Appendix

A. Internal defect grading guideline

Table A.1: Guidelines for visual internal defect inspection. Visual inspection was performed by cutting the fruit open along the sagittal plane.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Grade | 0 | 1 | 2 | 3 |
| Internal browning | No | Discoloration of the kernel membranes (light) | Discoloration in core <= 1 cm3 | Discoloration in core > 1 cm3 and/or discoloration of flesh |
| Internal cavity | No | Cavities < 5 mm | Cavities <= 1 cm3 | Cavities > 1 cm3 |
| Internal rots | No | Yes |  |  |

B. Adversarial Variational Autoencoder architecture

The Variational Autoencoder in this work consisted of two parts: an encoder and a decoder. Both were 3D-adapted networks using 3D convolutional layers and normalisation layers. The architectures of the best-performing model are listed in Table B.1 for the encoder and Table B.2 for the decoder. Table B.3 shows the architecture of the ResidualBlock3D shared by both encoder and decoder, used as a skip-connection mechanism (He et al., 2015). All the LeakyReLU layers have a slope of 0.2, and the Upsample layer in the decoder uses trilinear interpolation with a scaling factor of 2. All the mean tensor (*μ*), the logarithm of variance tensor (*σ²*), and the reparameterisation tensor have the shape of (1, 32, 32, 32).

Table B.1: The network architecture of the encoder. It takes the processed CT data as described in Section 2.1 as input. The last layer is replicated to produce the 3D mean tensor (*μ*) and the logarithm of variance tensor (*σ²*).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer | In channel | Out channel | Kernel size | stride | Padding |
| Conv3d | 1 | 256 | 3 | 2 | 1 |
| LeakyReLU |  |  |  |  |  |
| Conv3d | 256 | 512 | 3 | 2 | 1 |
| InstanceNorm3d |  |  |  |  |  |
| LeakyReLU |  |  |  |  |  |
| ResidualBlock3D | 512 | 512 |  |  |  |
| Conv3d \* 2 | 512 | 1 | 3 | 1 | 1 |

Table B.2: The network architecture of the decoder. The input is the reparametrized *z* as described in Eq. (1), and the output is the reconstructed CT scan.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer | In channel | Out channel | Kernel size | stride | Padding |
| Conv3d | 1 | 512 | 3 | 1 | 1 |
| Upsample |  |  |  |  |  |
| Conv3d | 512 | 256 | 3 | 1 | 1 |
| InstanceNorm3d |  |  |  |  |  |
| ReLU |  |  |  |  |  |
| ResidualBlock3D | 256 | 256 |  |  |  |
| Upsample |  |  |  |  |  |
| Conv3d | 256 | 128 | 3 | 1 | 1 |
| InstanceNorm3d |  |  |  |  |  |
| ReLU |  |  |  |  |  |
| ResidualBlock3D | 128 | 128 |  |  |  |
| Conv3d | 128 | 1 | 3 | 1 | 1 |
| Sigmoid |  |  |  |  |  |

Table. B.3: The network architecture of the ResidualBlock3D, shared by the encoder and the decoder. The last layer applies ReLU activation on the summation of the original input and the output from the second-last layer.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer | In channel | Out channel | Kernel size | stride | Padding |
| Conv3d | n | n | 3 | 1 | 1 |
| InstanceNorm3d |  |  |  |  |  |
| ReLU |  |  |  |  |  |
| Conv3d | n | n | 3 | 1 | 1 |
| InstanceNorm3d |  |  |  |  |  |
| ReLU (residual) |  |  |  |  |  |

Table B.4 lists the architecture of the best-performing discriminator for adversarial training. The SpectralNorm layer is used to balance the training of the adversarial framework (Miyato et al., 2018). The ResidualBlock3D and LeakyReLU layers are the same as those described in the Appendix A. The output tensor has the shape of (1, 8, 8, 8), each corresponding to the adversarial score of a 16 × 16 × 16 patch.

