

Leveraging Unresolved Hyperspectral Signatures for Robust Deep Learning Classification of Geosynchronous Satellites

Jason T. Kirkendall¹, Bartosz Krawczyk¹, Francis K. Chun², and Michael G. Gartley¹

¹Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, USA

²Department of Physics, United States Air Force Academy, Colorado Springs, CO, USA

PUBLIC RELEASE

Approved for public release: distribution unlimited. (USAFA-DF-2025-855)

DISCLAIMER

The views expressed in this article are those of the author and do not necessarily reflect the official policy or position of the United States Air Force Academy, the Air Force, the Department of Defense, or the U.S. Government.

ABSTRACT

Space Situational Awareness (SSA) has become increasingly urgent as the necessity for dependable tracking and characterization of Resident Space Objects (RSOs) expands. However, the limited availability of community-accessible hyperspectral data has hindered progress. In response, the United States Air Force Academy (USAFA) recently utilized its Falcon Telescope Network (FTN) to gather unresolved hyperspectral signatures of geosynchronous satellites. This initial dataset comprises 594 hyperspectral data samples of 28 satellites, collected over 15 nights from 2022-2025. Most satellites were observed five to ten times a night, although the collections were not uniform, resulting in an unbalanced dataset. The acquired images underwent further pre-processing to extract the exo-atmospheric first-order spectral data for each collection and were stored as an array of “time-series” data. Previously, this data had been investigated for its potential to employ machine learning for unresolved RSO (URSO) classification. Those studies have used the averages of the data samples per collection day to minimize noise per class and have worked exclusively with the time series data when training their algorithms. Keeping the characterization of URSOs in mind, this study builds on previous work. It introduces two novel concepts regarding this dataset: the effectiveness of using all data samples individually to create deep learning (DL) models capable of positively classifying URSOs. Creating a model robust to these variations is essential due to the differences observed between each collection for various reasons, including phase angle, atmospheric changes, etc. The second concept introduced was transforming the time series data into two-dimensional images using Gramian Angular Summation Field (GASF) encoding. Previous research indicates that convolutional neural networks (CNNs) typically learn more effectively using image data rather than time series data. With this in mind, we anticipated our GASF-fed models would yield greater accuracy throughout the confusion matrix compared to the time-series data. We aimed to achieve better results than previous studies; however, we were concerned about the scarcity and noisy data when utilizing every sample. To address data scarcity, we introduced augmented data, which comprised random Gaussian noise and scaling applied to the original GASF and time-series data samples. The models included three convolutional layers and three fully connected layers, with dropout and batch normalization between each layer, executed using k-fold cross-validation with five folds. Utilizing the GASF images, this architecture enabled our model to classify 83% of data from 119 held-out images from the data samples accurately. In comparison, the time-series model achieved an accuracy of only 74% when tested against the held-out data samples. Subsequently, we developed unique conditional variational autoencoders (CVAEs) for the time-series and GASF data, allowing our model to incorporate generative and transfer learning methods to explore the latent space of our data better and provide us with a more diverse set of samples for training. Incorporating the synthetic data generated by the CVAE rendered our model more robust than training with augmented data alone. The resultant model, a straightforward CNN, was trained and validated using a blend of 2000 generated samples from the CVAE and half of the original data, tested on 297 unseen

original images with an accuracy of 84%, surpassing other models from the previously mentioned research. This study demonstrates the effectiveness of utilizing GASF encoding for this type of data with simple yet robust DL techniques to enhance the accuracy and reliability of URSO identification. We also observed optimal results when incorporating all data samples, which is promising for the future of this research in supporting SSA's long-term objectives.

