Denoising Diffusion Probabilistic Models for Synthetic Histopathologic Image Generation

Sunny Son
New York University
sons01@nyu.edu

Abstract

In this study, we showcase using denoising diffusion probabilistic models (DDPMs) to generate synthetic histopathologic (cross-sectional) images of cancerous and non-cancerous lymph nodes. Models were ablated through utilizing various combinations of attention block and time schedule mechanisms, then selected based on their structural similarity and log likelihood scores after being trained on the cancerous-only dataset. The model that utilized a cosine beta schedule with no self-attention mechanism performed best, and two models with said architecture were trained on respective cancerous/non-cancerous datasets. After training, pathologist evaluators were able to distinguish between pairs of generated histopathologic images originating from models trained on cancerous versus non-cancerous data, as well as giving slight pause in their ability to differentiate generated vs. ground-truth (dataset) images. This work highlights the potential of DDPMs to generate high-quality, distinguishable synthetic histopathologic images, contributing to the growth of data resources in the medical imaging domain.

1. Introduction

Histopathologic image analysis is essential for diagnosing and treating various diseases, including cancer; however, the collection and labeling of large-scale, high-quality data is both time-consuming and expensive, resulting in a lack of synthetic test data for developing and evaluating machine learning models [9]. In this study, we propose the use of denoising diffusion probabilistic models (DDPMs) [3, 7, 13, 19] for generating synthetic histopathologic images trained on the Patch Camelyon dataset [1, 23], as DDPMs have shown remarkable performance in producing high-quality images by training a denoising score matching objective [24] in a noise-to-image diffusion process [20]. Our goal is to leverage the powerful generative capabilities of DDPMs to create realistic and diverse synthetic histopathologic images, overcoming the limitations of existing data augmentation techniques [17]. GAN-based models have been made to generate histology patches, however these models have shown issues with overfitting of the discriminator [26].Our approach generates synthetic histopathologic images, which retain key features of the original dataset. Furthermore, previous attempts to leverage DDPMs do not compare the differences of ablated models [12,18]. These images, reviewed and deemed acceptable by pathologists, provide additional data for machine learning models. This increase in data could potentially improve model performance in diagnosing and classifying diseases from histopathologic images.

1.1. Code Repository

The repository to the notebooks and code used in this paper can be found here[1]. Please refer to the `README.md` for a more thorough breakdown of the structure of the notebook and samples of the reverse diffusion process unable to be demonstrated in the paper.

2. Method

A modified, simple (non-conditional) DDPM model was trained on histopathologic data for a limited number of epochs, followed by ablation for different combinations of noise schedule [7, 19] and self-attention [22] using Google Colab's NVIDIA A100 GPUs. The best architecture was then chosen through evaluation using Structural Similarity Index [25] and Maximum Likelihood [2, 21] (in practice, minimizing Log Likelihood). Two models of that architecture was then trained on a larger number of epochs[2]. 256 images of each class (cancerous and non-cancerous) were sampled from each respective model, and the best 16 (through naive visual inspection) taken from each of 256 samples (for each model trained on cancerous/non-cancerous class) to be evaluated by experienced visual in-

[1] https://github.com/sunnydigital/cv-f22
[2] For parallelization of training and increased speed of achieving results

spection where correctness of metastatic features in generated images.

2.1. Data

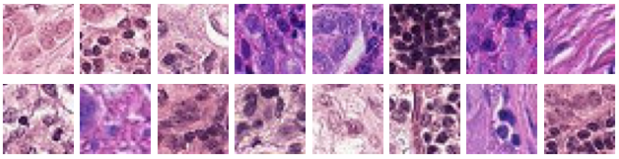

Figure 1. Sample cancerous (positive) Patch Camelyon images, also selected to be in the survey

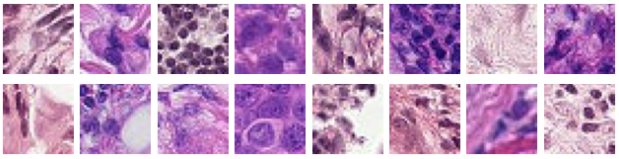

Figure 2. Sample non-cancerous (negative) Patch Camelyon images, also selected to be in the survey

The Kaggle version of the Patch Camelyon dataset [1,23] is a collection of 277483 96 × 96px histopathologic TIF images, split between `train` and `test` sets, with `train` having 220025 samples and `test` having 54758 samples. Given the unlabeled nature of `test`, it will be excluded from this analysis.

Of the `train` set, 89117 are metastatic identified (cancerous, positive) images and 130908 are non-metastatic identified (non-cancerous, negative) images. The images were ingested, then cropped to 32×32px at the center, due to the dataset being **only** identifiable as positive or negative classes for metastasis by a central 32×32px bounding box. A PyTorch Dataset object (HistoDataset) was then created to capture `train` and `eval` portions, with a `train`/`eval` split of 0.9/0.1 (of the original 220025 `train` samples) and a random seed to ensure reproductibility (of 12664675).

Given the nature of the DDPM to utilize pixel values between $[-1, 1]$ but TIF images, represented as PyTorch tensors are of range $[0, 1]$, an image transform function $[0, 1] \rightarrow [-1, 1]$ (and associated reverse transform $[-1, 1] \rightarrow [0, 1]$) were defined and made attributes of the HistoDataset class for viewing and visual evaluation purposes.

