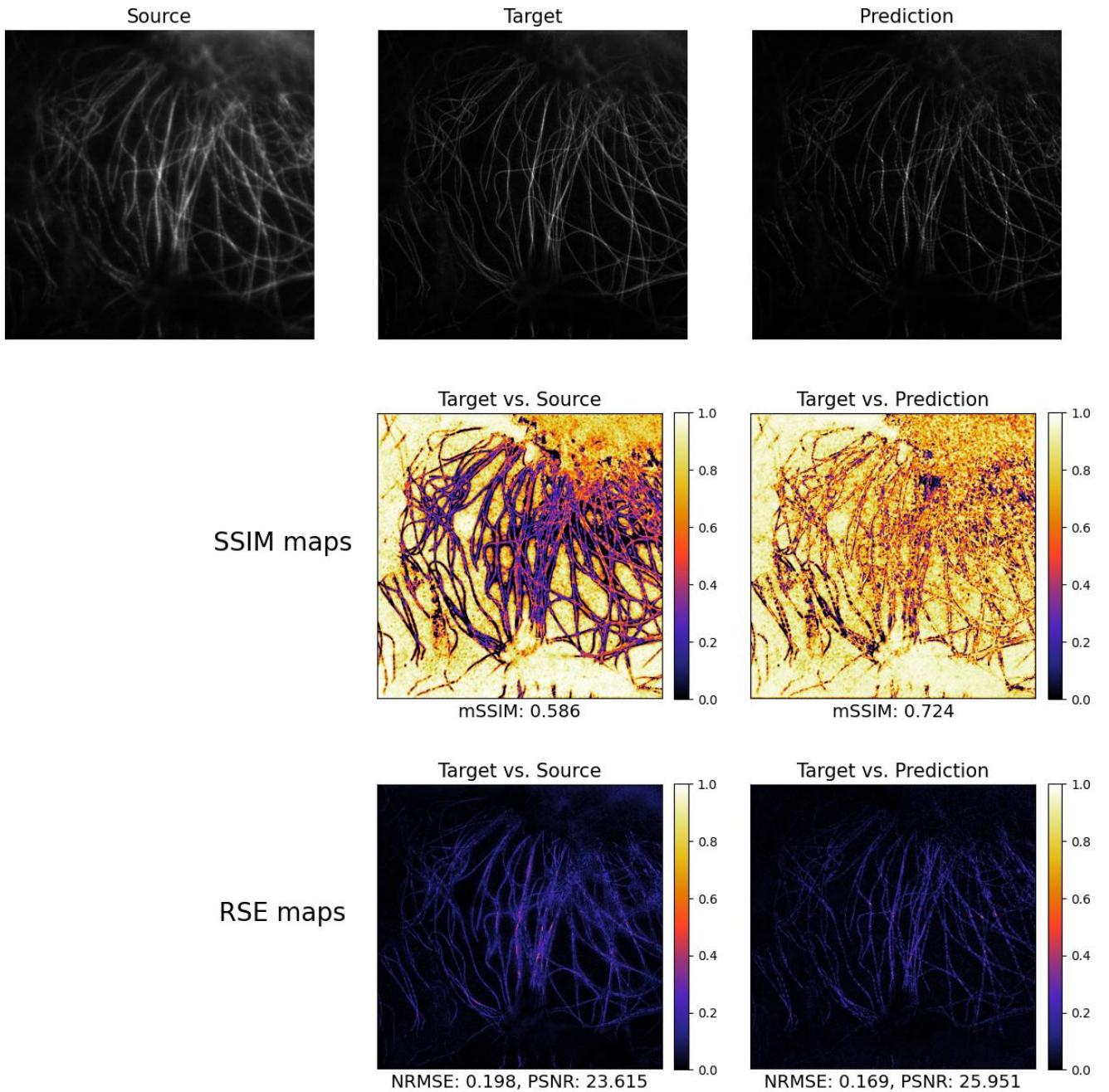


Figure 7: SSIM and RSE map for microtubules

In the following sections, we mostly demonstrate the results of our experiments using the SR3

architecture and compare them to the baseline DFCAN model. We provide an overview of the training process and discuss the validation results, highlighting the strengths and weaknesses of our approach. We also consider another model(LDM), which provides good results, but lacks technical documentation, which is considerable.