1. INTRODUCTION

The rapid increase in the number of Residential Space Objects (RSOs), which are objects that orbit the Earth and are either artificial or manufactured, necessitates more robust and effective methods for tracking and identifying them. Space Situational Awareness (SSA) is concerned with the safety of our space environment through the characterization and tracking of RSOs. SSA is mainly conducted from Earth-based systems using radar and optical telescopes. Optical systems characterize objects between MEO and GEO 10cm and larger. In comparison, radar can provide precise measurements and location of objects as small as 1cm, but is limited in range to a little over 2,000km. While the satellites in orbit follow strict plans and their orbits are otherwise mapped out, space debris and natural phenomena are under no such constraints. This can lead to on-orbit satellites being damaged or destroyed, and every collision can potentially increase the number of debris. Space debris also has the potential to cause injury and loss of life for our astronauts. The seriousness of this cannot be overstated, if nothing is done to mitigate current debris and limit future amounts, humanity will not have safe routes off Earth for future astronauts or satellites. Knowing the makeup of space debris allows us to find out what pieces contain combustible and or otherwise hazardous material, allowing us to mitigate the debris safely. Identifying satellites is also important, mainly to make sure they are staying in their slated orbits [1][2]. Thus, we need an effective and robust method for identifying RSOs through classification and material identification. This is a multi-faceted problem. We do not have open access information on the materials making up known space vehicles, and we still don't fully understand the effects of space weathering on individual materials in orbit. We do not have much ground truth data regarding images of known RSOs, much less unknown ones. The images we get are difficult to interpret due to the low intensity of light being reflected to us and the atmosphere distortion affecting the signal we receive. Multiple groups are researching to understand and overcome these challenges and others.

This research focuses on data scarcity challenges in satellite classification using hyperspectral data from unresolved images and investigating a meaningful transformation of it for deep learning (DL). The United States Air Force Academy (USAFA) collected and shared data in this study using telescopes from their Falcon Telescope Network (FTN). Low-resolution hyperspectral data was collected using a 100 lines per millimeter diffraction grating and low-cost CCDs with different plate scales. Over the course of five nights, 594 images were collected of twenty satellites. These images include the zeroth and first-order flux for each satellite. This data was pre-processed and distilled to numpy arrays of the first-order spectra.

Three recent studies have used this dataset, each building more tangible and robust models to classify the data. Those studies worked with the data only in its graphical form. The first showed the efficacy of using reduced-dimensional and synthetic data for three classes of satellites, netting an accuracy of 99.49% [3]. The second and third studies increased the classes to align with the data collected. Based on the class label as the satellite's name, they averaged the spectra collected per day, effectively filtering inherent noise in the data. This left them with eighty-six total data files to work with, which decreased the number per sample to between four and five. In the second study. The second study encountered some challenges and, using only a KNN and SVM, achieved 49% accuracy across the dataset [4]. The third study incorporated a conditional variational autoencoder (CVAE), training it with augmented data consisting of random masking and Gaussian noise and using the samples generated from it with very good results on a deep hybrid 1D Inception-ResNet model to classify the data. Pre-training the model on their generated data and fine-tuning it with half of the original data allowed them to get an accuracy of 83.6% on the remaining forty-three samples [6].

Building upon this work, our research focuses on improving the ability of generative models to produce training data that supports generalization across all samples within the dataset. The spectral signatures we observe depend on multiple uncontrolled factors, including the satellite's orientation, atmospheric conditions, and phase angle during observation to name a few. These variations can cause changes in the measured spectral shapes,

which, if not addressed, reduce a model’s ability to learn invariant and class-distinguishing features. To meet the needs of space situational awareness (SSA), it is essential that our generative models capture the underlying class structure while introducing enough variability to prevent overfitting to the test data. We present three new approaches regarding the data and models in this study. The first approach is to utilize all the data samples, as previously mentioned. The second involves transforming these samples from their original configuration of 1D time-series data into 2D images using the Gramian Angular Summation Field (GASF) technique. This preserves the spectral trends in the data while working as a PCA of sorts, and allowing convolutional layers in our models to understand more about the structure of the data. Finally, we trained various CVAEs using similar methods to the most recent study and then train a simple CNN using generated and original data.