2.2. Implementation Sources

A simple DDPM model was modified, following an original implementation cited [14]. From the original, the parameter `attention_layers` was added to programatically specify which layers of the U-Net model class to have self-attention with necessary changes made, a modified self-attention block with a multi-headed attention layer obtained and modified as cited [15], and a custom `cosine_beta_schedule` function to return timestep values based on a cosine β schedule. Finally, everything is wrapped together under the class `DiffusionModel`, including code representing the forward (training) and reverse (sampling) diffusion process.

2.3. Mathematical Overview

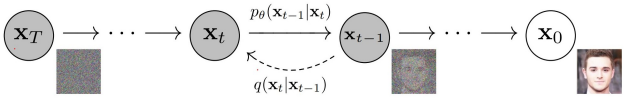

Figure 3. Forward and reverse diffusion processes from Ho et. al [7]

The Denoising Diffusion Probabilistic Models (DDPMs) are characterized by two key processes: the forward diffusion process and the reverse diffusion process.

The **forward process** corrupts the data $\mathbf{x}_0$ by adding noise iteratively over discrete time steps t in a Markov chain fashion following a β variance schedule $(0 < \beta_1 < \beta_2 < ... < \beta_T < 1)$. The process is defined by the transition distribution:

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

This is the same as stating that each new, noisier image is created by sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then updating $\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}$.

The **reverse process** denoises the corrupted data back to its original form. The process is defined by the conditional distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, parameterized by a neural network with parameters θ. Given a noisy image $\mathbf{x}_t$ at time step t, the network predicts the distribution of the denoised image $\mathbf{x}_{t-1}$ at the previous time step through sampling.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\right) \quad (2)$$

Here, $\mu_\theta(\mathbf{x}_t, t)$ and $\Sigma_\theta(\mathbf{x}_t, t)$ are the mean and variance predicted by the neural network, respectively. The objective of training is to learn the parameters θ that maximize the likelihood (minimize the log likelihood) of the data under this reverse process.

2.4. Training

Following the original authors' methods [7,19], the training process for a DDPM approximates the forward diffusion

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \text{Uniform}(\{1, ..., T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)||^2$
until converged

process by adding Gaussian noise to input images, transforming the distribution of the data into an isotropic Gaussian iteratively over a predefined number of timesteps (defined to be $T = 1000$ for all trained models). The amount of noise added is planned according to a variance schedule $\beta_1, ..., \beta_T$ (specifics defined later).

The DDPM predominantly learns the mean μ_θ of the conditional probability distribution, with the variance being kept fixed and untrained over time (according to the β schedule). As outlined in the original work [19], we set $\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I}$, where $\sigma_t^2 = \beta_t$.

The model's objective function is derived by recognizing the combination of q (forward process) and p_θ (reverse process) as a variational autoencoder (VAE [8]), utilizing the variational lower bound (evidence lower bound ELBO) to minimize the negative log-likelihood with respect to the ground truth data sample $\mathbf{x}_0$. This results in a sum of losses at each time step t, denoted as:

$$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1 + ... + \mathcal{L}_T \tag{3}$$

In this framework, each term of the loss (excluding L_0) is the Kullback-Leibler divergence between two Gaussian distributions, simplifying to an L2-loss with respect to the means. A direct outcome of the forward process q is the ability to sample $\mathbf{x}_t$ at any arbitrary noise level conditioned on $\mathbf{x}_0$, due to the sum of Gaussians being Gaussian. This can be described by:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{4}$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \sum_{s=1}^{t} \alpha_s$. This property allows for direct sampling of $\mathbf{x}_t$ by adding appropriately scaled Gaussian noise to $\mathbf{x}_0$. Consequently, it is possible to optimize random terms of the loss function $\mathcal{L}$ during training, enabling random sampling of t and optimization of $\mathcal{L}_t$, and as shown in Ho et al. [7], a reparameterization trick can be applied to the mean to induce a neural network to learn the added noise (using the network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$) for noise level t in the KL terms that constitute the losses, making the network a *noise* predictor rather than a *mean* predictor, computed as follows:

$$\mathcal{L}_t = \frac{1}{2\sigma_t^2}\left|\left|\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) - \mu_\theta(\mathbf{x}_t, t)\right|\right|^2 \tag{5}$$

Noting the similarity in the expanded definition of $\mu_\theta(\mathbf{x}, t)$:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, \mathrm{t})\right) \tag{6}$$

The equation can be written as:

$$\mathcal{L}_t = \frac{1}{2\sigma_t^2}\left|\left|\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, \mathrm{t})\right)\right|\right|^2 \tag{7}$$

and simplifies to:

$$\mathcal{L}_t = \frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathrm{t})\right|\right|^2 \tag{8}$$

Ho et. al found that removing the coefficient proceeding the difference of measured and predicted noise improved performance and optimized for speed [5]. This would finally imply the below $\mathcal{L}_{\text{simple}}$:

$$\begin{aligned}\mathcal{L}_{\text{simple}} &= \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathrm{t})\right|\right|^2\right] \\ &= \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\left|\left|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \mathrm{t})\right|\right|^2\right]\end{aligned} \tag{9}$$

Algorithm 1 Training describes this process, sampling a random $\mathbf{x}_0$ from the data distribution $q(\mathbf{x}_0)$. Then, a random timestep t is selected (between 1 and T). Noise is sampled from a Gaussian distribution, corrupting the image through the forward diffusion process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. Finally, the neural network $\epsilon_\theta(\mathbf{x}_t, \mathrm{t})$ is trained to predict the added noise given a noisy image $\mathbf{x}_t$.

