IAC-25-B1.4.4.x98985

Synthesis of a multispectral image dataset for ML-based space surveillance

Jordan Kildare*, Jarrad Kight, Michael Evans, Yee Wei Law

UniSA STEM, University of South Australia, Mawson Lakes, SA 5095, Australia.
Email: jordan.kildare@unisa.edu.au, jarrad.knight@mymail.unisa.edu.au, michael.evans@unisa.edu.au, yeewei.law@unisa.edu.au
* Corresponding Author

Abstract

Space surveillance involves detecting and tracking high-speed vehicles, such as re-entry vehicles. Due to the excessive speeds of atmospheric re-entry, objects entering the atmosphere from space are heated significantly and emit strongly in both the short-wavelength infrared (SWIR) and mid-wavelength infrared (MWIR) regimes. However, MWIR satellite imagery is significantly limited in the public domain, thus limiting capability for training machine learning (ML) algorithms to detect atmospheric entry in this band. Despite the limited MWIR imagery, there is a wealth of data from the Landsat program that encompasses both SWIR and long-wavelength infrared (LWIR) bands. The availability of this data provides some scope for interpolation between these measurements to generate alternative bands. This work proposes a diffusion model that incorporates existing spectral data to generate SWIR and MWIR band data from the surrounding spectral measurements. A hyperspectral data source — Earth Observation 1's Hyperion instrument — is used to develop the training dataset, where the conditional input bands are selected to match those available from the Landsat instruments, and target bands are selected outside the Landsat band ranges. The methodology of an existing hyperspectral image generator, HyperLDM, is adapted for small-scale inputs and rapid image synthesis, without the requirement of known spectral endmembers for radiance prediction. The developed model is capable of high-quality generation of images in the specified bands, typically with less noise artefacts than the ground truth bands. The decoder structure is further analysed using a class activation mapping method to improve interpretability of the model outputs. In particular, the conditional input is shown to provide the small-scale details of a scene during reconstruction, whereas the quantised latent code provides major spatial features.

Keywords: Space surveillance, satellite images, hyperspectral super-resolution, generative adversarial network, diffusion model

Nomenclature

E_x	Expectation taken over distribution of x
$\boldsymbol{x}$	Vector
$\mathbf{X}$	Matrix or tensor

Acronyms/Abbreviations

BCE	Binary cross-entropy
CBAM	Convolution block attention module
CNN	Convolutional neural network
CSP	Cross-spatial partial
DDPM	Denoising diffusion probabilistic model
DL	Deep learning
DNN	Deep neural network
GAN	Generative adversarial network
HSI	Hyperspectral image
HSR	Hyperspectral super-resolution
MAE	Mean absolute error
ML	Machine learning
MSE	Mean squared error
MSI	Multispectral image
MSSIM	Mean structural similarity
MWIR	Mid-wavelength infrared
OLI	Operational Land Imager
PSA	Position-sensitive attention
PSNR	Peak signal-to-noise ratio
SAM	Spectral angle mapper
SSIM	Structural similarity
SWIR	Short-wavelength infrared
VAE	Variational autoencoder
VQGAN	Vector-quantised GAN

1. Introduction

Space surveillance has become an increasing prevalent topic as the barrier to access space decreases. Some high-speed vehicles re-enter the atmosphere and fly at such low altitude that ground-based sensors that rely on line-of-sight detection and tracking become ineffective [1]. As an alternative, space-based multispectral or hyperspectral sensors focusing on the infrared bands can be deployed to detect, identify and track these vehicles from low or medium Earth orbits, because of these vehicles' unique spectral signatures [10]. Due to the high dimensionality of the resulting multispectral images (MSIs) and hyperspectral images (HSIs), object detection of this kind is well posed for deep learning (DL) approaches, especially using HSI data [20]. However, a major challenge in developing DL

models is the lack of appropriate training data. Another challenge is, due to the size of some re-entry vehicles (5-10m characteristic length), the space-based sensors may only be capable of observing a signal in a single pixel [40]. This work focusses on solving the first challenge, by proposing a method for synthesising satellite images with the desired spectral bands, enabling vehicle spectral signatures to be superimposed on these images for DL model development.

A challenge to satellite image data synthesis is the inherent tradeoff between spatial resolution and spectral resolution [18]: HSI data have higher spectral resolutions than MSI data, but MSI data have higher spatial resolutions and are available in larger quantities. An opportunity can be spotted here for synthesising an image dataset with high spectral and spatial resolutions from MSI and/or HSI data. *Hyperspectral super-resolution* (HSR) methods generate HSIs of high spatial resolutions from HSIs of low spatial resolutions. The method described in this paper is an HSR method based on HyperLDM [25].

The method proposed in this work applies the same workflow as the HyperLDM method, whereby an autoencoder network is trained as a *generative adversarial network* (GAN), and the latent space is used for training a diffusion model. The encoder and decoder in our work are both based on the C3k2 and position-sensitive attention (PSA) blocks, as implemented in the YOLO series of object detectors [21]. The result is substantially more lightweight than HyperLDM's ResNet-based architecture. Both our model and HyperLDM quantise the latent-space output from the encoder to create a learnable codebook, equivalent to latent-space endmembers [34][39]. Rather than learning endmember abundance maps, our model directly predicts the spectra produced without the requirement of a spectral library and linear unmixing process, imposing significantly lower resource requirements compared to HyperLDM. Further to demonstrating the effectiveness of the model in synthesising HSIs, the interpretability of the model is explored from the latent space distribution. In summary, the contributions of this work include:

- Reduction in model size and complexity compared to HyperLDM.
- Implementation of a channel-based self-attention block for better spectral contribution capture.
- High-speed inference of HSI images.
- Correlation of physical features with latent-space characteristics enhances model interpretability.

The rest of this paper is organised as follows. Section 2 provides the technical background. Section 3 describes the proposed HSI synthesis methodology. Section 4 presents some results, shares insights extracted from the results, and discusses limitations of the proposed approach. Section 5 concludes and sketches planned future work.

2. Preliminaries

This section briefly reviews related work to HSI synthesis and introduces two key building blocks of the proposed HSR method.

2.1 Related work

HSI synthesis methods, also known as remote sensing fake sample generation [43], aim at generating satellite images at high spectral and spatial resolutions. The following HIS synthesis methods generate high-spatial-resolution HSI data from different inputs:

- HSR methods take low-spatial-resolution HSI data as input. For example, HyperLDM [25] has been applied to the IEEE GRSS Data Fusion Contest 2018 dataset.
- Spectral super-resolution methods take high-spatial-resolution MSI data as input. For example, Chen et al.'s spectral-cascaded diffusion model [5] has been applied to a multispectral derivative of the IEEE GRSS Data Fusion Contest 2018 dataset.
- Image fusion methods take high-spatial-resolution MSI data and low-spatial-resolution HSI data as inputs. For example, Wu et al.'s HSR-Diff method [42] has been applied to fuse 30m-resolution HSIs and 10m-resoultion MSIs.

Most state-of-the-art HSI synthesis methods use a *denoising diffusion probabilistic model* (DDPM) in their core; see Section 2.2 for an introduction. There are disadvantages with using a GAN for HSI synthesis, as discussed in Section 2.3. Likelihood-based methods such as variational autoencoders are generally considered to be unable to synthesise high-quality images [6], and hence will not be discussed further.

2.2 Conditional denoising diffusion probabilistic model

A diffusion probabilistic model, or diffusion model for short, is a latent-variable model that takes the form of a parameterised Markov chain, which is trained using variational inference to produce samples matching the data after a finite time; the term "diffusion" refers to the Markov-chain-based conversion of a simple known distribution (e.g., Gaussian) into a target (data) distribution [25][38].

The central idea of *denoising diffusion probabilistic models* (DDPMs) is taking each training image and corrupting it in a multi-step noise process into a sample from a Gaussian distribution [14]. A deep neural network (DNN) is trained to invert this process so that once trained, the DNN can generate new images starting with Gaussian samples. As illustrated in Fig. 1, $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$ is the data to be generated, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ are latents of the same dimensionality as $\boldsymbol{x}_0$. The joint distribution $p_\theta(\boldsymbol{x}_{0:T})$ is called the *reverse process* (or reverse diffusion process), whereas the posterior distribution $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ is called the *forward process* (or forward diffusion process).

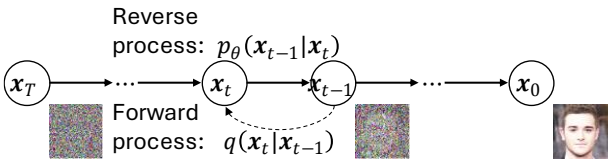

Fig. 1: A DDPM is a parameterised Markov chain whose transitions need to be learnt. Diagram is adapted from [14].

