Optimal background removal using denoising diffusion models

Ingo P. Waldmann
Spaceflux Ltd[*]
Marco Rocchetto, Marcel Debczynski
Spaceflux Ltd

ABSTRACT

In this paper we will discuss the use of generative deep learning models to learn the local detector characteristics, astronomical scene and weather patterns to produce adaptive background removal and optimally suppress noise in SSA images. Our approach is based on recent advances in deep learning and in particular the use of generative models to de-noise existing images.

The generative network is learned on the focal plane images derived from our Spaceflux optical sensor arrays and fine-tuned to the individual detector characteristics. A significant advantage of denoising diffusion generative models is their ability to accurately model non-linear relations within images. This characteristic can be exploited to represent astronomical backgrounds, time variable flat fielding (e.g. dawn/dusk gradients) and the effects of seeing and partial cloud coverage in one uniform model. We demonstrate the ability of Denoising Diffusion Neural Network (DDNN) to mimic real observations at a photorealistic level. These simulations can then be used to, for example, de-trend SSA observations with high-background levels.

1. INTRODUCTION

The field of space situational awareness (SSA) has seen significant growth in recent years, driven by the increasing number of objects in orbit around the Earth and the need to track and monitor their movements. Accurate detection and tracking of these objects is critical for the safety and security of space operations. However, the detection of faint objects is challenging due to the presence of noise in the images, which can make it difficult to distinguish between real signals and background noise.

Improvements in the quality and reliability of SSA observations, ultimately contribute to better space situational awareness and space traffic management. Furthermore, optimal noise suppression in SSA images allows for the detection of smaller and fainter targets, a more robust detection during sub-optimal weather conditions and more accurate absolute photometry. Improvements in photometric calibration lead to a better ability to characterise the resident space objects' (RSOs) morphologies and time-dependent behaviour through the study of their reflected light. Optimal photometry is also particularly important in low signal-to-noise conditions, such as the detection of small RSOs in high-altitude orbits or eccentric cis-lunar orbits.

Today, much of the machine learning and general AI literature in the SSA field has focused on the detection of signals (see [10] for a recent review) rather than the optimal removal of noise. In low-signal regimes such as high-altitude orbits or the detection of very small satellites, optimal noise suppression is an important prerequisite to improving satellite detection fidelity.

Machine learning provides a novel approach to optimally learn the detector, atmospheric and astronomical scene characteristics from a a set of observational data without prior knowledge of the system. Architectures such as Noise2Self [1], Generative Inpainting (e.g. [17]) and denoising diffusion networks [5] find broad applicability in the fields of image processing and image restoration.

In this paper, we demonstrate the use of generative Denoising Diffusion Neural Networks (DDNNs) to learn an optimal non-Gausssian noise suppression in observational images. We will demonstrate that both background star removal,

[*] Spaceflux Ltd, 71-75 Shelton Street, WC2H 9JQ, London, United Kingdom

flat-fielding and non-linear image distortions can be achieved simultaneously with the same algorithm and that no prior knowledge of the instrument, astronomical scene or target satellite is required. We will focus on the detection of faint geostationary (GEO) RSOs in staring-mode (i.e. no sideral tracking) observations. We note that this methodology can easily be adapted to any orbital regimes and observational setups.

2. TRAINING DATA

Before introducing the method in the section below, we here briefly provide a description of the data obtained to train the DDNN.

We have obtained LEO and GEO observations using our Spaceflux[1] USA1 telescope.

Spaceflux is UK-based SSA company, operating a global network of sensors with 10 locations worldwide expanding to more than 20 by the end 2024. The company is focused on accurate, real-time satellite and space debris tracking from LEO to GEO.

This study utilises data from one such site in New Mexico. The sensor used was a 35-cm F/3 telescope, equipped with a 4096x4096 pixels sized scientific-grade CMOS camera, and a resolution of 0.75 arcsec/px. Observations of a sample of GEO satellites was obtained on the night of the 14th of April 2023. Exposure times varied throughout the night but were typically kept in the 0.1 - 2 second range. For this study, we sub-sampled this data to obtain 4000 subframes with a size of 128x128 pixels.