2.4.1 Noise Schedule

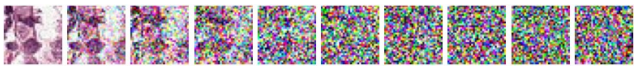

Figure 4. Forward diffusion process for **linear** β schedule

In the original DDPM paper [7], a **linear** β **schedule** was used, specifying β values between $\beta_1 = 10^{-4}$ and $\beta_T = 0.02$. Associated α_t and $\bar{\alpha}_t$ values would be derived from a respective β_t value.

Figure 5. Forward diffusion process for **cosine** β schedule

In the improved version by Nichol & Dhariwal [13], a **cosine β schedule** was used, derived in terms of $\bar{\alpha}_t$:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = cos\left(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2}\right)^2 \tag{10}$$

where s is a small value to prevent singularities (set to 0.008 in the paper), and β_t to be no larger than 0.999 to prevent the same problem nearing the end of the diffusion process where $t = T$. Note the more even addition of noise in Figuree 4 vs. the abrupt corruption of the image in Figure 5.

Utilizing the identity $\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}$, we can then use the standard method of re-deriving α_t, $\bar{\alpha}_t$, and $1 - \bar{\alpha}_t$ (we find βs for computational simplicity in an already structured model). This meant for improved efficiency in the reverse diffusion process, as well as a more even addition of noise (in terms of $\bar{\alpha}_t$) between steps.

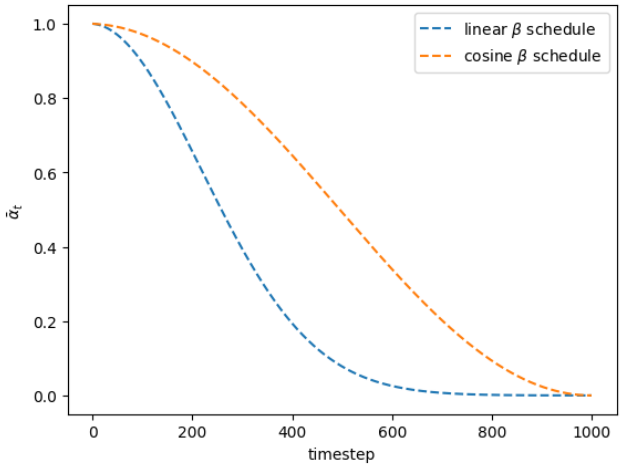

Figure 6. $\bar{\alpha}_t$ in diffusion with linear schedule vs. cosine schedule

As shown in Figure 4 (also in the original DDPM paper [7]), the utilization of a cosine β schedule evens out the corruption of images through timesteps, having less of a steep drop off toward the beginning of the process. This can be visually confirmed as well, comparing Figure 2 and Figure 3, seeing the increased evenness of noise added using a cosine β schedule.

2.5. Neural Network Architecture

For the Neural Network used in the implementation of this DDPM, we used a U-Net [16] architecture, due to it performing well with medical image segmentation at the time. Residual connections (ResNet Blocks [5]) are implemented in this architecture, improving gradient flow and allowing the training of models with more layers. Most importantly, it is the architecture best used to approximate denoising diffusion processes.

2.5.1 Positional (Time) Encoding

The standard Sinusoidal Positional Encoding from the 2017 "Attention is All You Need" [22] paper is utilized, encoding the timestep in the standard 512 time dimensions as stated by the authors. This has the advantage of capturing temporal dependencies between images, especially important given the sequential nature of the training process.

2.5.2 Base Block

The Base Block integrates time data into spatial features via a transformation of the time embedding, which is subsequently added to the input features. This is followed by a 2D convolutional layer for local feature extraction and a 2D batch normalization layer for output normalization, thereby enhancing model stability and training efficiency. The inclusion of a ReLU activation function imparts non-linearity, boosting the model's capacity to identify complex patterns, and the process repeats once, yielding final output features enriched with high-level information.

2.5.3 Encoder Block

The Encoder Block model handles feature extraction and downsampling. This begins with the input passing through a Base Block, where time-embedding is incorporated into the spatial features and a series of transformations including convolution, batch normalization, and activation are applied. These operations extract essential spatial features, normalize the output across the batch, and introduce non-linearity. Lastly, a max pooling operation is utilized for downsampling, halving the spatial dimensions by preserving the maximum value in each local 2x2 area. The resulting output includes downsampled feature maps and the pre-downsampling original feature maps, which are reserved for residual connections at the same resolutions in Decoder Blocks.