2. METHODOLOGY

2.1 Data Preparation

As previously mentioned, the data was given to us in graphical (time-series: TS) format. We worked on data with fast atmospheric compensation applied to mitigate the negative effects of the atmosphere. The y-axis represents unitless digital counts (DNs), corresponding to the flux values as a function of the x-axis wavelengths in nanometers (nm) ranging from 380-880nm. It is worth noting that we also have access to the solar analog version of this data, thereby acting as a relative reflectance measurement for the RSOs. We plan to use these in the future of our research. When working with this data in this 1D form, we normalized per sample between zero and one. It should be noted that all normalization was done using the standard min-max normalization in Python. A study by Wang and Oates discusses how using GAF images from time-series data can increase the accuracy of classifiers in general by allowing them to recognize the structure of the data more consistent with how they were designed to work. They used tiled CNNs due to the size of their images and their needs, but we found that we didn’t need tiled CNNs for our purposes. They also used global normalization without reshaping their data to consider the overall structure [6]. Another group later found that using GAF images in Autoencoders produced more accurate results than when using the data in its original time-series format, as the GAF methods encode both global patterns while retaining local ones from the 1D data, making it easier for DL models to learn from [7]. We attempted to normalize the data in a few approaches; their approach, per sample, and reshape before transforming. We found their approach was more suited for our data. The steps to transform this data into GAF images using global normalization are as follows: Given our original 594 samples, each of size $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{i501}]$ where $i = 1, 2, \dots, 594$, we aim to transform each sample into a Gramian Angular Summation Field (GASF) image using global normalization and then resize it to a new shape of (1, 18, 18).

2.1.1 Step 1: Global Normalization of the Time Series

First, we normalize the entire dataset globally to the range [0, 1]:

$$\tilde{x}_{ij} = \frac{x_{ij} - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})}, \quad \text{for } \begin{matrix} i = 1, 2, \dots, 594, \\ j = 1, 2, \dots, 501 \end{matrix} \quad (1)$$

where \mathbf{X} represents the entire dataset.

2.1.2 Step 2: Convert to Polar Coordinates

Next, we convert the globally normalized time series to polar coordinates:

$$\phi_{ij} = \arccos(2\tilde{x}_{ij} - 1), \quad \text{for } \begin{matrix} i = 1, 2, \dots, 594, \\ j = 1, 2, \dots, 501 \end{matrix} \quad (2)$$

2.1.3 Step 3: Compute the Gramian Angular Summation Field (GASF)

The GASF for each sample is computed as follows:

$$\mathbf{G}_i = \cos(\phi_{ij} + \phi_{ik}), \quad \text{for } j, k = 1, 2, \dots, 501 \quad (3)$$

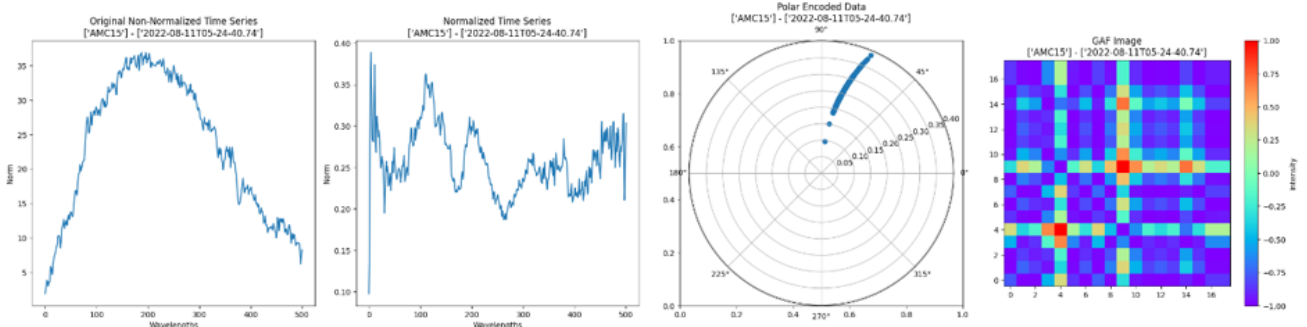


Figure 1: Shows the original TS data as it goes through the GASF transform as defined above

2.1.4 Step 4: Resize the GASF Image

Finally, we resize each GASF image from (501, 501) to (18, 18) using bilinear interpolation. The bilinear interpolation formula for resizing is given by:

$$I_{\text{resized}}(x, y) = \sum_{m=0}^1 \sum_{n=0}^1 I(x_m, y_n)(1 - |x - x_m|)(1 - |y - y_n|) \quad (4)$$

where I is the original image, I_{resized} is the resized image, (x, y) are the coordinates in the resized image, and (x_m, y_n) are the coordinates of the four nearest pixels in the original image.