By definition, $p(\boldsymbol{x}_T)$ is a zero-mean Gaussian distribution with unit covariance, i.e., $p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \mathbf{I})$. Unlike other latent-variable models, the diffusion process $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ is fixed to a Markov chain that gradually adds noise to the data according to the variance schedule or forward diffusion schedule $\beta_1, \dots, \beta_T$ $(0 < \beta_t < 1)$:

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \Pi_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \tag{1}$$

where $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_T; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}\right)$. Equivalently,

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_t, \tag{2}$$

where $\boldsymbol{\epsilon}_t$ follows the standard Gaussian distribution. The usage of square roots in Eq. (2) and the design of the variance schedule where the diffusion rate increases monotonically, help the mean of $\boldsymbol{x}_t$ converge to zero and the covariance of $\boldsymbol{x}_t$ converge to an identity matrix [3].

Training is performed by optimising the variational bound on the negative log likelihood [14]: $\mathcal{L} \triangleq \mathrm{E}_q\left[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$, or equivalently,

$$\mathcal{L} = \mathrm{E}_q\left[-\log p(\boldsymbol{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right]. \tag{3}$$

Expressing $\mathcal{L}$ in Eq. (3) in terms of the Kullback-Leibler divergence between q and p speeds up training, because the Kullback-Leibler divergence between Gaussians can be computed in a Rao-Blackwellised fashion [4] with closed-form expressions instead of high-variance Monte Carlo estimates [14].

Unlike likelihood-based methods, such as variational autoencoders (VAEs), diffusion models do not use competitive log likelihoods. However, diffusion models can be viewed as a form of hierarchical VAE, in which the encoder distribution is fixed (as Gaussian), and only the generative distribution is learnt [3]. Training of diffusion models can be parallelised but the need for multiple forward passes through the decoder network renders diffusion models computationally expensive [9].

For a dataset $\mathcal{D} = \{\boldsymbol{x}_i\}$, a DDPM learns the data distribution $p(\boldsymbol{x})$ and generates samples consistent with the distribution. A *conditional* DDPM is an extension of DDPM that, given a labelled dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}$, learns the conditional distribution $p(y|\boldsymbol{x})$ [43]. Given label y for inference, a conditional DDPM generates a sample consistent with the learnt distribution.

2.3 Conditional generative adversarial network

A generative adversarial network (GAN) consists of two neural networks – a generator and a discriminator – working in tandem [13]. The generator and discriminator are typically structured like the decoder and encoder of an autoencoder respectively [12][29]. The generator, denoted by G, generates test data samples, while the discriminator, denoted by D, classifies the generated samples as real or generated. G takes as input a noise variable $z \sim p_z$, while D takes as input $G(z)$ and outputs a value indicating the probability that its input is real rather than generated. While D is optimised to minimise the cross-entropy loss, G is optimised to minimise $\ln\left(1 - D\left(G(z)\right)\right)$, i.e., G aims to make D output a value that is close to 1 (representing "real") as possible. The result is having G and D play the following minimax game:

$$\min_G \max_D \left\{\mathrm{E}_{x\sim p_x}[\ln D(x)] + \mathrm{E}_{z\sim p_z}\left[\ln\left(1 - D\left(G(z)\right)\right)\right]\right\}. \tag{4}$$

A *conditional* GAN incorporates auxiliary information such as class label y into the minimax game [32]:

$$\min_G \max_D \left\{\mathrm{E}_{x\sim p_x}[\ln D(x|y)] + \mathrm{E}_{z\sim p_z}\left[\ln\left(1 - D\left(G(z|y)\right)\right)\right]\right\}. \tag{5}$$

Before the emergence of diffusion models, GANs represented the state of the art for the quality of generated samples. However, the optimisation formulation in Eq. (4) makes training of GANs tricky and often causes mode collapse [36]. Furthermore, regulariser design is complex [19]; it is a common observation that GAN-based methods require sophisticated regularisation and training. A commonly used regulariser is [31]:

$$R_1(\psi) = \frac{\gamma}{2}\mathrm{E}_{x\sim p_x}\left[\left\|\nabla D_\psi(x)\right\|^2\right], \tag{6}$$

where ψ is the discriminator's parameter vector, γ is a hyperparameter, p_x is the true data distribution, $\nabla D_\psi(x)$ is the gradient of the discriminator's output.

Nevertheless, a GAN is often used as part of larger image synthesis framework. For example, as shown in Fig. 2, the vector-quantised GAN (VQGAN) [12] architecture integrates a GAN (G and D) with an encoder (E) and a transformer, where the components G, D and E are convolutional neural networks (CNNs). This architecture has the advantage of combining the long-range modelling capabilities of a transformer with the inductive bias of CNNs for local interactions, without incurring the transformer's high computational cost for generating high-resolution images.

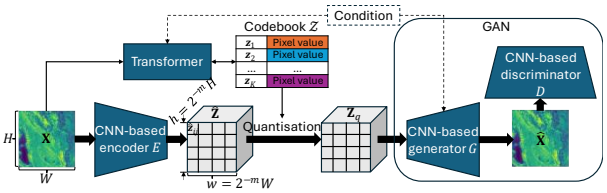

Fig. 2: Architecture of VQGAN [12]. Encoder E maps H-by-W image $\mathbf{X}$ to an h-by-w latent space. The quantisation operation is captured in Eq. (7). Generator G reconstructs $\mathbf{X}$ as $\hat{\mathbf{X}}$, which consists of 2^{2m} h-by-w patches. Discriminator D classifies each patch as fake or real. Conditioning information, if available, is applied to both the GAN and the transformer. Diagram is adapted from [12].

In VQGAN, the transformer expresses the content of an image as a sequence of codes. Given an RGB image $\mathbf{X} \in \mathbb{R}^{H\times W\times 3}$, encoder E compresses $\mathbf{X}$ to $E(\mathbf{X}) = \hat{\mathbf{Z}} \in \mathbb{R}^{h\times w\times n_z}$, which is quantised into the codes from n_z-dimensional codebook $\mathcal{Z} = \{\boldsymbol{z}_k\}_{k=1}^{K} \subset \mathbb{R}^{n_z}$:

$$\mathbf{Z}_q = \arg\min_{\boldsymbol{z}_k\in\mathcal{Z}} \left\|\hat{\boldsymbol{z}}_{ij} - \boldsymbol{z}_k\right\| \in \mathbb{R}^{h\times w\times n_z}. \tag{7}$$

The generator G generates $\hat{\mathbf{X}}$ from $\mathbf{Z}_q$, which can be considered as a reconstruction of $\mathbf{X}$, i.e., $G(\mathbf{Z}_q) = \hat{\mathbf{X}} \approx \mathbf{X}$. The discriminator D is a patch-based variant [16] that classifies each of the 2^{2m} patches in $\hat{\mathbf{X}}$ as real or reconstructed. Due to the GAN component, the optimisation function is a minimax formulation:

$$\min_{E,G,\mathcal{Z}} \max_{D} \mathrm{E}_{\mathbf{X}}\left[\mathcal{L}_{\mathrm{VQ}} + \lambda_{\mathrm{GAN}}\mathcal{L}_{\mathrm{GAN}}\right], \tag{8}$$

where

$$\mathcal{L}_{\mathrm{GAN}} = \ln D(\mathbf{X}) + \ln\left(1 - D(\hat{\mathbf{X}})\right), \tag{9}$$

$$\mathcal{L}_{\mathrm{VQ}} = \mathcal{L}_{\mathrm{rec}} + \left\|\mathrm{sg}[E(\mathbf{X})] - \mathbf{Z}_q\right\|^2 + \beta_{\mathrm{VQ}}\left\|\mathrm{sg}[\mathbf{Z}_q] - E(\mathbf{X})\right\|^2. \tag{10}$$

In Eqs. (8) and (10), λ_{GAN} and β_{VQ} are hyperparameters. In Eq. (10), $\mathcal{L}_{\mathrm{rec}}$ captures the reconstruction loss, which can be an L_2 loss or a "perceptual loss" [44]; the operator sg denotes the stop-gradient operation, which is necessary because the quantisation operation is not differentiable.

3. Methodology

3.1 Satellite data

Satellite data was collated from the Hyperion instrument onboard the Earth Observing (EO-1) satellite [11]. The resulting dataset includes images for multiple locations and times during the mission lifecycle, particularly over Australia and its surrounding oceans. Of the 220 unique hyperspectral bands collected by Hyperion, 198 are radiometrically calibrated, and are separated into visible to near-infrared, and short-wavelength infrared images. A process of digital number conversion to irradiance and data destriping was applied to process the raw data from Hyperion into a usable format. Dead and flat detector values were set to zero, and a median filter with a 3×3 kernel was applied to these pixels. Fig. 3 shows a comparison of the original Level 1 Hyperion data and the processed data used in the model. Minimal other post-processing was performed to demonstrate robustness of the model for potential onboard processing.