Table B.4: The network architecture of the adversarial discriminator.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer | In channel | Out channel | Kernel size | stride | Padding |
| Conv3d | 1 | 16 | 3 | 1 | 1 |
| SpectralNorm |  |  |  |  |  |
| LeakyReLU |  |  |  |  |  |
| Conv3d | 16 | 32 | 3 | 2 | 1 |
| SpectralNorm |  |  |  |  |  |
| LeakyReLU |  |  |  |  |  |
| ResidualBlock3D | 32 | 32 |  |  |  |
| Conv3d | 32 | 64 | 3 | 2 | 1 |
| SpectralNorm |  |  |  |  |  |
| LeakyReLU |  |  |  |  |  |
| ResidualBlock3D | 64 | 64 |  |  |  |
| Conv3d | 64 | 128 | 3 | 2 | 1 |
| SpectralNorm |  |  |  |  |  |
| LeakyReLU |  |  |  |  |  |
| ResidualBlock3D | 128 | 128 |  |  |  |
| Conv3d | 128 | 256 | 3 | 2 | 1 |
| SpectralNorm |  |  |  |  |  |
| LeakyReLU |  |  |  |  |  |
| ResidualBlock3D | 256 | 256 |  |  |  |
| Conv3d | 256 | 1 | 3 | 1 | 1 |

The hyperparameter choices included the number of channels in the first VAE layer (128 or 256), the number of channels in the first discriminator layer (16, 32, or 64), the discriminator batch size (16 or 32), and the weighting of the adversarial loss, which was either scheduled as a sigmoid function (from 0 to 1) of epochs, dynamically adjusted based on the discriminator loss, or fixed at 0.01; initial learning rate (1e-4). Training was optimised using the Adam optimiser.

C. U-Net denoiser

The architecture of the U-Net as the denoiser in the latent diffusion framework is shown in Table C.1. Note that the U-Net should take both the noise and the timestep as input, so it needs to embed the timestep *t* throughout its architecture. Table C.2 shows the Multi-Layer Perceptron for sinusoidal timestep embedding (Vaswani et al., 2023), and Table C.3 shows the ResBlock3D specifically designed for the denoiser to adapt the timestep embeddings.

Table C.1: The U-Net denoiser architecture. The ResBlock3D (skip) layer takes the concatenation of the outputs from its last layer and its mirroring layer as the input.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer | In channel | Out channel | Kernel size | stride | Padding |
| Conv3d | 1 | 128 | 3 | 1 | 1 |
| ResBlock3D | 128 | 256 |  |  |  |
| Conv3d | 256 | 256 | 3 | 2 | 1 |
| ResBlock3D | 256 | 512 |  |  |  |
| Conv3d | 512 | 512 | 3 | 2 | 1 |
| ResBlock3D | 512 | 1024 |  |  |  |
| Conv3d | 1024 | 1024 | 3 | 2 | 1 |
| ResBlock3D | 1024 | 1024 |  |  |  |
| AttentionBlock | 1024 | 1024 |  |  |  |
| ResBlock3D | 1024 | 1024 |  |  |  |
| Upsample |  |  |  |  |  |
| Conv3d | 1024 | 1024 | 3 | 1 | 1 |
| ResBlock3D (skip) | 2048 | 512 |  |  |  |
| Upsample |  |  |  |  |  |
| Conv3d | 512 | 512 | 3 | 1 | 1 |
| ResBlock3D (skip) | 1024 | 256 |  |  |  |
| Upsample |  |  |  |  |  |
| Conv3d | 256 | 256 | 3 | 1 | 1 |
| ResBlock3D (skip) | 512 | 128 |  |  |  |
| GroupNorm |  |  |  |  |  |
| SiLU |  |  |  |  |  |
| Conv3d | 128 | 1 | 3 | 1 | 1 |

Table C.2: Multi-layer Perceptron for timestep embedding.

|  |  |  |
| --- | --- | --- |
| Layer | In feature | Out feature |
| SinusoidalEmbeding | 1 | 128 |
| Linear | 128 | 512 |
| SiLU |  |  |
| Linear | 512 | 512 |