The resulting GASF images have the shape (1, 18, 18) and are normalized between [-1,1]. Fig. 1, above, shows this process for one sample. The final part of preparing the data was to encode the labels for use in our DL models; all original and generated data using the exact label mapping, as seen in table 1 below.

Table 1: Label Mapping

Index	Label Name	Index	Label Name
0	amc15	10	echostar17
1	anikflr	11	galaxy16
2	anikf2	12	inmarsat4f3
3	anikg1	13	mexsat3
4	dtv10	14	nimiq2
5	dtv12	15	ses11
6	dtv14	16	ses3
7	dtv15	17	skyterra1
8	echostar10	18	spaceway3
9	echostar11	19	wildblue1

2.2 Data Augmentation

The lack of data samples between labels meant that our models had limited diversity to use during training. The samples numbered roughly thirty per label, with one having twenty-five and another twenty-nine. To overcome this limitation and help our model generalize more effectively and be robust when working with data collected in various scenarios, we augmented the data samples in our initial use of CNNs and used those augmented data samples to train the CVAEs. Initially, we trained a CNN using the 1D data after applying principal components analysis (PCA) to find the most important features. We performed Principal Component Analysis (PCA) on the dataset and examined the eigenvalues from the covariance matrix. Eighteen principal components were collected

because they capture a significant portion of the variance in the dataset, thereby retaining the most important features. This allowed us to reduce the dimensionality of the data, as the second group working with this data did previously [4]. We did not have to do this for the GAF images due to the compression naturally reducing noise, while keeping in mind the tradeoff could include some important finer details. Random Gaussian noise and random intensity scaling from a uniform distribution were added to the data being trained to mimic the variability in the data. We did not do this for the GAF images, as we wanted to compare their baseline to the 1D data with augmentations. Since the later CNNs were trained using the synthetic data generated by the CVAEs, we did not use data augmentation for either dataset for those models.

When creating the training data for the CVAEs we used the first two augmentations and an additional one. Randomly masking points down to the minimum value for that sample. This was done instead of setting it to zero to maintain the data distribution without introducing bias. Types of augmentation were chosen to approximate real-world variations and add as much physical grounding as we could, including sensor noise, variations in instrumentation, changes in flux that could be caused by varying atmospheric scenarios and/or phase angles/orientations of the RSOs. We were worried that there might be too much variation to add and obtain quality results, but it proved very helpful in generating robust data, as we will discuss. Starting with the original data file \mathbf{o} , we express the equations that transform \mathbf{o} into $\hat{\mathbf{o}}$.

2.2.1 Add Noise

Gaussian noise is added to the original data:

$$\mathbf{o}_1 = \mathbf{o} + \mathbf{n} \quad (5)$$

where $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise with mean 0 and variance σ^2 appropriate for the data.

2.2.2 Scaling

The data is scaled by a random factor:

$$\mathbf{o}_2 = \mathbf{o}_1 \cdot s \quad (6)$$

where $s \sim \mathcal{U}(a, b)$ is a uniform distribution between a and b .

2.2.3 Sensor Dropout

Sensor dropout is dependent on the data's minimum points:

$$\mathbf{o}_3 = \mathbf{o}_2 \cdot \mathbf{m} \quad (7)$$

where \mathbf{m} is a binary mask with $c\%$ of the elements being 1 and $d\%$ being the minimum value.

2.2.4 Combined Transformation

The combined transformation can be expressed as:

$$\hat{\mathbf{o}} = \mathbf{o}_3 = (\mathbf{o} + \mathbf{n}) \cdot s \cdot \mathbf{m} \quad (8)$$

Fig. 2, above, shows an example of these effects on some of the class labels of the 1D and 2D data. This is displayed with some labels overlapping the GAF images, those labels are not implicit to the data for the augmentations.