Spectral response functions from the Landsat 8 Operational Land Imager (OLI) instrument [33] were used to aggregate the Hyperion bands into a representative Landsat 8 image [17]. Before being input

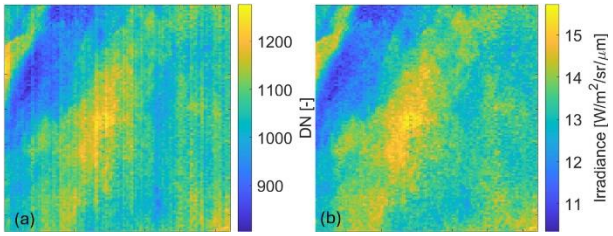

Fig. 3: Comparison of example Hyperion data processing. (a) before and (b) after the destriping and scaling process was applied.

into the model, minimum and maximum values are calculated for the entire dataset on a band-wise basis, and used to normalise the inputs to a range of -1 to 1.

A total of 3504 images were used to train this model, with a 70/30 split for training and testing. Each input HSI consists of six hyperspectral bands, each with a 10nm bandwidth, that are selected from outside the range of the Landsat OLI. These bands are 1020nm, 1195nm, 1215nm, 1530nm, 1720nm, and 2215nm, selected as they contain relatively clean data signals. The conditional MSIs consist of four multispectral bands, with bandwidths ranging from 70-200nm, corresponding to the NIR, Cirrus, SWIR1, and SWIR2 Landsat bands.

3.2 Machine learning model architecture

A conditional latent space diffusion model coupled to a GAN, inspired by HyperLDM [14], was applied in this work. An architecture diagram is presented in Fig. 4 for the applied model. The encoder of the original implementation consisted of repeated ResNet-style [8] dense blocks to increase latent channel dimensions. A partial decoding was applied up until a specified level of spatial downsampling was achieved, before the output is passed to an embedder. Comparatively, the encoder portion of the network, E in Fig. 4, in this work applies a series of cross-spatial partial (CSP) blocks, and PSA blocks, drawing from the latest YOLO architecture [21]. The attention blocks applied in HyperLDM use separate learnable convolutions for query, key, and value

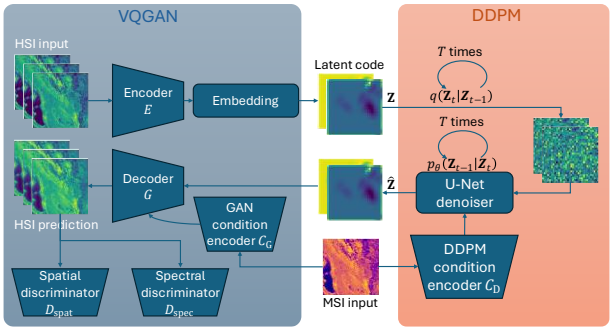

Fig. 4: Architecture of the proposed method combining a conditional VQGAN with a DDPM. HSI inputs are only required for the training of the VQGAN. $\mathbf{Z}$ denotes the output of the embedder. $\hat{\mathbf{Z}}$ denotes the output of the denoiser. Diagram is adapted, with inaccuracies rectified, from [24].

projections, whereas the attention mechanism applied in our work shares the projection to reduce parameters. Partial decoding is still applied before the embedding step, as in the original HyperLDM implementation. In addition, multi-head spectral self-attention blocks are also applied in our encoder to preserve long distance spectral information, as in a squeeze-and-excitation model [15]. The embedder is a codebook of learnable vectors that allows quantisation of the latent space into a more compact format. The output of the encoder is matched to an embedding value using a minimum Euclidean distance metric, with the loss from this metric used to train the embedding parameters. In this work, 512 embeddings are used, as opposed to the 1024 of HyperLDM.

The decoder/generator, G in Fig. 4, continues the decoding trajectory of the encoder using CSP and spatial attention blocks, and includes a conditional input pathway. Skip connections are applied between the conditional input pathway and the decoder stages to preserve fine spatial features. As in the case of HyperLDM, the same conditional input is also used to train the diffusion model. Compared to HyperLDM, no spectral library is used to convert the abundance maps to the HSI. Instead, it directly generates the HSI.

The diffusion model is a U-Net-based [35] model, which is applied iteratively to the conditional input and the noisy latent space representation of the HSI data. The diffusion model in this work uses ResNet blocks with convolution block attention modules (CBAMs) and timestep embedding, which include channel and spatial attention in each stage through depth-wise convolutions and adaptive pooling. Comparatively, the diffusion model in HyperLDM uses ResNet blocks, as in the VQGAN, with timestep embedding.

Both a spatial discriminator and a spectral discriminator are used in this work. The spatial discriminator, D_{spat} in Fig. 4, follows a CNN format with four hidden layers. Each layer consists of a strided convolution that downsamples the input by a factor of two, and filters through a 4×4 kernel with $c2^n$ channels, where c is the base number of channels for the network (128), and n is the current layer. After each convolution in the hidden layers, a batch normalisation is applied. The final convolution reduces the channel dimension to unity without striding, so that the BCE loss can be calculated.

The spectral discriminator, D_{spec} in Fig. 4, consists of fully connected (linear) layers in the channel dimension. As such, the two-dimensional input requires pre-processing to apply discrimination to a single dimension. At each step, a number of intermediate pixels are sampled from the prediction, which corresponds to a square location as determined by a random sampling process for the central point. The discriminator iterates through the selected locations and applies the linear layers and a leaky rectified linear unit (LeakyReLU) activation function [28]. A sigmoid function is applied at the end of the network before the output is rearranged to apply the BCE loss function.

Training the VQGAN part of the HyperLDM-based model involves solving the minimax problem:

$$\min_{E,G,Z,C_G,C_D} \max_{D} \mathrm{E}_{\mathbf{X}}\left[\mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{adv}}\right], \tag{11}$$

where $\mathcal{L}_{\text{gen}}$ is a variation of $\mathcal{L}_{\text{VQ}}$ from Eq. (10):

$$\begin{aligned}\mathcal{L}_{\text{gen}} = \lambda_1 \mathcal{L}_{\text{pxl}}(\mathbf{X}, \hat{\mathbf{X}}) + \lambda_2 \mathcal{L}_{\cos}(\mathbf{X}, \hat{\mathbf{X}}) \\ + \left\|\text{sg}[E(\mathbf{X})] - \mathbf{Z}_q\right\|^2 \\ + \beta_{\text{VQ}} \left\|\text{sg}[\mathbf{Z}_q] - E(\mathbf{X})\right\|^2,\end{aligned} \tag{12}$$

$$\begin{aligned}\mathcal{L}_{\text{adv}} = \ln D_{\text{spat}}(\mathbf{X}) + \ln\left(1 - D_{\text{spat}}(\hat{\mathbf{X}})\right) \\ + \lambda_3 \left[\ln D_{\text{spec}}(\mathbf{X}) \right. \\ \left. + \ln\left(1 - D_{\text{spec}}(\hat{\mathbf{X}})\right)\right].\end{aligned} \tag{13}$$

In Eq. (12), $\mathcal{L}_{\text{pxl}}(\mathbf{X}, \hat{\mathbf{X}}) = \left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_1$ and $\mathcal{L}_{\cos}(\mathbf{X}, \hat{\mathbf{X}})$ measures the cosine similarity between the spectral vectors in $\mathbf{X}$ and the spectral vectors in $\hat{\mathbf{X}}$ [26]. For the results reported below, the hyperparameters were configured as follows: $\lambda_1 = 1$, $\lambda_2 = 10$, $\lambda_3 = 0.1$ and $\beta_{\text{VQ}} = 0.25$.

Instead of the typical variational bound loss function of the original DDPM formulation in Eq. (3), the loss function for our DDPM consists of an L_2 loss term between the output of the denoiser and the output of the embedder, alongside a multi-scale structural similarity index measure (MSSIM) term [41] for improving visual recreation quality.