Table C.3: ResBlock3D for the denoising U-Net with timestep embeddings. The block at first has two parallel paths; one processes the noise, while the other processes the time embeddings. Outputs from both paths are then summed and passed through the last path. The final output is the summation of the original input and the output from the last layer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Path | Layer | Kernel size | stride | Padding |
| x | GroupNorm |  |  |  |
| SiLU |  |  |  |
| Conv3d | 3 | 1 | 1 |
| t | SiLU |  |  |  |
| Linear |  |  |  |
| x + t | GroupNorm |  |  |  |
| SiLU |  |  |  |
| Conv3d (skip) | 3 | 1 | 1 |

The selected hyperparameters included the number of initial channels (128), total diffusion timesteps (1000), initial learning rate (1 × 10⁻5), and number of denoising steps (200). Training was optimised using the Adam optimiser, with the learning rate further refined through cosine annealing with warm restarts. The optimal model was chosen based on the epoch that achieved the lowest validation loss.

D. Model optimisation

The experiment was conducted mainly to identify the optimal neural network structure capable of balancing the adversarial loss and the discriminator loss, with results shown in Table D.1. Initial results indicate that a weak VAE produces poor reconstructions, as reflected by low SSIM scores, which in turn allows the discriminator to easily distinguish between real and generated samples. Moreover, the adversarial loss weight (γ) is critical for maintaining this balance; without proper tuning, one loss may diverge while the other remains at a low value. Patch size was also found to have a strong influence on SSIM, which reflects the level of image clarity. The best-performing VAE model in terms of SSIM is the one that balances the structures of the VAE and the Discriminator while also using a small patch size for sharper results and a low gamma to stabilise adversarial training. It was also shown that the discriminator does not need to be excessively large; it only needs to be sufficiently strong to guide the VAE in achieving high-quality reconstructions.

Table D.1: Evaluation metrics on validation dataset for adversarial VAEs with difference hyperparameter combinations

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | VAE channel | Discriminator channel | Patch size | gamma | MAE | SSIM | PSNR |
| 1 | 128 | 64 | 16 | 0-1 | 0.0064 | 0.9704 | 29.8712 |
| 2 | 128 | 64 | 16 | 1 | 0.0094 | 0.9556 | 26.0403 |
| 3 | 128 | 32 | 32 | 0-1 | 0.0065 | 0.9697 | 28.6096 |
| 8 | 256 | 32 | 32 | 0.1-1 | 0.0075 | 0.9624 | 27.2807 |
| 5 | 256 | 64 | 32 | 0-1 | 0.01 | 0.9424 | 24.5202 |
| 6 | 256 | 32 | 16 | 0.01-1 | 0.0066 | 0.9655 | 28.537 |
| 7 | 256 | 32 | 32 | 0.01-1 | 0.0045 | 0.9773 | 31.0849 |
| 8 | 128 | 16 | 16 | 0.01 | 0.0034 | 0.9852 | 34.1527 |
| **Best** | **256** | **16** | **16** | **0.01** | **0.0025** | **0.9915** | **36.9226** |

E. The training curves

Fig. E.1 presents the learning curves of the best adversarial VAE model. The overall loss (a) shows an initial sharp decline followed by a gradual decreasing trend in the training set, whereas the validation loss exhibits greater fluctuation. This pattern can be attributed to the stable adversarial loss observed during training (b), in contrast to the sudden changes and instability in the validation adversarial loss (c). The adversarial weighting hyperparameter γ was tuned such that both adversarial loss and MAE loss dominated the loss function, while the KL loss remained minimal. Fig. E.2 shows that the adversarial loss on both the training (a) and validation (b) sets was generally higher than the discriminator loss, indicating that generating realistic data is more challenging than distinguishing real from synthetic samples. Moreover, although the value of adversarial loss did not consistently improve, the visual quality of reconstructions steadily increased over time. Notably, the validation adversarial loss displayed sudden surges followed by sharp declines, underscoring the inherent instability of training despite the use of hinge loss in this study. During training, very high GPU memory usage (>30 GB) was observed with a batch size of 4, primarily due to gradient tracking in the upsampling layers of the decoder. Training the best-performing adversarial VAE required approximately 30 hours on an NVIDIA H100 GPU.

A screenshot of a graph

AI-generated content may be incorrect.

Fig. E.1: (a) Learning curves of the best VAE model; (b) training loss and (c) validation loss decomposed into three loss components: , , and , respectively.