5.3 SR3 Model

5.3.1 Training the Model

We were able to successfully utilize the SR3 architecture provided in (Chitwan Saharia 2021) for inference and training. The model is computationally intensive, and so we had to use HPC (NVIDIA Tesla V100) to work with it. It takes about 3 minutes to process a single image for inference. The inference process for a sample image is summarized on image (Figure 9). We began by using a pre-trained model trained on human faces, which performed well after 830,000 iterations. We found that fine-tuning the model is better than training from scratch, even though the original dataset was completely unrelated to biomedical data. So we attempted to fine-tune the model on the BioSR dataset, which contains 5 different classes of medical structures. For fine-tuning, we used a V100, which took around 10-15 minutes to train for 1000 iterations.

First, we trained the model for 1000 and then 10,000 iterations on each dataset separately (The results for these were shown in the preliminary report). For 10,000 iterations, it took us about 2-3 hours per class. The results were not completely satisfactory (we hypothesize that this is because the original model was trained for more than 100,000 iterations, and we do not have that much resources at our disposal), so we proceeded by training on microtubule dataset for a total of 70,000 iterations, which took us more than 12 hours. Unfortunately, due to long queues for HPC access, computational complexity of the problem, and time constraints, we were not able to run for large number of iterations on other parts of the dataset.

In the next two sections we briefly discuss the results on the classes for which we trained for 10,000 iterations only and then provide more details about the microtubule class, for which we were able to produce meaningful results.

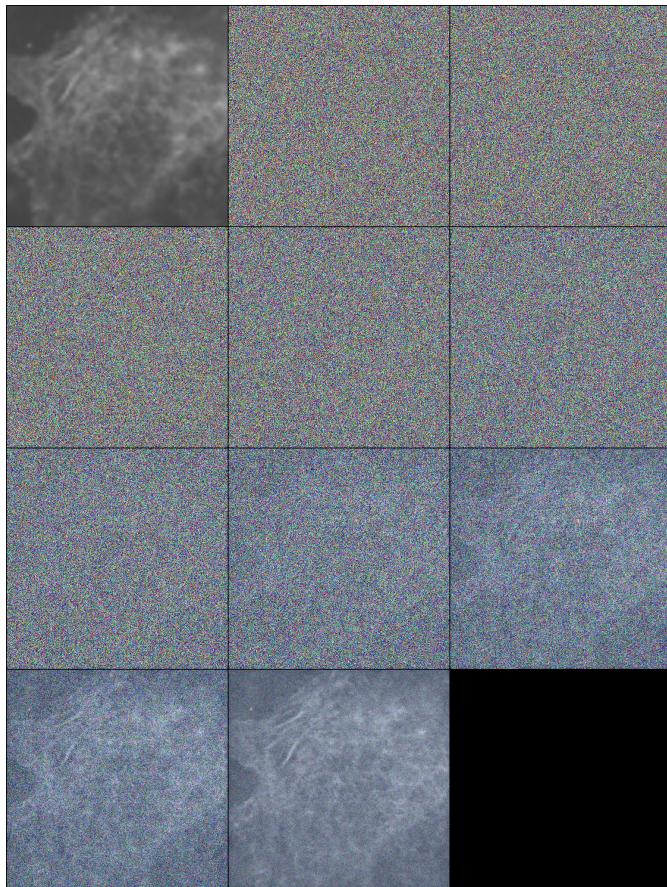


Figure 8: Inference Process by SR3 Visualized

5.3.2 Class-wise Performance

Class	NMSE	SSIM
F-actin	3.7685904868045	0.13768342285272
ER	7.022250348724631	0.03848880397692287
Microtubules	5.512801359815129	0.054930509608423966
F-actin (Non-Linear)	5.766438855683204	0.05123344513696936
CCP	9.8111849216556	0.0034405798969699

Table 2: NMSE and SSIM values for different classes on results obtained from SR3 model.

Initially, the best class to upscale was F-actin. We can compare the low-scale, high-scale (original), and prediction by the model (Figure 9). As we can see, the model clearly improves the quality of the low-scale image. Unfortunately, even though the results are somewhat meaningful, they are still worse compared to the baseline. The baseline results from the corresponding section also show the results for F-actin (Figure 4), so they can be compared directly. Both from the prediction images themselves, we can see that the baseline model performs better.

As we can see from the results, the hardest class to upscale was CCP. Let's consider the three different images we generated (Figure 10). The images were generated after three different training runs with different hyperparameters/subsets of the dataset. As we can see, the quality of the upscale is far from being perfect. The data is very noisy, and the data (mostly just points or very small circles) is hard to distinguish from the noise. Therefore, the model does not perform very well on this data. We would expect this class to be the hardest to upscale even for better models.