MSSIM is defined based on structural similarity (SSIM), which measures for every channel similarity in terms of luminance, contrast and structure. The MSSIM loss is calculated as the weighted contribution of the SSIM from five levels of image coarseness. An 11×11

window region based on a two-dimensional Gaussian distribution is convolved via a 2D convolution with the predicted image to produce mean value $\mu_{\hat{z}}$, and separately with the ground-truth image to produce mean value μ_z. The convolution is applied depth-wise and thus conserves the channel dimensions of the inputs. A similar process is applied to the element-wise square of the image, and to the element-wise multiplication of the predicted and ground-truth images. The luminance term of the SSIM is defined as:

$$l(z,\hat{z}) = \frac{2\mu_z\mu_{\hat{z}} + C_1}{\mu_z^2 + \mu_{\hat{z}}^2 + C_1}, \tag{14}$$

where $C_1 = (0.01 \cdot \max(\mu_z))^2$, and $\max(\mu_z) = 1$ by definition. The contrast term in the SSIM calculation is defined by the standard deviation of the distribution of the prediction and ground truth:

$$\sigma_z = \left(\frac{1}{N-1}\sum_{i=1}^{N}(z_i - \mu_z)^2\right)^{\frac{1}{2}}. \tag{15}$$

where N is the number of pixels. The resulting contrast term in SSIM calculation is similarly constructed to the luminance term as:

$$c(z,\hat{z}) = \frac{2\sigma_z\sigma_{\hat{z}} + C_2}{\sigma_z^2 + \sigma_{\hat{z}}^2 + C_2}, \tag{16}$$

with $C_2 = (0.03 \cdot \max(\mu_z))^2$. The structural component of the SSIM is derived from the geometric angle between the vectors of mean images as:

$$\sigma_{z\hat{z}} = \frac{1}{N-1}\sum_{i=1}^{N}(z_i - \mu_z)(\hat{z}_i - \mu_{\hat{z}}), \tag{17}$$

which is then applied to define the structure term:

$$s(z,\hat{z}) = \frac{\sigma_{z\hat{z}} + C_3}{\sigma_z\sigma_{\hat{z}} + C_3}, \tag{18}$$

with $C_3 = C_2/2$.

Multiplying the terms defined by Eqs. (14), (16), and (18), the SSIM can be defined and simplified to:

$$\text{SSIM} = \frac{(2\mu_z\mu_{\hat{z}} + C_1)(2\sigma_{z\hat{z}} + C_2)}{(\mu_z^2 + \mu_{\hat{z}}^2 + C_1)(\sigma_z^2 + \sigma_{\hat{z}}^2 + C_2)}. \tag{19}$$

An average pooling process is applied with a kernel size of 2 and a stride of 2 to generate each level of multi-scale input for the MSSIM metric. The contrast term is combined with a weighting term to give the multi-scale MSSIM as:

$$\text{MSSIM} = \left(\prod_{n=1}^{M}\left(\frac{2\sigma_{z\hat{z}} + C_2}{\sigma_z^2 + \sigma_{\hat{z}}^2 + C_2}\right)_n^{w_n}\right)\text{SSIM}_M^{w_M}. \tag{20}$$

where n indexes the multiscale level, and M is the number of multi-scale levels. In this work, $M = 5$ and $w_{1:5} = [0.0448, 0.2856, 0.3001, 0.2363, 0.1333]$.

The VQGAN was trained for 4000 epochs with a cosine-based annealing learning rate that is separate for the GAN, denoising model, and the discriminators. The DDPM is then trained for a further 4000 iterations based on the latent code produced by the encoder of the GAN. The Adam optimiser [23] was applied to the optimisation of the generator and discriminators, with the AdamW optimiser [27] used for optimisation of the denoising model. To promote convergence of the discriminator networks, the R_1 regularisation term in Eq. (6) was applied every three discriminator iterations, where the hyperparameter γ is set to 10. The training time was 60h using an NVIDIA RTX A4000 GPU with 16GB of VRAM, alongside an Intel Core i7-12700K CPU operating at 3.6GHz with 32GB of RAM.

3.3 Performance metrics

The performance of the model is assessed against several image quality metrics. The peak signal-to-noise ratio (PSNR) is applied to the error between the DDPM input ($\mathbf{Z}$) and the denoised output ($\hat{\mathbf{Z}}$), calculated as:

$$\text{PSNR} = 10\log\left(\frac{I_{max}^2}{\text{MSE}}\right), \tag{21}$$

where I_{max} is the maximum extent of the data (2 in this work), and MSE is the mean squared error of the data. Consequently, as MSE is already calculated, it is also used as a metric for image quality. It should be noted that the PSNR metric for radiometric calculations may be inherently erroneous due to I_{max} being dependent on the optical train and observed flux in the available data. The MSSIM is used to assess perceptual accuracy of the model outputs, i.e. how well it compares by eye between the original and reconstructed versions.

Model size and inference time are non-loss-based metrics that are used as metrics for model performance. The model size is calculated as the number of trainable parameters, while the inference time is calculated as an average time per image for inference.

3.4 Interpretability of model

Interpretability is a widespread concern in the dependability of DNN models for physics-based applications due to the black-box nature of these models. For assessing model interpretability, the Gradient-weighted Class Activation Method (Grad-CAM) method [37] was applied to investigate the activations of the decoder/generator G in Fig. 4. Grad-CAM backpropagates the model output, rather than the loss, and evaluates the activations of each specified layer, weighted by the gradients of the output; this enables a recreation of a "heatmap" of the key components of the input that the model focusses most on for recreation. Stepping through the layers of the decoder allows analysis of the contributions spatially to the reconstruction. In the case of a conditional decoding step, as in our model, this is especially useful to determine the

influence of the conditional input compared to the sampled latent-space input on the generated image.

Formally, Grad-CAM is predicated on the assumption that the final model output for class c, denoted by y^c, can be expressed as:

$$y^c = \sum_k \alpha_k^c \sum_{i,j} A_{ij}^k, \tag{22}$$

where k indexes the activation layers, i, j index the width and height dimensions of the k-th layer, and A_{ij}^k is the feature-map activation at coordinates (i, j) of layer k. The sum $\sum_{i,j} A_{i,j}^k$ expresses the global-average-pooled result of the activations in layer k. In Eq. (22), α_k^c is the class-c layer-k neuron importance weight, defined as:

$$\alpha_k^c = \frac{1}{N_k} \sum_{i,j} \frac{\partial y^c}{\partial A_{ij}^k}, \tag{23}$$

where N_k is the number of pixels in layer k. In Grad-CAM, the final class-discriminative saliency map, which is typically visualised as a "heatmap", is defined as:

$$L_{ij}^c = \text{ReLU}\left(\sum_k \alpha_k^c A_{ij}^k\right). \tag{24}$$

By stepping through the blocks within the DNN model, evaluation of the contributions of the different components to the final output can be visually assessed.

4. Results and Discussion

4.1 Model performance

The generative capability of the GAN was tested using the testing portion of the dataset. Fig. 5 shows a series of example image inputs across the six hyperspectral bands (see Section 3.1), and the recreation from the conditional GAN both with and without use of the diffusion model. The scene contains clouds, cloud shadows, water, and a mixture of land types. A similar, but more spatially complex scene is shown in Fig. 6. Qualitatively, similarity between the original image and the recreation is significant, with few observable aberrations, suggesting that the model is well conditioned for HSI reconstruction. In fact, in the cloudy bands, recreation quality appears to capture some of the background features more readily, where these features may typically be hidden in the purely hyperspectral images. Fine-grained features are well characterised and predicted using the VQGAN, however, some smoothing is observed throughout. The smoothing is particularly prevalent in regions of very low signal, likely due to the limited data exposure in these regions.

The results of the latent-space diffusion in both Fig. 5 and Fig. 6 show high-quality recreation of HSIs across all six channels. Comparing the generated images in those figures, sampling from the latent-space distribution does degrade some of the semantic features of the data. Particularly, edges are produced intermittently, which suggests features that are not physical. Despite those features, there is accurate reconstruction of the hyperspectral bands with a significantly lighter-weight model than HyperLDM. Further sets of example images are presented in Appendix A.

Fig. 7 shows ten sampled spectra at four different locations from the diffusion model compared to the ground truth value. The spread in the data suggests that the diffusion model is capable of sampling with high repeatability within an appropriate data range across multiple locations within the scenes. Although there is some underprediction of the 1st and 2nd bands, the relative error between the values is on a similar order of magnitude to the measured radiances.

Table 1 summarises the values of the performance metrics for a subset of the model variants (see Appendix C for a full list) in terms of MSE, mean absolute error (MAE), mean spectral angle mapper (SAM), mean MSSIM, and mean PSNR for the test set. Comparing the use of the VQGAN as a conditional autoencoder as opposed to a conditional VQGAN-DDPM, there is some reduction in the PSNR and MSE metrics, however, the resulting model still outperforms other model variants in GAN-only mode.