2.5.4 Decoder Block

The Decoder Block upsamples and refines the features learned by the the model. It employs a transposed convolution operation, also known as deconvolution, on the input to effectively double the spatial dimensions. Following this, the upsampled feature maps are concatenated with the corresponding residual resolutions from the encoder, reintroducing high-frequency details potentially lost during downsampling. These combined feature maps are then processed by the Base Block, which integrates time-embedding and applies a series of transformations such as convolution, batch normalization, and activation. The resulting output is an enhanced, upsampled version of the input features.

2.5.5 Self-Attention

The Self-Attention Block constitutes a block with a multi-head attention module that enables simultaneous processing of various aspects of the input. Layer normalization is applied to standardize the features, enhancing training stability and speed. Attention values are further processed by a feed-forward network with linear transformations and a Gaussian Error Linear Units (GELU) activation function to learn complex patterns. The final output, matching the original input dimensions, combines the feed-forward network output with the original attention values, ensuring compatibility with the model's subsequent operations.

Self-Attention can be applied to any layer, however in our implementation we limit attention layers to both encoder and decoder output resolutions of 128 and 512 (4 layers in total).

2.5.6 Complete U-Net

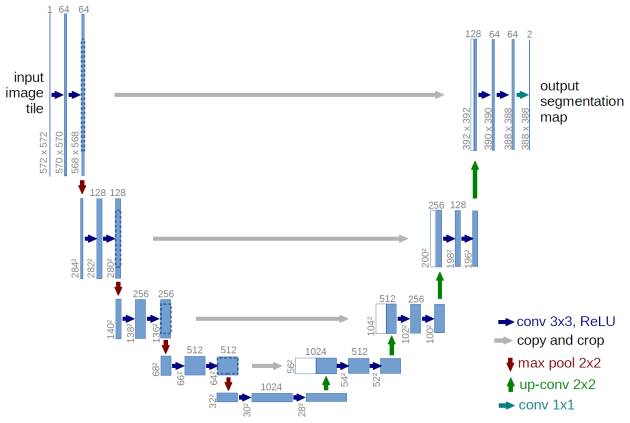

Figure 7. U-Net architecture from the original paper

The implemented U-Net architecture starts with a time-embedding layer that transforms a temporal feature into a high-dimensional representation. The model is then composed of an encoder-decoder structure with self-attention mechanisms incorporated at various layers based on a specified configuration, itself dependent on an input parameter. Each Encoder Block within the structure processes the input, returning a residual connection from that resolution, followed by a downscaling operation, and applies a self-attention mechanism if specified. A bottleneck Base Block processes the encoded features before passing to the decoder portion of the model. The Decoder Blocks upscale the features and combines the upsampled features with corresponding residual Encoder Block features, finally applying self-attention mechanisms to that layer, if prompted. The final output is generated using a 2D convolution layer, producing a feature map of the same size as the initial input.

The resolutions of the network closely resemble that in Figure 7, with an input channel of 3 and a time embedding dimension, upsampled to 1024 through powers of 2 using Encoder Blocks (a residual connection made at every resolution except the input 3), a bottleneck from 1024 to 512 (without residual connection), and downsampling using Decoder Blocks in powers of 2 at the same resolutions as were upsampled, each accepting respectively resolutioned residual connections, to finally output the original 3 channels of information.

2.6. Sampling

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, ..., 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

To draw generated samples, **Algorithm 2** Sampling was followed, however in our case, we used the identity $\beta_t = 1 - \alpha_t$ for easier implementation. With all else the same, first a noisy image $\mathbf{x}_T$ is drawn from the isotropic distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, for *reversed* timesteps $T, ..., 1$ another variable $\mathbf{z}$ is drawn from an isotropic distribution (if t is not 1), ensuring that at every timestep except the last, our model will *predict* the difference in noise level (in this case $\mathbf{x}_t$ subtracted by a cumulative product function of the current noise schedule β_t, multiplied by the noise predicted by our model, $\epsilon_\theta(\mathbf{x}_t, t)$, added to the scheduled variance times the drawn variable, $\sigma_t \mathbf{z}$. In the last timestep (when $t = 1$), $\mathbf{z} = 0$, implying no added noise.

3. Model Selection

After building a model robust to different combinations of different parameters, an ablation study is performed on every model with a different combination of parameters after training for a limited number of epochs, and evaluated against benchmarks of Structural Similarity Index [25] and Maximum Likelihood [2, 21].

3.1. Ablation Study

Four models were initially trained for 100 epochs over the cancerous (positive class) dataset, each having one of a different combination of schedule and attention mechanism. The below table represents this:

Table 1. Schedule/Attention Combinations

Schedule	Attention
cosine	none
cosine	self
linear	none
linear	self

The loss for every model was very similar, summarized in the below graph:

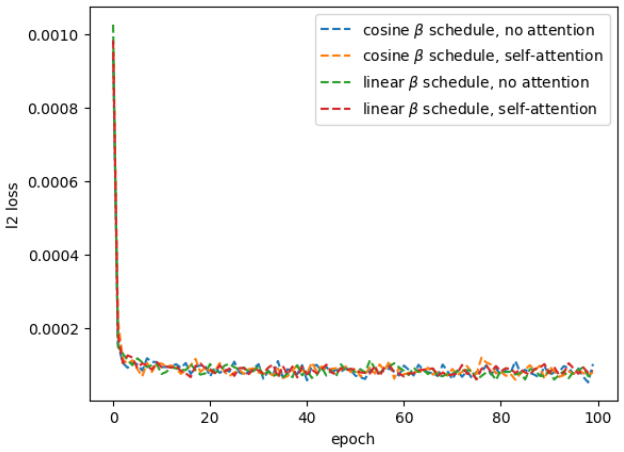

Figure 8. Training loss of each of four potential combinations of schedule/attention mechanism

To evaluate, for each trained model over 5 splits of the `eval_dataset`, SSIM and Maximum Likelihood were obtained over the data and averaged for each model.