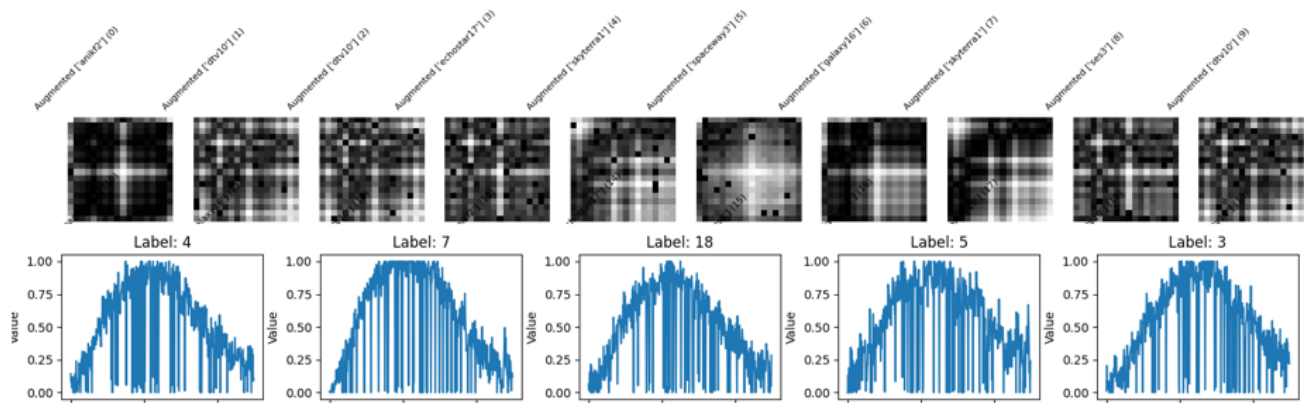


Figure 2: Top: GAF augmented samples Bottom: TS augmented samples

2.3 Deep Learning Models

We created four simple DL models and one with moderate complexity. The main question we wanted to answer was: Can we accurately identify nuanced data between 20 classes, and will the addition of generated data based on the entire set of 594 data files increase diversity in an approach that enhances the overall robustness and accuracy of our models? Five different models were finally used. These consisted of two CNNs and three CVAEs. The road to these final models was fraught with challenges, but this paper will focus only on the final forms. We will begin with the CNNs and move on to discuss the CVAEs. CNNs are generally well-known, so mathematical equations will be omitted for conciseness. The CNNs used the Rectified Linear Unit (ReLU) activation function at each layer besides the last, where SoftMax was used for classification. Each optimization was also the same, using the standard “Adam” optimizer. 3x3 kernels were used in all convolutional layers as well. Their main structure is for 1D data, including: an input layer, three 1D convolutional layers, followed by two fully connected layers, with the output layer positioned last. Within each of these was batch normalization and dropout, and the data was flattened between the convolutional layers and fully connected layers.

The 2D data was similar, but we used blocks to make it easier to keep track of. Four blocks were in total; the first three consisted of a 2D convolutional layer, batch normalization, and max pooling. The last block contained one fully connected layer, dropout, and the output layer. The data was flattened between the convolutional and fully connected blocks. There were attempts to create deeper architectures, but due to the size and complexity of the data, these were found to be appropriately deep. We could have gone down many different avenues when creating classification models for our data, but upon creating these models and seeing the results, it was decided these would be an excellent starting point for our specific question above. Now that the CNNs have been described, we can move on to the more challenging models, the CVAEs.

CVAEs are only different from VAEs in their use of the condition “c” incorporated in the encoder and decoder. This condition is the class label so that the CVAE correctly trains per class distribution instead of mixing all the data distributions. The last group who worked with this data described the VAE and CVAE in detail, which will be omitted in this paper, but please use their paper as a reference point if needed [5]. When deciding what model type to use to generate training data for our classifiers, CVAEs and deep convolutional generative adversarial networks (DCGANs) were both in the running. The CVAE was chosen over DCGAN for a few reasons. They are generally easier to train and less prone to mode collapse. DCGANs are fighting against themselves during training as the generator constantly tries to fool the discriminator, which can lead to successful training without understanding the complete distribution of the data it is training with. The nature of CVAE loss function optimization makes it much more suited for our needs. One type of generator, a WGAN was found near the end of this research as the team worked towards creating a robust CVAE for the graphical data. WGAN is unique from what we could find in the world of GANs as it is immune to mode collapse. The Wasserstein distance between two probability distributions P_r (real data distribution) and P_g (generated data distribution) is defined as:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \quad (9)$$

where:

- $\Pi(P_r, P_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are P_r and P_g , respectively.
- $\|x - y\|$ is the cost function, typically the Euclidean distance between x and y .