4.2 Class activation mapping results

The principles of Grad-CAM were applied to the decoder layers of the VQGAN to highlight the differences in contributions between the latent-space code and the conditional input. Fig. 8 shows the average heatmaps at the output of several layers in the conditional input pathway of the decoder/generator. The heatmaps are scaled from zero to unity. Alongside the heatmaps is a normalised image of the conditional input associated with the heatmap, in the NIR band. Contributions from the input convolution process, as well as the first sequence of C3k2 blocks are predominantly in the fine details of the scene for reconstruction. At the deepest conditional input layer, C3k2 Block 4, larger structural features of the scene are extracted. In particular, the land surrounding the lake is highlighted, as is the vegetative region in the lower right quadrant. Interestingly, a large portion of the scene is ignored near the centre of the scene, corresponding partially to the predominantly uniform region of the input image.

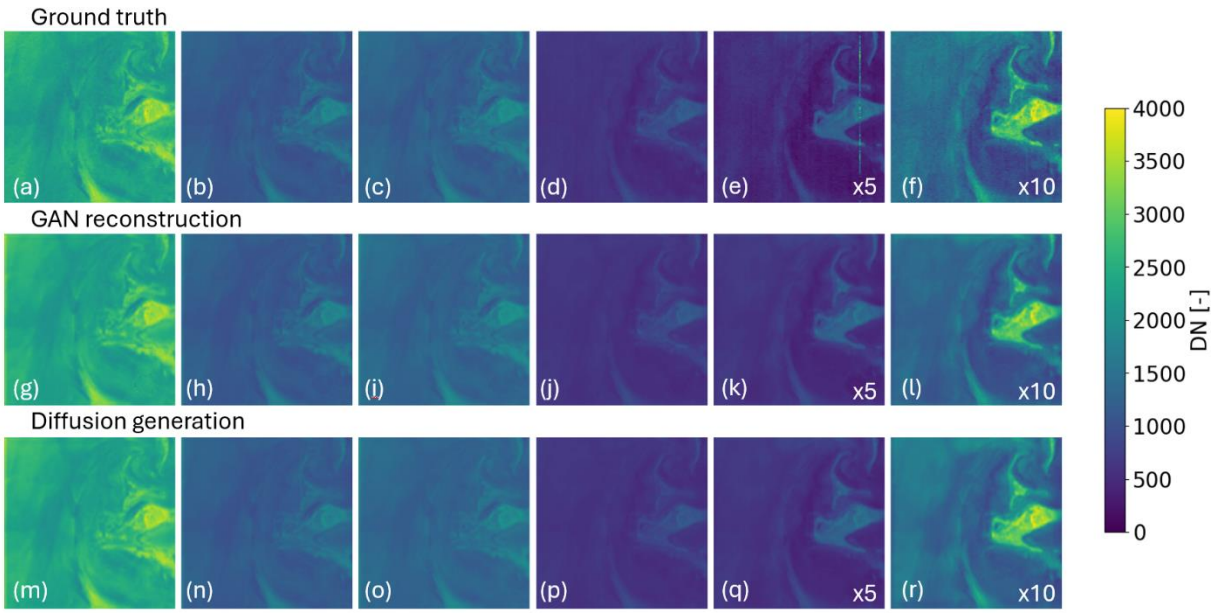

Fig. 5: Example array of cloudy images showing the original hyperspectral scene across the six bands (a–f), a reconstruction using only VQGAN (g–l), and a reconstruction using VQGAN + DDPM (m–r).

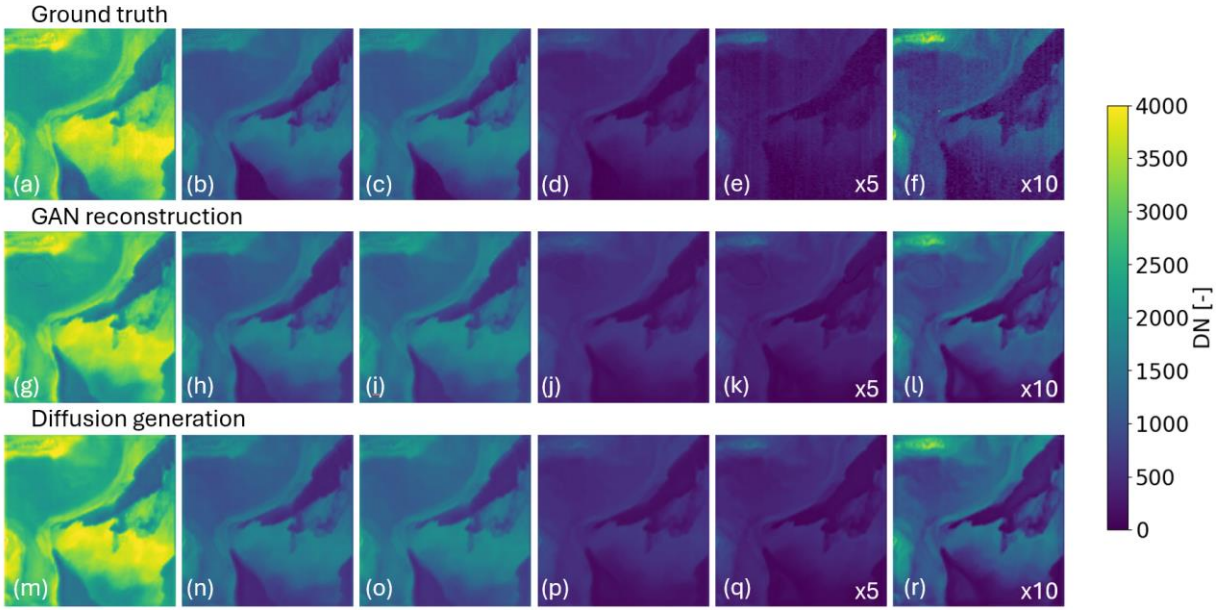

Fig. 6: Example array of complex background images showing the original hyperspectral scene across the six bands (a–f), a reconstruction using only VQGAN (g–l), and a reconstruction using VQGAN + DDPM (m–r).

Table 1: Performance metrics for VQGAN and DDPM, including several model variants. Variants are summarised in Appendix C. Down arrows (↓) indicate smaller results are better. Up arrows (↑) indicate larger results are better.

Model	Metric						
	MSE ↓	MAE ↓	SAM ↓	MSSIM ↑	PSNR ↑ (dB)	# Params ↓ (M)	Inference time (s per image)
Model 15	0.000925	0.0205	0.0198	0.921	36.36	3.34	-
Model 33	0.177	0.381	0.279	0.631	13.53	12.82	-
Model 38	0.000946	0.0203	0.0195	0.865	36.26	**3.27**	-
Model 39 (final)	**0.000616**	**0.0166**	**0.0197**	0.943	**38.12**	3.28	-
Model 39 DDPM	0.000871	0.0193	0.0233	0.886	36.62	5.59	0.146
Model 40	0.000805	0.0182	0.0199	**0.956**	36.96	3.28	-
Model 40 DDPM	0.000784	0.0199	0.0230	0.881	37.08	5.59	**0.113**

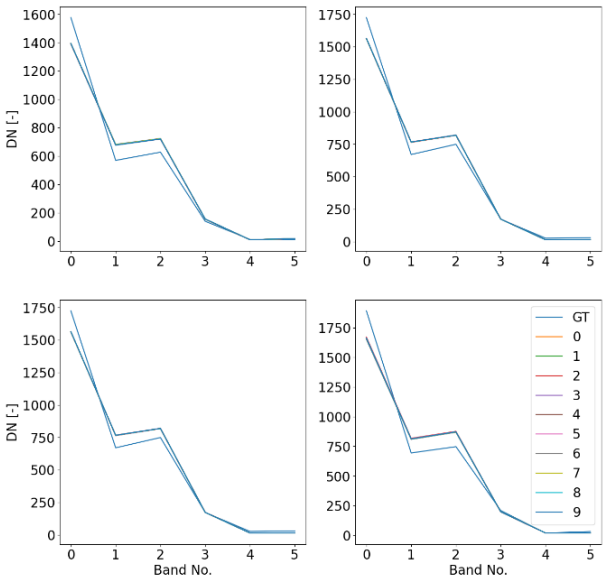

Fig. 7: Comparison of randomly sampled spectra from the diffusion process over ten samples, vs the ground truth at four random pixel locations.

Fig. 9 shows the average heatmap outputs of several layers leading up to the final decoded image. At the deepest layer of the decoder, only the lake region has significant activation from the latent code, complementing the conditional input activation largely ignoring that region. After concatenation of the deep conditional inputs, activation of regions external to the lake are more prominent, however, still limited. Outputs from the shallow conditional C3k2 block show a significant increase in the fine-scale detail representation. This is in agreement with the expected behaviour of the model, as the conditional input skip connects are applied to ensure high spatial fidelity is recovered. The activations of the C3k2 Block 6 consolidate the major feature of the lake, and refine the small contrastive details.