We applied a ZScale [2] stretch and normalisation to the data, followed by random cropping to 256×256 pixel images. Please note that we did not apply any bias corrections, flat-fielding or other data processing. We specifically omit these standard steps to test whether our DDNN methodology can learn all instrument defects and behaviours from the observed data themselves without any further processing. Figure 3 shows examples of our obtained data product. We also note that while longer exposure times are possible, background-star streaks become excessively long and image stacking is usually preferred (e.g. [18]). We manually selected 1000 RSO-free images containing streaks and background noise. Though the inclusion of RSOs in our training data is likely not detrimental to the algorithmic convergence, we have opted to proceed with a RSO-free sample in this paper.

3. DIFFUSION NEURAL NETWORKS

In order to learn the underlying instrument noise and stellar streaks, we opt for a Denoising Diffusion Neural Network. DDNNs are part of a larger suite of generative AI models built to learn the unknown likelihood distribution of a training data set without the need to parameterise said likelihood. In contrast to Generative Adversarial Networks (GANs, [3]), diffusion models have proven to be significantly more stable in training, and have been shown to achieve state-of-the-art photo-realistic synthesis of image data [5, 13, 12], outperforming techniques such as Normalising Flows [7] and Variational Autoencoders [6]. Diffusion network modelling has received very significant attention in recent years and we refer the interested reader to two excellent recent review papers [2] & [16]. In this paper, we will train a denoising diffusion neural network based on the description presented in [15] with an additional conditioning on observed data during inference [9].

3.1 Background

Here we will briefly outline how DDNNs are trained. Please note that this is a high-level description and we refer the interested reader to the cited literature for further details.

First we define the forward diffusion process. The forward diffusion process transforms a data set with the probability distribution $x_0 \sim q(X_0)$ to a Gaussian noise probability distribution

$$q(X_T) := \mathcal{N}(0, I) \tag{1}$$

where $\mathcal{N}(\mu, I)$ is a multivariate Normal distribution. This transformation happens incrementally over T time steps from [0 ... T]. The point of this gradual degradation is that the transformation from $x_t \rightarrow x_{t+1}$ can be learned, and once

[1] https://spaceflux.io/optical-sensor-network/

[2] https://docs.astropy.org/en/stable/api/astropy.visualization.ZScaleInterval.html

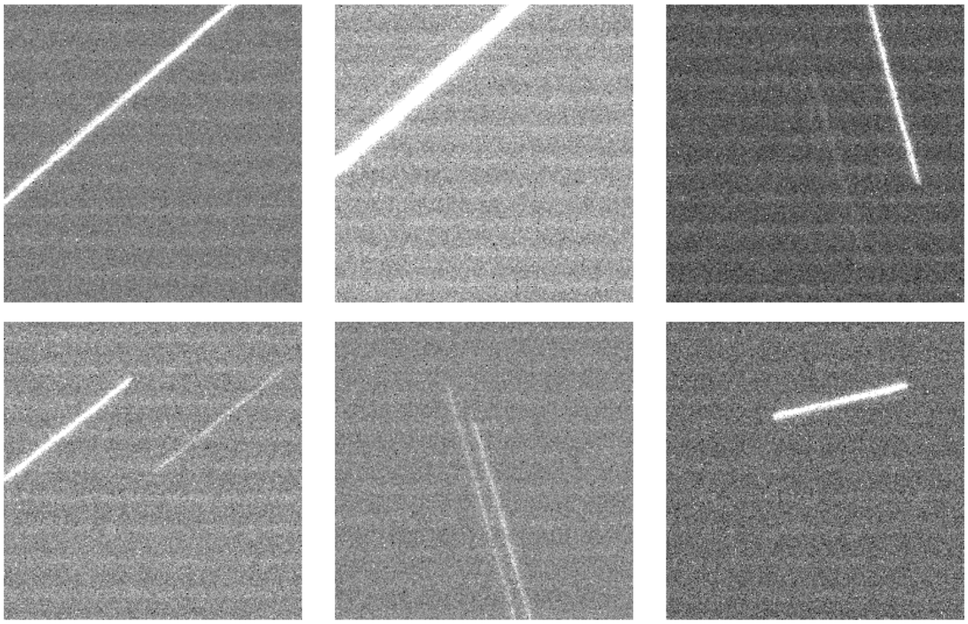

Fig. 1: Examples of observed data after zscale normalisation and cropping. The bright diagonal streaks are background stars whilst the horizontal banding is CMOS banding noise.

learned it can later on be reversed. In other words, we are learning a mapping from data space $q(x_0)$ to a Gaussian distribution. We denote this incremental process with the forward model kernel $q(x_t|x_{t-1})$. This forward diffusion process can now be described as a Markov chain of the form

$$q(x_1,\ldots,x_T|x_0)=\prod_{t=1}^{T}q(x_t|x_{t-1}) \tag{2}$$

with the forward diffusion kernel defined as the Normal distribution

$$q(x_t|x_{t-1}):=\mathcal{N}(x_t;\sqrt{1-\beta_t}x_{t-1},\beta_t I) \tag{3}$$

where β_t is the diffusion rate at time set t and set by the diffusion scheduler as a decreasing sequence $\beta_{1:T}\in(0,1]^T$.