3.1.1 Structural Similarity

SSIM was determined to be a viable evaluation metric for a DDPM in this course of experiment, given the unavailability of contemporary such as Inception Score (IS) [21] and Fréchet Inception Distance (FID) [6], commonly used to evaluate generative models, mostly due to the underlying Inception model being trained on CIFAR, and a benchmark model for the Patch Camelyon dataset was unable to be found.

The proposed metric focuses on three distinct features *luminance*, *contrast*, and *structure*, where:

- **luminance**: comparing between two images X and Y, for pixel values x and y we determine the mean brightness for each using the equation:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{11}$$

(respectively the same for y values), the luminance:

$$l(\mathbf{x},\mathbf{y}) = \frac{x\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},\ \ C_1 = (LK_1)^2 \tag{12}$$

- **contrast**: comparing between two images X and Y, for pixel values x and y we determine the standard deviation brightness for each using the equation:

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}} \tag{13}$$

(respectively the same for y values), the contrast:

$$c(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_x^2 + C_2},\ \ C_2 = (LK_2)^2 \tag{14}$$

where $L = 255$ or the dynamic range of RGB pixel values

- **similarity**: comparing between two images X and Y for pixel values x and y, the similarity is defined as

$$s(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{15}$$

where

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{16}$$

and C_3 to be defined later on

Finally,

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha}\cdot[c(\mathbf{x},\mathbf{y})]^{\beta}\cdot[s(\mathbf{x},\mathbf{y})]^{\gamma} \tag{17}$$

where $\alpha > 0, \beta > 0, \gamma > 0$

Simplifying by setting $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$, we obtain:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{18}$$

The rationale is that, while generated images may differ from those in the evaluation set, histopathological scan images exhibit considerable visual similarity in terms of luminance, contrast, and structure as a *class*. This similarity is sufficient to allow discernable differences between different generative model architectures to be identified and analyzed, over a large enough sample, over a large enough number of splits.

3.1.2 Maximum Likelihood

Originally used to evaluate GANs [4], Maximum Likelihood (in a more complicated form [2]) is now being applied to a diverse group of generative models. In practice, the log likelihood is calculated between generated and evaluation set images following the equation:

$$\text{BCE} = -\sum_{s,b} [y \times \log(p) + (1-y) \times \log(1-p)] \quad (19)$$

where $s =$ all samples and $b =$ all batches.

3.2. Ablation Results

After training every model (each a combination of linear/cosine schedule and self/no-attention) for 100 epochs, the results of the ablation study for the metrics SSIM and Maximum Likelihood are as follows:

Table 2. Average SSIM & Maximum Likelihood Results

Sched.	Attn.	SSIM (↑)	Log. Likelihood (↓)
linear	none	0.00013885	0.01569078
linear	self	0.00010798	0.01568433
cosine	none	0.00026722	0.01448672
cosine	self	0.00015712	0.01710702

With an average SSIM of 0.00026722 and average Log Likelihood of 0.01448672, the model with attributes of a cosine β schedule and no self-attention mechanism is chosen to be trained further, due to its' benchmarked performance being best.

4. Experiment

Taking the chosen architecture for each model, two are created, one for positive (cancerous) and one for negative (non-cancerous) class images, and trained for 200 total epochs. 256 images are generated from each model (Fig.11, Fig.12), with a simple visual inspection used to determine the best, most suitable 16 from each (32 total). It must be noted that the greater presence of fat cells in non-cancerous (negative class) images may be the cause of many generated images to be presented as white. From a biological perspective, because the Patch Camelyon dataset only considers Lymphomas, this makes sense as negative classes with central bounding boxes of 96×96px would me more likely to not contain tissue cells altogether. Initial visual inspection would allow for the selection of a *balanced* cohort of images to be presented for pathologist evaluation.

4.1. Survey

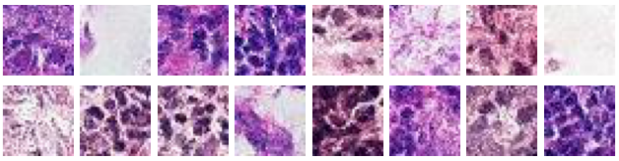

Figure 9. 16 cancerous (positive class) images selected from the generated batch of 256 to be used in the survey

Figure 10. 16 non-cancerous (negative class) images selected from the generated batch of 256 to be used in the survey

A survey[3] was then conducted using Google Forms to have experienced pathologists determine the quality of the generated images. However, unlike previous works [18], a diverse set of questions was asked to compare both generated/ground-truth images and positive/negative class images. Containing 16 questions total and utilizing 16 images, the survey is broken up into four parts:

- **4 Questions of**: Matching class labels, one a generated positive (cancerous) class image and one a generated non-cancerous (negative) class image are shown next to one another.
- **4 Questions of**: Determining whether a generated image is of cancerous (positive) class or non-cancerous (negative) class.
- **4 Questions of**: Matching generative labels, one a generated image of stated class, and the other a ground-truth image of the same class from the evaluation set.
- **4 Questions of**: Determining whether an image is generated or a ground-truth image from the evaluation set given the class label.
- **2 Questions of**: Rating a generated image with a known class on features of:
  - *cellularity*: whether the cells presented in the generated image are realistic
  - *atypia*: whether abnormalities in the image (in terms of tissue/cells) are realistic

[3]The link to the form can be found here: https://forms.gle/rexmYg7onTvid3fJ7

- *color*: whether the image presents a realistic standard of pigmentation
- *overall quality*: based on the unstated experience and overall judgement of the evaluator.

4.2. Results

As of the submission of this paper, 9 responses were received, with results shown in the table below:

Table 3. *Four* Questions of +/- Class Matching

Correct (%)	Incorrect (%)
25 (69.4%)	11 (30.6%)

Table 4. *Four* Questions of +/- Class Identification

Class	Correct (%)	Incorrect (%)
+	15 (83.3%)	3 (16.7%)
-	11 (61.1%)	7 (38.9%)

Table 5. *Four* Questions of Generated/Ground-Truth Matching

Class	Correct (%)	Incorrect (%)
+	8 (44.4%)	10 (55.6%)
-	8 (44.4%)	10 (55.6%)

Table 6. *Four* Questions of Generated/Ground-Truth Identification

Class	Correct (%)	Incorrect (%)
GT +	4 (44.4%)	5 (55.6%)
GT -	2 (22.2%)	7 (77.8%)
Gen +	2 (22.2%)	7 (77.8%)
Gen -	3 (33.3%)	6 (66.7%)

Table 7. *Two* Questions of Overall Quality (Scores range from 1 to 5)

Class	Cellularity	Atypia	Color	Overall
+	3.89	4.22	3.67	4.33
-	4.33	4.00	3.67	4.11

5. Discussion

From the results, we can see that generated images preserve the features associated with cancerous/non-cancerous histopathologic scans, in terms of human evaluation. This is made apparent in the first two sets of questions, shown in Table 3 and 4 (with an over 50% accuracy in matching/identifying correct classes). Even when comparing generated to ground-truth images, a general inability for pathologists to clearly distinguish between the two (as shown in Table 5) gives credence to the realism present in the generated batches. However perplexing is the majority of incorrect evaluations from each class of positive/negative when consider ground-truth vs. generated images (shown in Table 6). Over all evaluators, results were generally promising over the three compared attributes of *cellularity*, *atypia*, and *color* (for an average of 3.92 for positive class and 4.00 for negative class), and a very favorable *overall*, above 4.0.

6. Conclusion

In this study, we have demonstrated two key findings:

1. Distinct metastatic features are discernible in images synthesized using a Denoising Diffusion Probabilistic Model (DDPM).

2. There is a minimal observable difference between images generated from different classes.

However, the range of variance observed in the images generated from both positive and negative classes is substantial, indicating potential areas for further enhancement. Future work could involve exploration of more advanced DDPM architectures, such as those employing ConvNeXt Blocks [10] in *lieu* of traditional ResNet Blocks [5]. Additional feature engineering or further data preprocessing may also be beneficial, such as separating images based on cellular composition of adipose (white, fat) tissue vs. lymphatic tissue.

Furthermore, to save on training time, two *unconditional* models were created, without the inclusion of a *conditional* [11][4] U-Net architecture. In the future, with more time/access to more compute resources, a more holistic, conditional model may be considered.

Our assumptions about image fidelity, specifically the identifiability of class status based on the central 32×32px region, lead us to suggest that experimenting with the full 96×96px images available in the Patch Camelyon dataset may yield intriguing results, which may help infer the degree of the presence of cancerous/non-cancerous features affecting distribution learning and therefore image generation. This, in combination with potential upscaling methods [27], may allow for generated image resolution to even surpass that of the original, and comparison into learned features from upscaling may be interesting.

In conclusion, the utilization of DDPMs for the generation of histopathologic images demonstrates considerable promise. These models, with further refinement and enhancement, could play a significant role in improving the understanding and diagnosis of metastatic tissues.