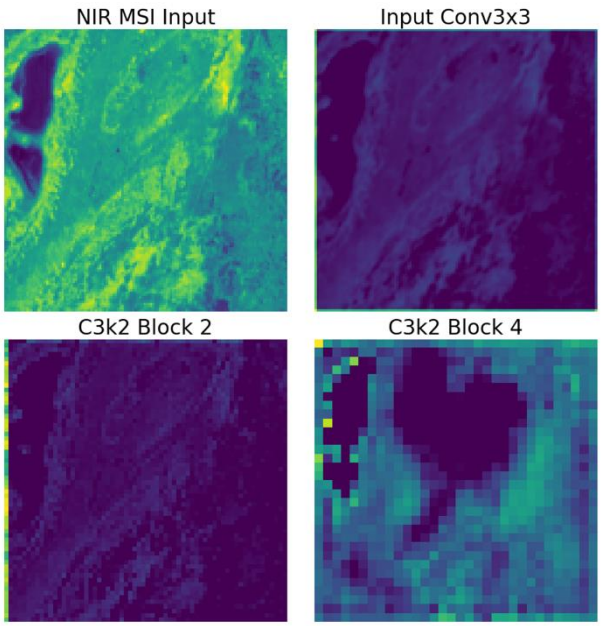

Fig. 8: Heatmaps of activations from several blocks in the conditional input pathway of the decoder/generator for an example scene, as well as the multispectral input in the NIR band.

Similar observations can be made for other inputs images, with these supporting figures included in Appendix B. Predominantly, fine-scale features are supplied by the conditional input, and propagated through the model, whereas large structural features are captured by the latent-space deep layers.

5. Conclusions

To detect high-speed vehicles (e.g., re-entry vehicles) using space-based multispectral/hyperspectral sensors, there is a need to synthesise satellite images containing these vehicles' spectral signatures to train DL-based object detection models. To this end, this paper proposes a method for synthesising satellite images with the

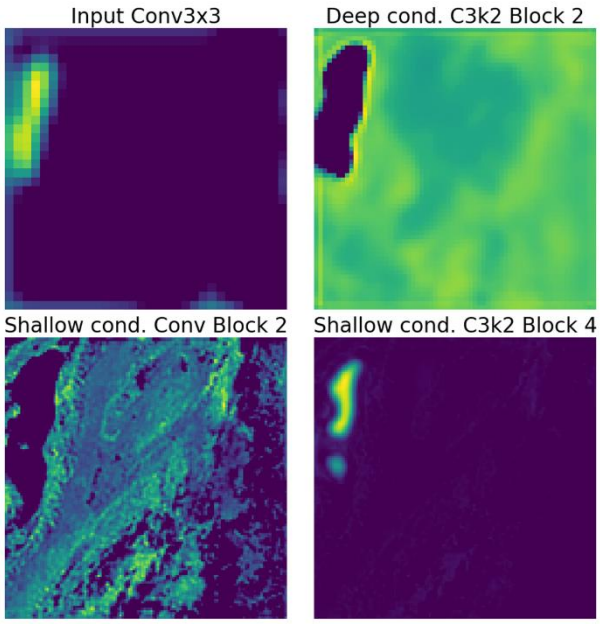

Fig. 9: Heatmaps of activations from select layers of the latent code decoding blocks for an example scene. "Deep cond." and "shallow cond." refer to where there is a skip connection from the conditional input pathway at a deep layer or a shallow layer of the decoder.

desired spectral bands, so that vehicle spectral signatures can be superimposed on these images later for model development. The proposed method is an adaptation of the HyperLDM hyperspectral super-resolution model for rapid generation of SWIR and MWIR data. The model and implementation are significantly more lightweight than the original HyperLDM. The implementation performs similarly well in terms of reconstruction losses and spectral accuracy. The model also has the advantage of not requiring a separate spectral library for endmember matching, as the model is capable of directly predicting the observed radiance from the input MSI.

Operating as a GAN, the proposed model results in low reconstruction error, with high spectral fidelity. When integrated with a diffusion model for latent space sampling, the model retains high quality reconstruction of the HSIs. All metrics were fractionally degraded through the DDPM process, however, the fully conditional generated samples outperform previous model iterations operating in autoencoder modes. The final model, namely Model 39 in Table 1 and Appendix C, can reliably generate spatially and spectrally accurate HSIs using only an MSI input.

Analysis of the embedded model decision making process through the Grad-CAM method demonstrated the importance of the conditional input for fine-scale reconstruction. The latent-space code captures large-scale features, such as bodies of water, forest areas, or areas with large contrastive components (e.g., coastlines). Comparatively, pixel-wise contrastive features are captured via the skip connections from the conditional input, allowing high-fidelity fine-scale reconstruction of the HSIs.

Acknowledgements

This project, iLAuNCH Project H-3, is supported by the Australian Government Department of Education through the Trailblazer Universities Program and the Research Training Program Stipend (RTPS).

Appendix A (Selection of example synthesised images)

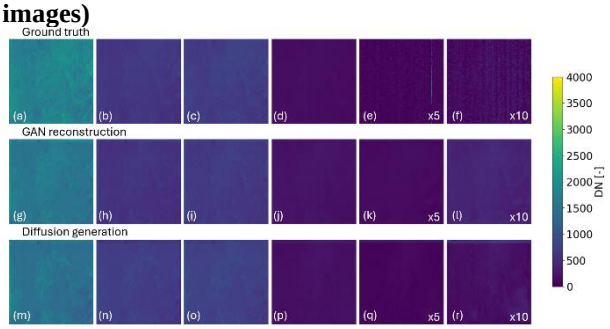

Fig. 10: Example of a low-SNR image reconstruction.

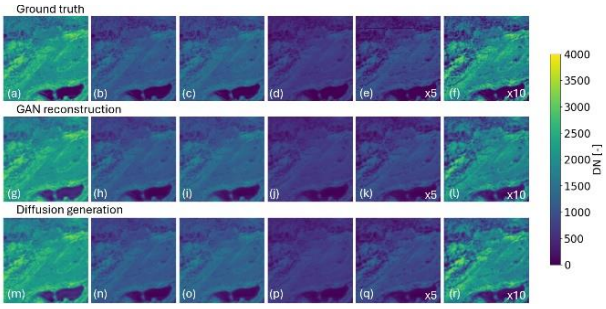

Fig. 11: Example of a high-SNR image reconstruction.

Appendix B (Grad-CAM output examples)

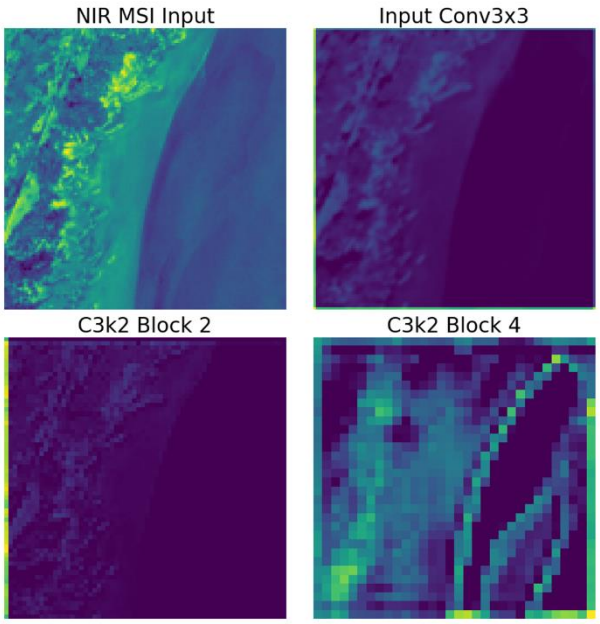

Fig. 12: Heatmaps of activations from several blocks in the conditional input pathway of the decoder/generator for an example scene, as well as the multispectral input in the NIR band.

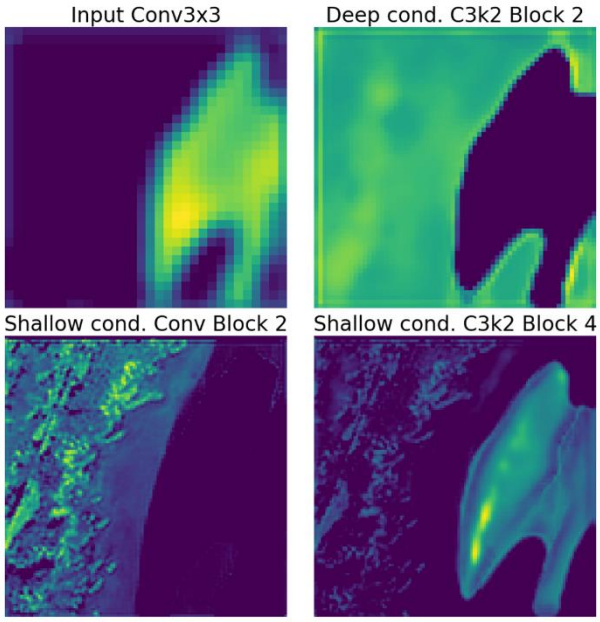

Fig. 13: Heatmaps of activations from select layers of the latent code decoding blocks for an example scene. Deep cond. and shallow cond. refer to where there is a skip connection from the conditional input pathway at a deep layer or a shallow layer of the decoder.