Now that we have defined the forward diffusion process, we will now turn our attention to the reverse process and undo the forward diffusion. Starting from a pure noise distribution $x_T\sim p(x_T):=\mathcal{N}(0,I)$, we can compute the reverse step by

$$p_\theta(x_{t-1}|x_t)=\mathcal{N}(x_{t-1};\mu_\theta(x_t,t),\Sigma_\theta(x_t,t)). \tag{4}$$

As this process is difficult to compute in practice as it requires the integral of all possible reverse trajectories. We hence approximate $p(x_{t-1}|x_t)$ using a neural network with parameters θ. The neural network takes as input a noisy image, x_t, and computes the mean, $\mu_\theta(x_t,t)$, and covariance, $\Sigma_\theta(x_t,t)$, at time step t.

We now define the optimisation function for the neural network. Since the computation of the maximum likelihood is intractable, [5] have defined a variation lower bound optimisation funciont (also see Appendix A in [2]). After simplifying these expressions (not shown here), we can derive the following loss-function

$$L_{simple}=\mathbb{E}_{t\sim[1:T],x_0\sim p(x_0),\varepsilon_t\sim\mathcal{N}(0,I)}\left[||\varepsilon_t-\varepsilon_\theta(x_t,t)||^2\right] \tag{5}$$

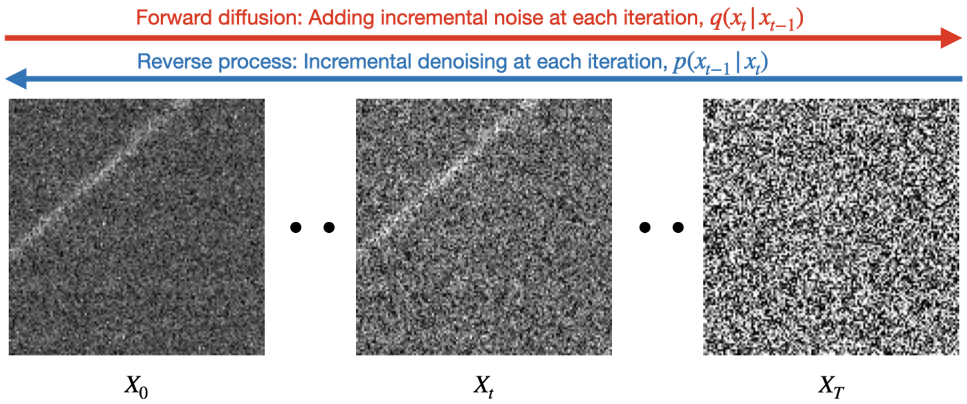
Forward diffusion: Adding incremental noise at each iteration, $q(x_t | x_{t-1})$
Reverse process: Incremental denoising at each iteration, $p(x_{t-1} | x_t)$
X_0
X_t
X_T

Fig. 2: Schematic showing the diffusion forward modelling process (red arrow) and its reverse (blue arrow). During training, the algorithm is trained using the forward process. During production, we use the inverse process to de-trend our data. During training, we incrementally add Gaussian noise to our observed raw data (X_0) for T number of steps. We then learn the mapping between X_t and X_{t+1} until convergence. During inference, we compute the incremental inverse steps going from a noise distribution X_T to the data distribution X_0. Note that X_0 in this figure is real, observed data. No data simulations were used in this paper.

where $\varepsilon_\theta(x_t, t)$ is the noise estimate by the neural network for x_t, and e_t is the real noise at step t. In other words, we try to minimise the difference between the real noise in the image at time t and the noise estimated by the neural network. In this implementation, the neural network does not predict the signal at each time step but only the noise.