An efficient approximation of the Wasserstein Earth-Mover distance equation, replaces the discriminator with what is known as the “critic” in this GAN, allowing for a meaningful loss metric and improved stability during training as the distance between the two distributions is minimized [8]. Since this was thought of late in this research, it wasn’t implemented, but the author will keep it in mind for the future.

2.3.1 First two CVAEs architecture

The loss functions used stray a bit from the classical loss function. In the GAF image case, we add a new term, “perceptual loss”. This perceptual loss is measured as the mean square error (MSE) between the input and output feature maps collected using the VGG16 model at the end of the CVAE pipeline. This was added after attempting reconstruction without it, and results improved considerably with its use. VGG16 is a deeper model trained for classification on ImageNet, a database of 1,000 classes and 14 million images. This training allows it to extract features from various images. It can assist in many applications beyond what it has specifically been trained for, as noted by a research group that used VGG16 as a feature extractor to train a classifier to assist in detecting Alzheimer’s [9]. We used a different version of the reconstruction terms for both data types without changing anything practically. Generally, CVAEs try to maximize the log-likelihood while minimizing the Kullback-Leibler divergence (KLD), and since we are adding the conditional requirement, this is done per class label. Instead, we aim to minimize the MSE between class labels while retaining KLD as our regularization term. We ensure the latent features sampled from the prior distribution are meaningful, allowing the model to create realistic data per label in this approach. For our purposes, using MSE allows for easier training and manipulation of the model, and minimizing the MSE is practically the same as maximizing the log-likelihood used in the reconstruction term. To show this assume that the data \mathbf{x} given the latent variable \mathbf{z} follows a Gaussian distribution with mean μ (predicted by the decoder) and variance σ^2 :

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mu\|^2\right) \quad (10)$$

where $p_\theta(\mathbf{x}|\mathbf{z})$ is the likelihood distribution and d is the dimensionality of \mathbf{x} .

The log-likelihood of the data given the latent variable is:

$$\log p_\theta(\mathbf{x}|\mathbf{z}) = \log\left(\frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mu\|^2\right)\right) \quad (11)$$

Simplifying the log-likelihood:

$$\log p_\theta(\mathbf{x}|\mathbf{z}) = -\frac{d}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{x} - \mu\|^2 \quad (12)$$

To maximize the log-likelihood, we can equivalently minimize the negative log-likelihood:

$$-\log p_\theta(\mathbf{x}|\mathbf{z}) = \frac{d}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\|\mathbf{x} - \mu\|^2 \quad (13)$$

Since $\frac{d}{2} \log(2\pi\sigma^2)$ is a constant with respect to the parameters, minimizing the negative log-likelihood is equivalent to minimizing:

$$\frac{1}{2\sigma^2} \|\mathbf{x} - \mu\|^2 \quad (14)$$

The MSE between the prediction μ and the real image \mathbf{x} is:

$$\text{MSE} = \frac{1}{d} \|\mathbf{x} - \mu\|^2 \quad (15)$$

Minimizing $\frac{1}{2\sigma^2} \|\mathbf{x} - \mu\|^2$ is equivalent to minimizing $\|\mathbf{x} - \mu\|^2$, which is the MSE (up to a constant factor). Each activation function was the same except for the output functions. Parametric ReLU (PReLU) was used. It is like Leaky ReLU (LReLU), but with the flexibility of training itself for the slope of negative values using alpha.