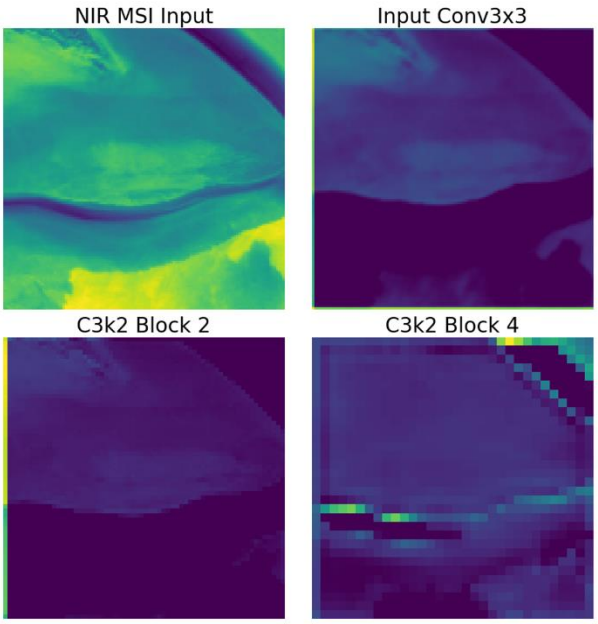

Fig. 14: Heatmaps of activations from several blocks in the conditional input pathway of the decoder/generator for an example scene, as well as the multispectral input in the NIR band.

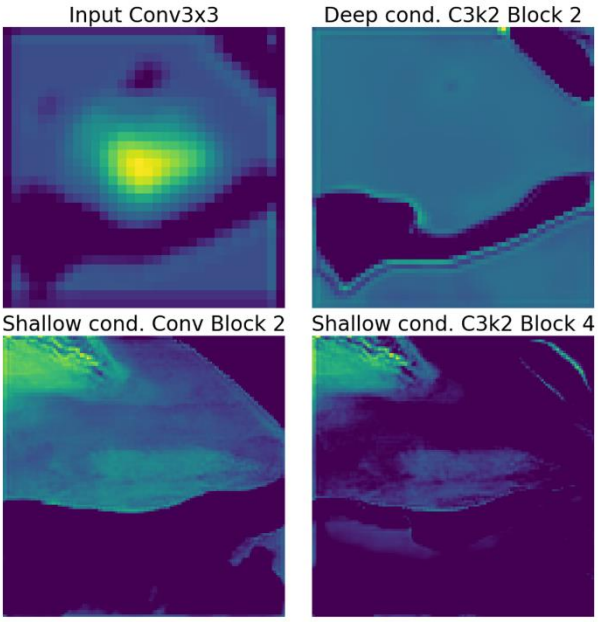

Fig. 15: Heatmaps of activations from select layers of the latent code decoding blocks for an example scene. Deep cond. and shallow cond. refer to where there is a skip connection from the conditional input pathway at a deep layer or a shallow layer of the decoder.

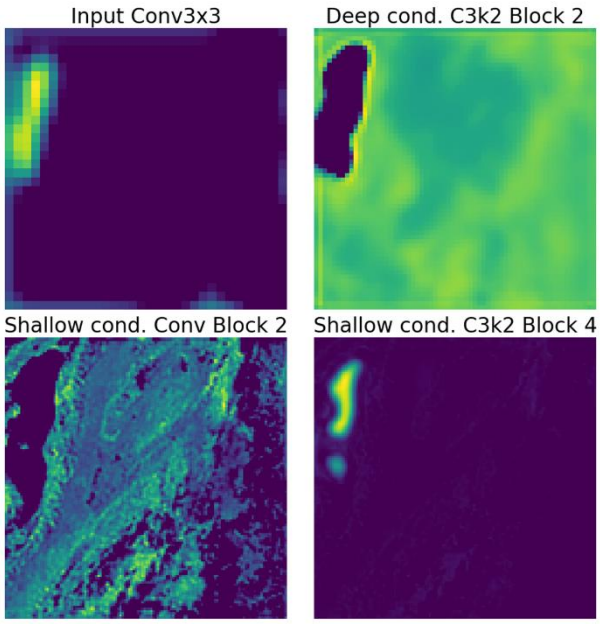

Fig. 16: Heatmaps of activations from select layers of the latent code decoding blocks for an example scene.

Appendix C (Summary of model variations)

Model	Encoding depth	Model base channels	Activation function	Decoding depth	Channel attention	Embedding dimensions	Latent channels	Spectral discr. layers	Spec discr. channels	Spec discr. points	Spat discr. layers	Spat discr. Channels
1	4	32	SiLU	2	N	1024	16	6	128	16	5	64
2	5	16	SiLU	2	Y	-	-	6	128	16	5	64
3	5	8	SiLU	2	Y	-	-	6	128	16	5	64
4	5	8	SiLU	2	Y	1024	16	6	128	16	5	64
5	5	8	SiLU	2	N	512	16	6	128	32	4	64
6	5	16	SiLU	2	N	512	16	6	128	32	4	64
7	5	8	SiLU	2	Y	512	16	6	128	32	6	64
8	5	8	LeakyReLU(0.2)	2	Y	512	16	6	128	32	6	64
9	5	16	LeakyReLU(0.05)	2	Y	512	16	6	128	32	4	128
10	5	16	LeakyReLU(0.05)	4	Y	-	-	6	128	32	5	128
11	5	16	LeakyReLU(0.05)	4	Y	512	16	6	128	32	5	128
12	5	16	LeakyReLU(0.05)	2	Y	512	16	6	128	32	5	128
13	6	16	LeakyReLU(0.05)	2	Y	512	16	6	128	32	5	128
14	5	32	LeakyReLU(0.05)	2	Y	512	16	6	128	32	5	128
15	4	32	LeakyReLU(0.05)	2	Y	512	16	6	128	32	5	128
16	4	16	LeakyReLU(0.05)	2	Y	512	16	6	128	32	5	128

17	4	16	LeakyReLU(0.05)	2	Y	512	16	6	128	32	5	64
18	5	16	LeakyReLU(0.05)	2	Y	1024	16	6	128	32	5	64
19	5	32	LeakyReLU(0.05)	2	Y	1024	16	6	128	32	5	64
20	5	16	LeakyReLU(0.05)	3	Y	1024	32	6	128	32	5	64
21	5	16	SiLU	3	Y	-	-	5	128	16	4	64
22	5	16	SiLU	2	Y	-	-	5	128	16	4	64
23	5	16	LeakyReLU(0.05)	2	Y	512	16	5	128	16	4	64
24	5	16	LeakyReLU(0.05)	2	N	1024	16	5	128	16	4	64
25	5	16	LeakyReLU(0.05)	2	N	-	-	5	128	16	4	64
26	5	8	LeakyReLU(0.05)	2	N	-	-	5	128	16	4	64
27	5	8	LeakyReLU(0.05)	2	Y	1024	16	5	128	16	4	32
28	5	16	LeakyReLU(0.05)	2	Y	512	64	5	64	16	4	32
29	5	16	LeakyReLU(0.05)	2	Y	512	64	5	64	16	4	32
30	4	32	SiLU	2	Y	1024	16	6	128	16	5	64
31	5	8	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32
32	5	8	LeakyReLU(0.05)	2	Y	512	32	5	64	16	4	32
33	5	16	LeakyReLU(0.05)	2	Y	1024	32	5	64	16	4	32
34	6	16	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32
35	5	16	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32
36	5	16	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32
37	5	16	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32
38	4	32	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32
39	4	32	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32
40	4	32	LeakyReLU(0.05)	2	Y	512	16	5	64	16	4	32

Note: Models 36-40 involved variations to data normalisation.