Once we have calculated $\varepsilon_\theta(x_t, t)$, we subtract this noise estimate from our data (x_t) to obtain x_{t-1} and continue this process until we obtain $p_\theta(x_0)$.

Once the diffusion network has been trained, we can proceed to the inference on real-data. By simply generating from $x_T \rightarrow x_0$ will result in realistic looking images of $q(x_0)$ but not necessarily result in the observed data instances we wish to correct. We therefore need to condition our diffusion network on the observed data. Loosely following the ideas presented in [9], we take our observed data x_0 and progress it partially up the forward modelling chain before reversing the process. In other words, we take our observed image, add just sufficient Gaussian noise to allow the reverse diffusion to reconstruct our data. We have found that optimal results are found for a maximum forward model distance of t = 100-150 (with T = 1000) before reversing the diffusion.

While most image applications have noise free training and inference data, x_0, our data contains intrinsic photon-noise and other sources of random noise. An obvious solution would be to pre-process the data to remove high-frequency noise and only train on ‘cleaned’ images. We decide not to go down this route as any pre-cleaning of the data may distort or even remove faint underlying signals. In the following sections we show that we can derive a ‘noise-free’ map of the systematic signals (stellar streaks, instrument noise, etc) despite training on noisy data.

Each instance of $p_\theta(x_0)$ contains Gaussian noise at the level of the original training data. This is an intrinsic property of the original distribution $q(x_0)$. Given then Gaussian nature of this noise, we can use the central limit theorem to suppress this noise by re-running this forward $\rightarrow$ backward process N times and averaging multiple instances:

$$\bar{p}_\theta(x_0) = \frac{1}{N}\sum^{N} p_{\theta,n}(x_0). \tag{6}$$

Averaging N instances suppresses Gaussian noise as well as averaging out sub-optimal generation instances.

Once $\bar{p}_\theta(x_0)$ has been computed, we can subtract it from our observed data to obtain the noise-corrected residual (see Results for examples).

Down layers (Channels)	Up layers (Channels)
ResNet (128)	ResNet (512)
ResNet (128)	Attention + ResNet (512))
ResNet (256)	ResNet (256)
ResNet (256)	ResNet (256)
Attention + ResNet (512)	ResNet (128)
ResNet (512)	ResNet (128)

Table 1: U-Net architecture used for a 128×128 monochrome input image. Each row corresponds to a layer in the downsample and upsample parts. There are two layers per block (denoted by dashed lines). The lowest blocks contain one attention layer on either side of the U-Net.

3.2 Implementation

We base our implementation on a customised version of the DDPMPipeline[3] based on [5] with a denoising scheduler based on [15]. Our neural network to estimate $p_\theta(x_{t-1}|x_t)$ is a U-Net [14] using three down-sampling and three up-sampling blocks. Each block has two ResNet layers [4] apart from the second-to-last downsample and 2^{nd} upsample layers which have been replaced by attention layers [11]. Table 1 summarises the U-Net architecture.

Due to computational reasons, we opt to downsample our input data to 128×128 images which resulted in 4,000 unique training images[4]. Additionally, we re-normalise our images to unit variance but do not perform any additional pre-processing.

We then proceeded to train our diffusion network over 1000 time steps, T, and 100 epochs. We employed a time-variable learning rate with a cosine annealing [8] and a warm-up period of 500 iterations up to a maximum learning rate of 1e-4.

Once our diffusion network has been successfully trained, we proceed to generating new test-case examples. We prepare new and unseen data in the same way as our training data with the addition of point-like RSOs. We parameterise the RSOs using a 2D Gaussian PSF with a $\sigma = 0.5$ pixel width. We normalise the RSO's PSF to unity Signal-to-Noise, SNR = 1, over the full frame. We therefore simulate ultra-small and faint objects just above the detection limit of the data (e.g. left column in figure 3).

Finally, we ran our diffusion process $N = 20$ times to obtain $\bar{p}_\theta(x_0)$.

4. RESULTS

In figure 3 we demonstrate our diffusion neural network denoising map, $\bar{p}_\theta(x_0)$. We can clearly see that our averaging over N instances of the diffusion process (equation 6) has suppressed the high-frequency noise components and left us with a smooth flatfield with which we can correct our data. We can clearly see in the figures that our algorithm was able to correctly model the stellar streak contamination as well as the CMOS detector banding noise. We stress again that this flatfield was not generated with any additional pre- or post-processing and purely derived from the raw observed data. This generality of solution allows us to model any non-Gaussian and non-linear noise in the data (e.g. instrument optics noise, detector noise, variable sky backgrounds, cloud contamination, etc). The injected low-SNR RSO signal is clearly visible in the residual map and we measured an average increase of SNR in the RSO detection (defined here as peak amplitude at the PSF center over standard deviation of the surrounding pixels) of 60%.