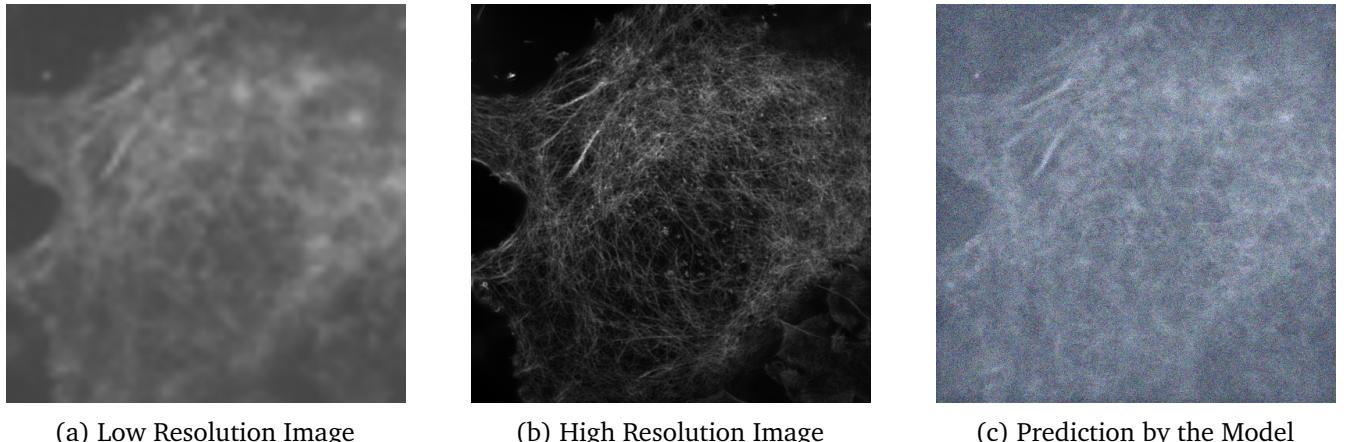


Figure 9: Predictions on F-actin by our Model (SR3)

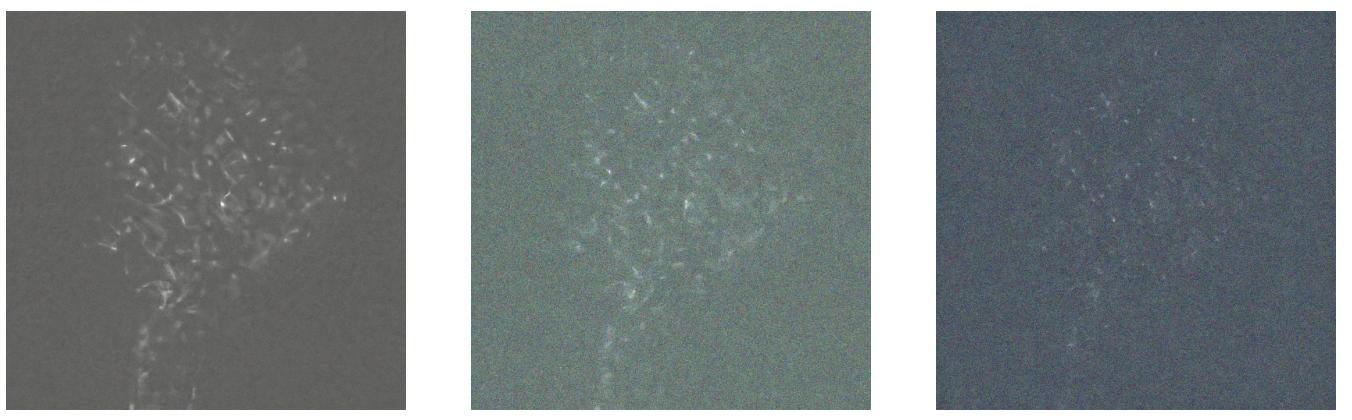


Figure 10: Several Different Predictions for the Same Image (CCP) after 3 Different Training Runs.

Table 3: Comparison of the SR3 model results after training for larger number of iterations

Iterations	NMSE	SSIM
20,000	0.5145	0.1379
40,000	0.3345	0.2531
60,000	0.2258	0.5839
70,000	0.0713	0.7920
Baseline (DFCAN)	13.5608	0.2266

5.3.3 Microtubule Results

As discussed earlier, the microtubule class was the only class we were able to train for 70,000 iterations. The other classes of data were trained for 10,000 iterations and we omit these results for the purpose of space.