References

[1] Audebert, N., Le Saux, B., Lefevre, S., 2019. Deep Learning for Classification of Hyperspectral Data: A Comparative Review. IEEE Geoscience and Remote Sensing Magazine 7, 159–173. https://doi.org/10.1109/MGRS.2019.2912563

[2] Bateson, A., Curtiss, B., 1996. A method for manual endmember selection and spectral unmixing. Remote Sensing of Environment 55, 229–243. https://doi.org/10.1016/S0034-4257(95)00177-8

[3] Bishop, C.M., Bishop, H., 2024. Deep Learning: Foundations and Concepts. Springer Cham. https://doi.org/10.1007/978-3-031-45468-4
[4] Casella, G., Robert, C.P., 1996. Rao-Blackwellisation of sampling schemes. Biometrika 83.
[5] Chen, B., Liu, L., Liu, C., Zou, Z., Shi, Z., 2024. Spectral-Cascaded Diffusion Model for Remote Sensing Image Spectral Super-Resolution. IEEE Transactions on Geoscience and Remote Sensing 62, 1–14. https://doi.org/10.1109/TGRS.2024.3450874
[6] Croitoru, F.-A., Hondru, V., Ionescu, R.T., Shah, M., 2023. Diffusion Models in Vision: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 45, 10850–10869. https://doi.org/10.1109/TPAMI.2023.3261988
[7] Dahlgren, M., Karako, T., 2023. Getting on Track: Space and Airborne Sensors for Hypersonic Missile Defense (a report of the CSIS Missile Defence Project).
[8] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: CVPR. pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
[9] Dhariwal, P., Nichol, A., 2021. Diffusion Models Beat GANs on Image Synthesis, in: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 8780–8794.
[10] Du, Y., Li, X., Shi, L., Li, F., Yuan, S., 2022. Optimizing Spectral Waveband Selection for Spectral Radiation Detection of Hypersonic Vehicle. IEEE Transactions on Plasma Science 50, 4683–4692. https://doi.org/10.1109/TPS.2022.3208925
[11] Earth Resources Observation and Science (EROS) Center, 2019. USGS EROS Archive - Earth Observing One (EO-1) - Hyperion. https://doi.org/10.5066/P9JXHMO2
[12] Esser, P., Rombach, R., Ommer, B., 2021. Taming Transformers for High-Resolution Image Synthesis, in: 2021 CVPR. pp. 12868–12878. https://doi.org/10.1109/CVPR46437.2021.01268
[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets, in: Advances in Neural Information Processing Systems. Curran Associates, Inc.
[14] Ho, J., Jain, A., Abbeel, P., 2020. Denoising Diffusion Probabilistic Models, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 6840–6851.
[15] Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E., 2020. Squeeze-and-Excitation Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 42, 2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372
[16] Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-To-Image Translation With Conditional Adversarial Networks, in: CVPR. pp. 5967–5976. https://doi.org/10.1109/CVPR.2017.632
[17] Jarecke, P., Barry, P., Pearlman, J., Markham, B., 2001. Aggregation of Hyperion hyperspectral spectral bands into Landsat-7ETM+ spectral bands, in: IEEE IGARSS 2001, pp. 2822–2824. https://doi.org/10.1109/IGARSS.2001.978175
[18] Jia, J., Chen, J., Zheng, X., Wang, Y., Guo, S., Sun, H., Jiang, C., Karjalainen, M., Karila, K., Duan, Z., Wang, T., Xu, C., Hyyppä, J., Chen, Y., 2022. Tradeoffs in the Spatial and Spectral Resolution of Airborne Hyperspectral Imaging Systems: A Crop Identification Case Study. IEEE Transactions on Geoscience and Remote Sensing 60, 1–18. https://doi.org/10.1109/TGRS.2021.3096999
[19] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T., 2020. Analyzing and Improving the Image Quality of StyleGAN, in: 2020 CVPR. pp. 8107–8116. https://doi.org/10.1109/CVPR42600.2020.00813
[20] Ke, C., 2017. Military object detection using multiple information extracted from hyperspectral imagery, in: 2017 International Conference on Progress in Informatics and Computing. pp. 124–128. https://doi.org/10.1109/PIC.2017.8359527
[21] Khanam, R., Hussain, M., 2024. YOLOv11: An Overview of the Key Architectural Enhancements. https://doi.org/10.48550/arXiv.2410.17725
[22] Kherif, F., Latypova, A., 2020. Chapter 12 - Principal component analysis, in: Mechelli, A., Vieira, S. (Eds.), Machine Learning. Academic Press, pp. 209–225. https://doi.org/10.1016/B978-0-12-815739-8.00012-2
[23] Kingma, D.P., Ba, J., 2015. Adam: A Method for Stochastic Optimization, in: ICLR. https://doi.org/10.48550/arXiv.1412.6980
[24] Kingma, D.P., Welling, M., 2014. Auto-Encoding Variational Bayes, in: ICLR. https://doi.org/10.48550/arXiv.1312.6114
[25] Liu, L., Chen, B., Chen, H., Zou, Z., Shi, Z., 2023. Diverse Hyperspectral Remote Sensing Image Synthesis With Diffusion Models. IEEE Transactions on Geoscience and Remote Sensing 61, 1–16. https://doi.org/10.1109/TGRS.2023.3335975
[26] Liu, L., Li, W., Shi, Z., Zou, Z., 2022. Physics-Informed Hyperspectral Remote Sensing Image Synthesis With Deep Conditional Generative Adversarial Networks. IEEE Transactions on Geoscience and Remote Sensing 60, 1–15. https://doi.org/10.1109/TGRS.2022.3173532

[27] Loshchilov, I., Hutter, F., 2019. Decoupled Weight Decay Regularization, in: ICLR. https://doi.org/10.48550/arXiv.1711.05101

[28] Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models, in: Proceedings of the 30th International Conference on Machine Learning.

[29] Martinez, E., Jacome, R., Hernandez-Rojas, A., Arguello, H., 2023. LD-GAN: Low-Dimensional Generative Adversarial Network for Spectral Image Generation with Variance Regularization, in: 2023 CVPRW. IEEE Computer Society, pp. 265–275. https://doi.org/10.1109/CVPRW59228.2023.00032

[30] Meerdink, S.K., Hook, S.J., Roberts, D.A., Abbott, E.A., 2019. The ECOSTRESS spectral library version 1.0. Remote Sensing of Environment 230, 111196. https://doi.org/10.1016/j.rse.2019.05.015

[31] Mescheder, L., Geiger, A., Nowozin, S., 2018. Which Training Methods for GANs do actually Converge?, in: Dy, J., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research. PMLR, pp. 3481–3490.

[32] Mirza, M., Osindero, S., 2014. Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784.

[33] NASA, 2025. Spectral Response of the Operational Land Imager In-Band, Band-Average Relative Spectral Response [WWW Document]. Landsat Science. URL https://landsat.gsfc.nasa.gov/satellites/landsat-8/spacecraft-instruments/operational-land-imager/spectral-response-of-the-operational-land-imager-in-band-band-average-relative-spectral-response/ (accessed 08.09.25).

[34] Ozkan, S., Kaya, B., Akar, G.B., 2019. EndNet: Sparse AutoEncoder Network for Endmember Extraction and Hyperspectral Unmixing. IEEE Transactions on Geoscience and Remote Sensing 57, 482–496. https://doi.org/10.1109/TGRS.2018.2856929

[35] Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer International Publishing, Cham, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

[36] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X., 2016. Improved Techniques for Training GANs, in: Advances in Neural Information Processing Systems. Curran Associates, Inc.

[37] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. International Journal of Computer Vision 128, 336–359. https://doi.org/10.1007/s11263-019-01228-7

[38] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, in: Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research. PMLR, pp. 2256–2265.

[39] Suresh, S., Arun, P.V., Porwal, A., 2023. Unmixing in latent space: A novel unsupervised approach for geological mapping of lunar surface, in: 2023 IEEE India Geoscience and Remote Sensing Symposium (InGARSS). pp. 1–4. https://doi.org/10.1109/InGARSS59135.2023.10490382

[40] Votta, R., Schettino, A., Bonfiglioli, A., 2013. Hypersonic high altitude aerothermodynamics of a space re-entry vehicle. Aerospace Science and Technology 25, 253–265. https://doi.org/10.1016/j.ast.2012.02.001

[41] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13, 600–612. https://doi.org/10.1109/TIP.2003.819861

[42] Wu, C., Wang, D., Bai, Y., Mao, H., Li, Y., Shen, Q., 2023. HSR-Diff: Hyperspectral Image Super-Resolution via Conditional Diffusion Models, in: ICCV. pp. 7083–7093.

[43] Yuan, Z., Hao, C., Zhou, R., Chen, J., Yu, M., Zhang, W., Wang, H., Sun, X., 2023. Efficient and Controllable Remote Sensing Fake Sample Generation Based on Diffusion Model. IEEE Transactions on Geoscience and Remote Sensing 61, 1–12. https://doi.org/10.1109/TGRS.2023.3268331

[44] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric, in: CVPR, pp. 586–595. https://doi.org/10.1109/CVPR.2018.00068