Figure 4 shows the power spectrum as a function of spatial frequencies, k, of the bottom row example in figure 3. Here we can see that a substantial decrease in low-frequency (i.e. large feature) power was achieved across the image. This is to be expected as our DDNN noise map only contains smooth features. We can however also see that a slight reduction in high-frequency power was achieved. We attribute this to slight variations in the CMOS banding noise.

[3] `https://huggingface.co/docs/diffusers/api/pipelines/ddpm`

[4] Downsampling to 128×128 was a pure computational consideration and the algorithm will perform equally well, or better, for larger images. We do however note that images with sizes exceeding 512×512 require a slightly different approach as computational costs are too severe. We suggest an autoencoder compression step in these cases (e.g. [13])

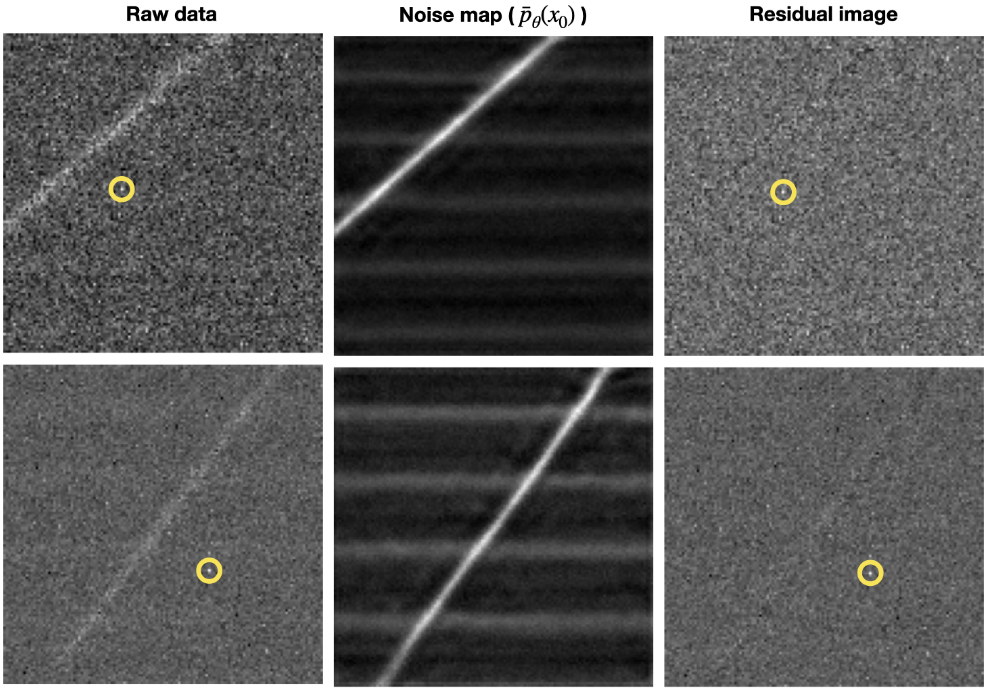

Fig. 3: Showing one output example per row: Left: Normalised raw data with injected RSO marked by yellow circle We note that this is real observed data with the RSO PSF injected. Middle: Derived DDNN map ($\bar{p}_\theta(x_0)$) given raw observed input. The stellar streak as well as the CMOS banding noise can be seen clearly. Right: Residual image after subtracting diffusion map from raw input. The remaining satellite is marked by a yellow circle.

5. DISCUSSION AND CONCLUSION

In this paper we introduce deep learning generative diffusion models to model instrument and astrophysical noise in SSA images. We show that diffusion models can accurately capture instrument systematics as well as accurately model background star streaks without any prior knowledge or assumptions about the data. These data-derived systematic noise maps can then be used to de-trend the observed data or used as analytic tool to better understand and model instrument and environmental noise. Such noise maps are valuable as it will allow for a more optimal stacking on data in ultra-low signal-to-noise regimes and this technique can seamlessly be incorporated in existing image stacking pipelines.