The metrics for microtubule data, as well as comparison to baseline can be seen in table 3. Note that the results were evaluated on slightly different validation sets containing 3 images for SR3 and 5 images for DFCAN, and the results were averaged. Overall, we observe that after about 40,000 iterations, the SR3 model beats DFCAN in both metrics. As we train for more and more iterations, both metrics improve. For 70,000 iterations the model achieves very good results, and these results are unmatched by any models we tried throughout working on this project.

We also provide the comparison between sample images the models generated. Figure 11 shows the input, the desired output and the baseline prediction. Both SR3 and DFCAN were asked to generate a high-resolution image based on (a). The result generated by DFCAN is shown as (c) and the ground truth result (not provided to models) is shown as (b). The results for SR3 after it was trained for different number of iterations are shown in figure 12. As you can see, even the result for 40,000 iterations, although still very noisy, already beats DFCAN in terms of number of details, we can observe that the lines it generates are thinner and overall there are more lines and less blur. The results for 60,000 and 70,000 iterations are even better and provide a greater level of detail, with the result for 70,000 iterations being hardly distinguishable from the desired image from human perspective.

Even though the results show a lot of potential, it is hard to say how well the model would generalize on the outside data. The data we used for training and validation, although separated, was mostly from the same distribution, and the model may over-fit on this data. Measuring how well the model generalizes and improving generalization results could be a potential direction of further research.

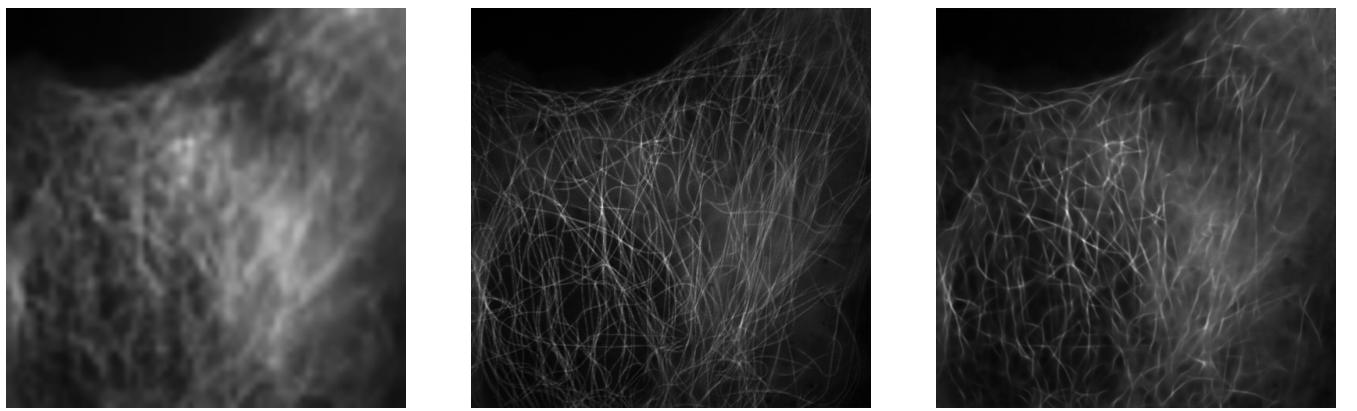
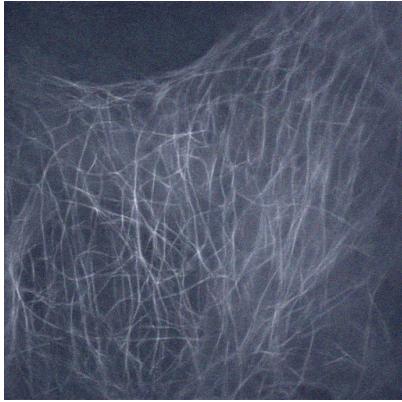


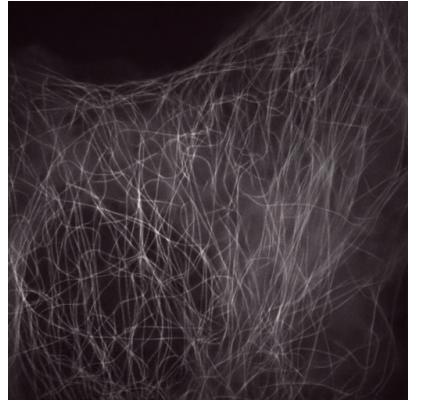
Figure 11: Input, Desired Output and Baseline Prediction