[4]The conditional feature was previously implemented in GANs

References

[1] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes van Diest, Bram van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen A. W. M. van der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, Oscar Geessink, Nikolaos Stathonikos, Marcory CRF van Dijk, Peter Bult, Francisco Beca, Andrew H Beck, Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, Aoxiao Zhong, Qi Dou, Quanzheng Li, Hao Chen, Huang-Jing Lin, Pheng-Ann Heng, Christian Haß, Elia Bruni, Quincy Wong, Ugur Halici, Mustafa Ümit Öner, Rengul Cetin-Atalay, Matt Berseth, Vitali Khvatkov, Alexei Vylegzhanin, Oren Kraus, Muhammad Shaban, Nasir Rajpoot, Ruqayya Awan, Korsuk Sirinukunwattana, Talha Qaiser, Yee-Wah Tsang, David Tellez, Jonas Annuscheit, Peter Hufnagl, Mira Valkonen, Kimmo Kartasalo, Leena Latonen, Pekka Ruusuvuori, Kaisa Liimatainen, Shadi Albarqouni, Bharti Mungal, Ami George, Stefanie Demirci, Nassir Navab, Seiryo Watanabe, Shigeto Seno, Yoichi Takenaka, Hideo Matsuda, Hady Ahmady Phoulady, Vassili Kovalev, Alexander Kalinovsky, Vitali Liauchuk, Gloria Bueno, M. Milagro Fernandez-Carrobles, Ismael Serrano, Oscar Deniz, Daniel Racoceanu, and Rui Venâncio and. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22):2199, Dec. 2017. 1, 2

[2] Lo-Bin Chang, Eran Borenstein, Wei Zhang, and Stuart Geman. Maximum likelihood features for generative image models. *The Annals of Applied Statistics*, 11(3), Sept. 2017. 1, 5, 7

[3] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 1

[4] Hamid Eghbal-zadeh and Gerhard Widmer. Likelihood estimation for generative adversarial networks, 2017. 7

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3, 4, 8

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 2017. 6

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1, 2, 3, 4

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. 3

[9] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, Dec. 2017. 1

[10] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. 8

[11] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 8

[12] Puria Azadi Moghadam, Sanne Van Dalen, Karina C. Martin, Jochen Lennerz, Stephen Yip, Hossein Farahani, and Ali Bashashati. A morphology focused diffusion probabilistic model for synthesis of histopathology images, 2022. 1

[13] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. 1, 4

[14] Minje Park, 2023. 2

[15] Dominic Rampas, 2022. 2

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[17] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), July 2019. 1

[18] Aman Shrivastava and P. Thomas Fletcher. Nasdm: Nuclei-aware semantic histopathology image generation using diffusion models, 2023. 1, 7

[19] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 1, 2, 3

[20] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021. 1

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. 1, 5, 6

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 1, 4

[23] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. June 2018. 1, 2

[24] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning - ICML '08*. ACM Press, 2008. 1

[25] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004. 1, 5

[26] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans, 2021. 1

[27] Tongren Xu, Zhixia Guo, Shaomin Liu, Xinlei He, Yangfanyu Meng, Ziwei Xu, Youlong Xia, Jingfeng Xiao, Yuan Zhang, Yanfei Ma, and Lisheng Song. Evaluating different machine learning methods for upscaling evapotranspiration from flux towers to the regional scale. *Journal of Geophysical Research: Atmospheres*, 123(16):8674–8690, Aug. 2018. 8

7. Supplementary Materials

Figure 11. 256 generated cancerous (positive class) images to be selected for potential use in survey

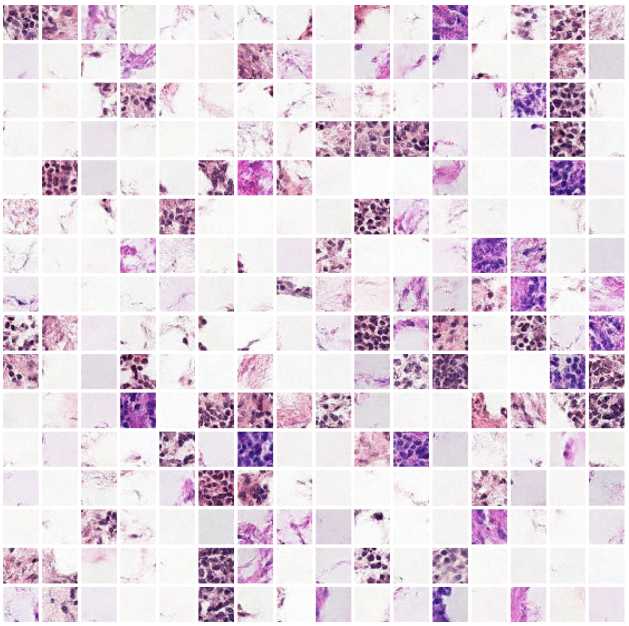

Figure 12. 256 generated non-cancerous (negative class) images to be selected for potential use in survey

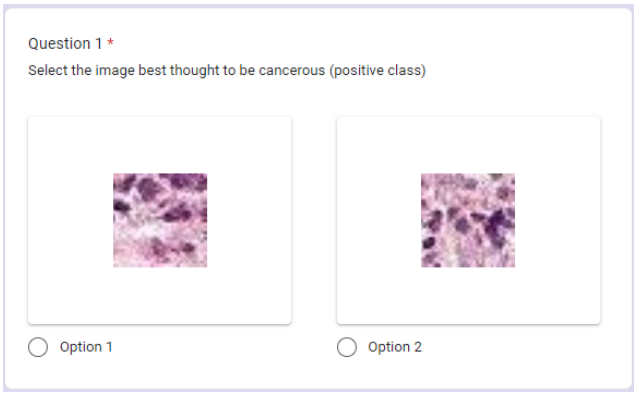

Figure 13. Screenshot of the Google Forms Survey for Section 1, matching class labels for positive and negative class images