In this paper, we provide a simple proof of concept but do not claim that our hyperparameter optimisation is optimal. With the rapid advances in diffusion modelling and further hyperparameter optimisation, it is likely that future versions of this algorithm will outperform current results. In future pupblications we will address these as well as extend our work to LEO orbital regimes and twilight observations.

ACKNOWLEDGEMENTS

Software presented here was written in python 3 (https://www.python.org) using Matplotlib (https://matplotlib.org), SciPy (https://scipy.org), NumPy (https://numpy.org), AstroPy (https://www.astropy.org) and PyTorch (https://pytorch.org). We also kindly acknowledge libraries provided by Hugging Face (https://huggingface.co).

6. REFERENCES

[1] Joshua Batson and Loic Royer. Noise2Self: Blind Denoising by Self-Supervision. *arXiv e-prints*, page arXiv:1901.11365, January 2019.

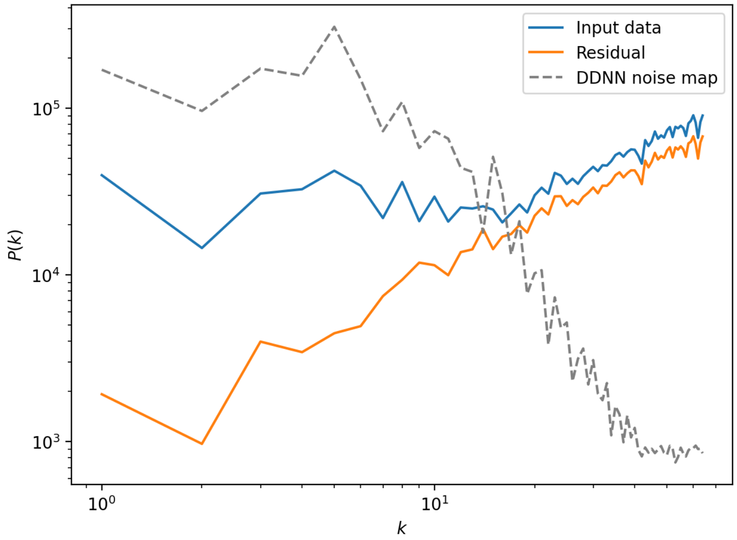

Fig. 4: Power spectrum as function of wave number (k) of the bottom row plots in Figure 3. A low k corresponds to large-scale features and a high k to smaller (high frequency) features. We can clearly see that our DDNN map successfully removes large-scale structure as well as reducing some of the overall high-frequency noise. The dotted line shows the power spectrum of the DDNN map($\bar{p}_\theta(x_0)$) which confirms the predominance of smooth, low-frequency features.

[2] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion Models in Vision: A Survey. *arXiv e-prints*, page arXiv:2209.04747, September 2022.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1406.2661, June 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv e-prints*, page arXiv:1512.03385, December 2015.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *arXiv e-prints*, page arXiv:2006.11239, June 2020.

[6] Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *arXiv e-prints*, page arXiv:1906.02691, June 2019.

[7] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *arXiv e-prints*, page arXiv:1908.09257, August 2019.

[8] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv e-prints*, page arXiv:1608.03983, August 2016.

[9] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. *arXiv e-prints*, page arXiv:2201.09865, January 2022.

[10] Federica Massimi, Pasquale Ferrara, and Francesco Benedetto. Deep learning methods for space situational awareness in mega-constellations satellite-based internet of things networks. *Sensors*, 23(1), 2023.

[11] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv e-prints*, page arXiv:1804.03999, April 2018.

[12] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *arXiv e-prints*, page arXiv:2102.12092, February 2021.

[13] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv e-prints*, page arXiv:2112.10752, December 2021.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, page arXiv:1505.04597, May 2015.

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. *arXiv e-prints*, page arXiv:2010.02502, October 2020.

[16] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion Models: A Comprehensive Survey of Methods and Applications. *arXiv e-prints*, page arXiv:2209.00796, September 2022.

[17] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative Image Inpainting with Contextual Attention. *arXiv e-prints*, page arXiv:1801.07892, January 2018.

[18] Gaetano Zarcone, Lorenzo Mariani, Mascia Bucciarelli, Shariar Hadji Hosseim, Federico Curianò, Paola Celesti, and Luigi Di Palo. Image processing for geo detection. *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 00:1–6, 2021.