(a) SR3 prediction (40,000 iters)



(b) SR3 prediction (60,000 iters)

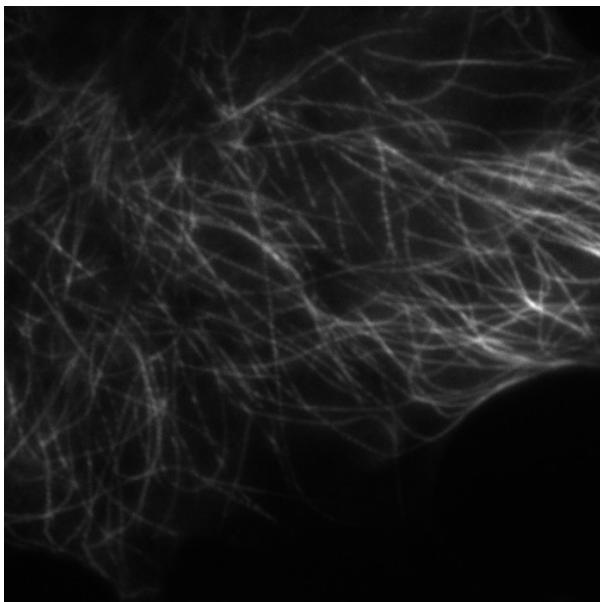


(c) SR3 prediction (70,000 iters)

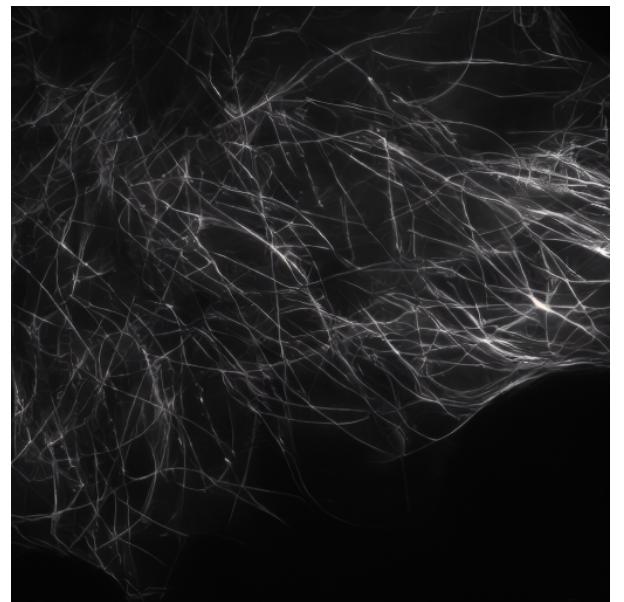
Figure 12: Predictions by SR3 after training for different number of iterations

5.4 Additional Results for LDM

Due to limited computational resources, we were unable to train the LDM model we identified. However, we would like to present the results it can achieve directly. Figures 13 and 14 illustrate the impressive quality of the results produced by the LDM model. From a human perspective, the quality appears to be exceptionally high, potentially matching or surpassing the baseline model. This observation is further supported by the numerical metrics provided in Table 4.