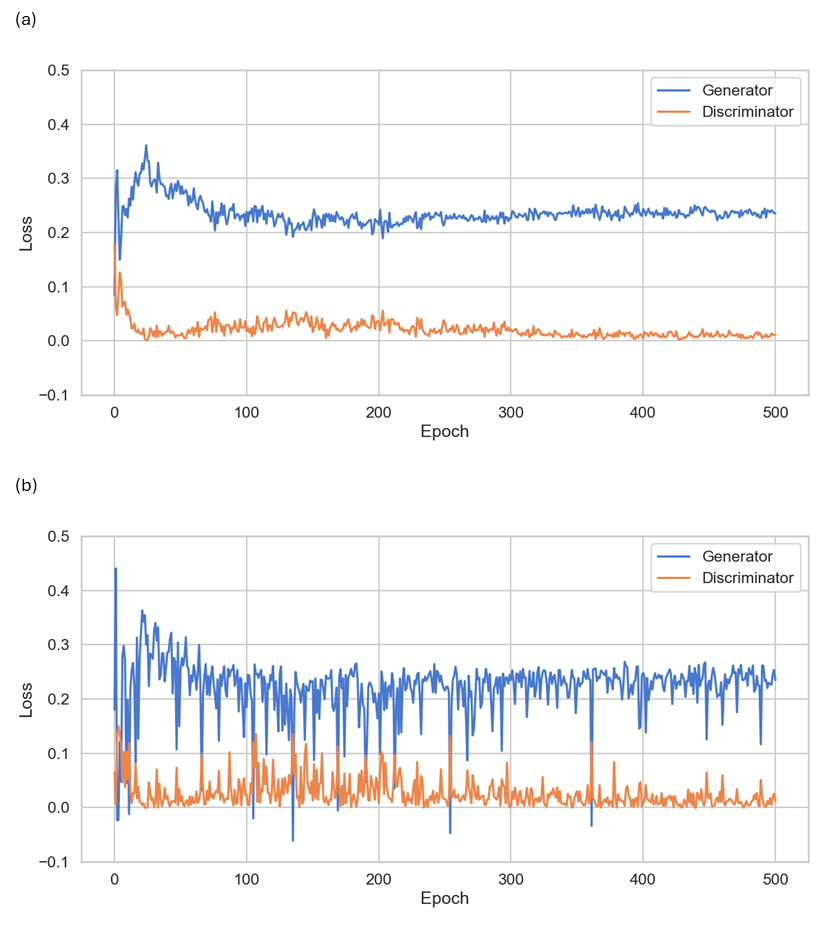


Fig. E.2: (a) training and (b) validation adversarial loss versus discriminator loss.

The learning curves of the best U-Net denoiser are shown in Fig. E.3. Both training and validation losses exhibit a sharp initial decrease, followed by a gradually declining yet fluctuating trend. The curves remain close to each other, indicating no overfitting, although the validation loss shows greater fluctuation than the training loss. The diffuser achieved a training loss of 0.0067 and a validation loss of 0.0060, indicating a rather accurate prediction of the target noise term. Training the denoiser model took approximately 19 hours on an NVIDIA H100 GPU.

A graph with orange and blue lines

AI-generated content may be incorrect.

Fig. E.3: Learning curves of the best denoiser

F. More synthetic data visualisation

A group of white objects

AI-generated content may be incorrect.

Fig. F.1: Coronal plane images extracted from randomly generated CT images (1).

A group of white teardrops

AI-generated content may be incorrect.

Fig. F.2: Coronal plane images extracted from randomly generated CT images (2).

A group of white objects

AI-generated content may be incorrect.

Fig. F.3: Coronal plane images extracted from randomly generated CT images (3).

A group of white objects

AI-generated content may be incorrect.

Fig. F.4: Coronal plane images extracted from randomly generated CT images (4).

A group of white objects

AI-generated content may be incorrect.

Fig. F.5: Coronal plane images extracted from randomly generated CT images (5).

References

1. He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. http://arxiv.org/abs/1512.03385
2. Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). *Spectral Normalization for Generative Adversarial Networks*. http://arxiv.org/abs/1802.05957
3. Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2023). *Attention Is All You Need*.