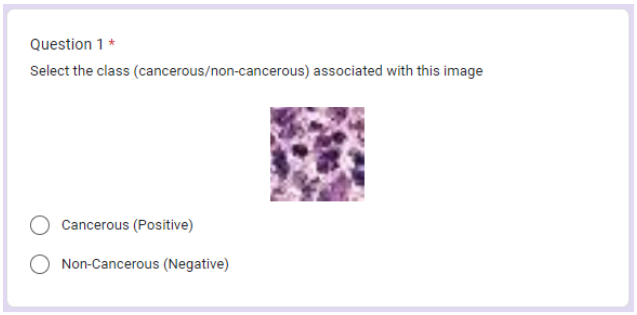

Figure 14. Screenshot of the Google Forms Survey for Section 2, determining whether a known generated image is cancerous or non-cancerous

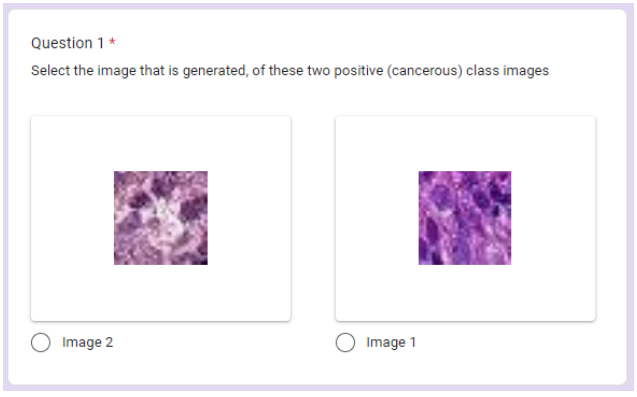

Figure 15. Screenshot of the Google Forms Survey for Section 3, matching two images, one generated and one ground truth, to respective labels (knowing the cancerous/non-cancerous status of the images)

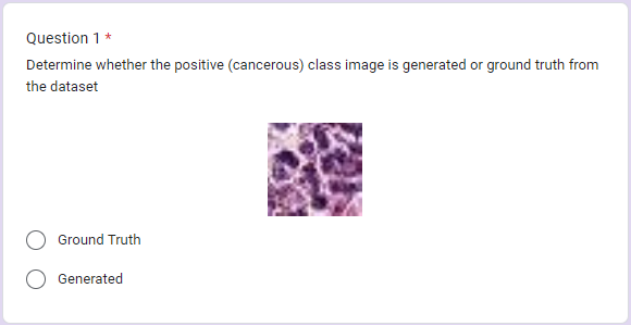

Figure 16. Screenshot of the Google Forms Survey for Section 4, given a known image class (cancerous or non-cancerous), determining whether the image was generated or is ground-truth

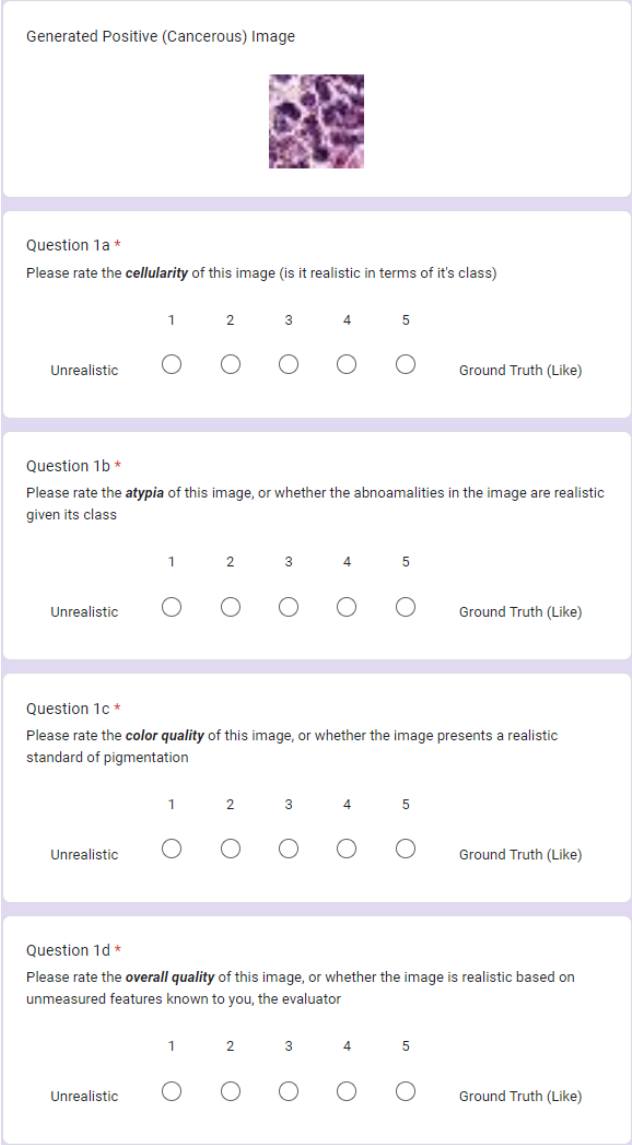

Figure 17. Screenshot of the Google Forms Survey for Section 5,rating a generated image of a known class on qualities of *cellularity*, *atypia*, *color*, and *overall quality*