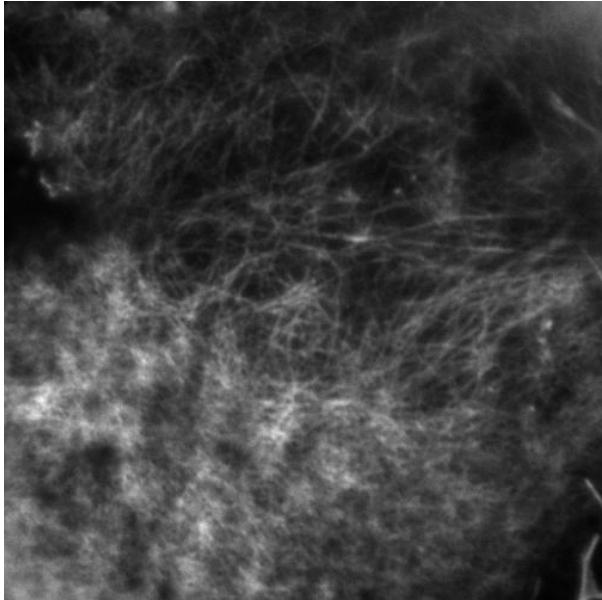


(a) Original Low resolution image

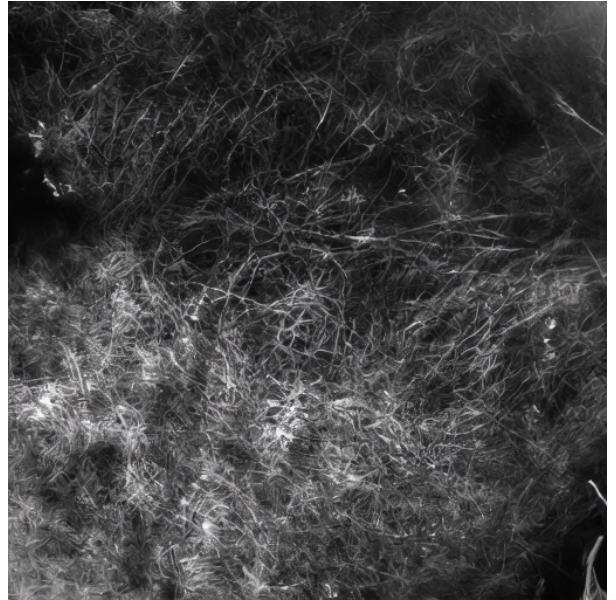


(b) LDM Generated image

Figure 13: Output from Latent Diffusion Model on a microtubule image



(a) Original Low Resolution Image



(b) LDM Generated image

Figure 14: Output from Latent Diffusion Model on a f-actin image

Class	NMSE	SSIM
Microtubules	0.029149479623279528	0.560779409352382
F-actin	0.04925276741200591	0.5293382668018352

Table 4: NMSE and SSIM values for different classes on results obtained from LDM.

6 Author contributions

	Shambavi Seth	Roman Vakhrushev	Aryan Prasad	Ishan Yadav
Conceptualization				
Formal Analysis				
Investigation				
Methodology				
Resources				
Software				
Report Writing				

Figure 15: Author Contributions

All the code for this project can be found [here](#).

References

- Breedijk, R. M. P. et al. (Mar. 2020). “A live-cell super-resolution technique demonstrated by imaging germinosomes in wild-type bacterial spores”. In: *Scientific Reports* 10.1. ISSN: 2045-2322. DOI: 10.1038/s41598-020-62377-1. URL: <http://dx.doi.org/10.1038/s41598-020-62377-1>.
- Chang Qiao Di Li, Y. G. C. L. T. J. Q. D. D. L. (2021). “Evaluation and development of deep neural networks for image super-resolution in optical microscopy”. In: *Nature methods*.
- Chitwan Saharia Jonathan Ho, W. C. T. S. D. J. F. M. N. (2021). “Image Super-Resolution via Iterative Refinement”. In: *arXiv:2104.07636v2*.

- Nehme, E. et al. (2018). “Deep-STORM: super-resolution single-molecule microscopy by deep learning”. In: *Optica* 5.4, pp. 458–464. DOI: 10.1364/OPTICA.5.000458. URL: <https://opg.optica.org/optica/abstract.cfm?URI=optica-5-4-458>.
- Ovesný, M. et al. (May 2014). “ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging”. In: *Bioinformatics* 30.16, pp. 2389–2390. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu202. eprint: https://academic.oup.com/bioinformatics/article-pdf/30/16/2389/48925507/bioinformatics_30_16_2389.pdf. URL: <https://doi.org/10.1093/bioinformatics/btu202>.
- Saguy, A. et al. (2024). “This microtubule does not exist: Super-resolution microscopy image generation by a diffusion model”. In: *bioRxiv*. DOI: 10.1101/2023.07.06.548004. eprint: [https://www.biorxiv.org/content/early/2024/05/15/2023.07.06.548004](https://www.biorxiv.org/content/early/2024/05/15/2023.07.06.548004.full.pdf).
- Schermelleh, L. et al. (2019). “Super-resolution microscopy demystified”. In: *Nature Cell Biology* 21.1, pp. 72–84. ISSN: 1476-4679. DOI: 10.1038/s41556-018-0251-8. URL: <https://doi.org/10.1038/s41556-018-0251-8>.
- Verma, N., D. Kaur, and L. Chau (2023). “Image Reconstruction using Enhanced Vision Transformer”. In: arXiv: 2307.05616 [cs.CV]. URL: <https://arxiv.org/abs/2307.05616>.