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \quad (16)$$

where α is a learnable parameter. This allows us to overcome vanishing gradients and optimizes the model more than LReLU. One consideration when using PReLU is remembering that if you use a regularization term in the model's initialization, it will err towards ReLU. Fortunately, we can use PReLU in CVAEs without worrying about that since the KLD is for the latent space distribution and doesn't affect the parameters as L1 and L2 regularization do [10]. Since the 1D and 2D data are normalized between [0,1] and [-1,1], the output activation functions for the two CVAEs reflect this: the sigmoid and tanh functions, respectively. Skip connections were used in the final 1D version. The latent space of the distributions seemed challenging for the model to interpret correctly. Without adding skip connections, almost always, at least one class would generate very noisy data, possibly indicating latent space collapse. When trying to train a CNN using this model-generated data, it would rarely get above 50% accuracy on the test data. Introducing skip connections helps the model understand the underlying structure of the data more, allowing it to generate higher-quality data [11]. While the CVAE using the VGG feature maps was able to outperform the 1D data as we will discuss in results, the training and validation left us uneasy.

For our last CVAE, we looked into what we could specifically do to address the latent space regularization and create diverse samples without increasing confusion for our 2D classifier. We took advantage of the bus-type and created the Bus-assisted DeepSMOTE CVAE (BaDSCVAE). To ensure there was no leakage between the training/validation data and the test data used in the CNNs we used a basic MSE and nearest neighbor (NN) check for the 1D data. For the 2D data created by the VGGCVAE we applied an NN threshold and checked the Structural Similarity Index Measure (SSIM) between images. We were able to apply more quality/novelty filters for the BaDSCVAE images. We applied the same as in the VGGCVAE and then introduced per-class Mahalanobis filtering, before finally conducting a k-center greedy selection. By adding these two, we ensured the samples would exclude outliers while including a more diverse and less redundant variety of samples per class [12]. The CNNs are then trained on a combination of real data and filtered synthetic data, while the held-out test set remains strictly real.

2.3.2 Bus-assisted DeepSMOTE-CVAE

The DeepSMOTE (DS) method for CVAEs has previously been noted for creating diverse, high-quality samples, with this performance not degrading for small sample sizes of difficult data [13]. Considering this could help us with our dataset, we examined how to implement a version tailored more specifically to our domain. Our design emphasizes the following: keeping a robust latent structure, and utilizing the bus type to assist with sample diversity instead of solely relying on satellite name as the label. The BaDSCVAE we trained, decodes images conditioned on the original class labels as stated, but has additional terms to keep it aware of the bus type for each satellite since many share a bus type and this can lead to more confusion between the original class labels if not accounted for. The generative model extends the original loss function of our model with classification, contrastive, and covariance regularizers:

$$\mathcal{L} = \underbrace{\|x - \hat{x}\|_1}_{\text{reconstruction}} + \beta \underbrace{D_{\text{KL}}[q_\phi(z|x, y_{\text{sat}}) \parallel \mathcal{N}(0, I)]}_{\text{regularization}} + \lambda_{\text{ce}} \underbrace{\mathcal{L}_{\text{CE}}(f_\psi(\mu), y_{\text{sat}})}_{\text{label-smoothed, class-weighted}} + \lambda_{\text{cov}} \underbrace{\mathcal{L}_{\text{DIP-Lite}}(\mu)}_{\text{covariance to } I} + \lambda_{\text{sup}} \underbrace{\mathcal{L}_{\text{SupCon}}(\mu; y_{\text{sat}}, y_{\text{bus}})}_{\text{bus-aware contrast}} \quad (17)$$

Here $\hat{x} = p_\theta(z, y_{\text{sat}})$, and $z \sim q_\phi(z|x, y_{\text{sat}})$. β was annealed during the warm-up. Then label smoothing was applied to improve calibration and generalization, as shown in the paper by Müller, Kornblith, and Hinton [14]. Let $p = \text{softmax}(f_\psi(\mu))$. With smoothing ϵ , $\tilde{y} = (1 - \epsilon)y_{\text{sat}} + \epsilon \mathbf{1}/C$ and weights w_c :

$$\mathcal{L}_{\text{CE}} = - \sum_{c=1}^C w_c \tilde{y}_c \log p_c. \quad (18)$$

The Disentangled Inferred Prior(DIP)-Lite-VAE penalty in $\mathcal{L}_{\text{DIP-Lite}}$ encourages $\text{Cov}(\mu) \approx I$. With $C_\mu = \text{Cov}(\mu)$, we penalize off-diagonal energy (correlations between different latent dimensions), pushing the latent space to stay disentangled and better structured overall as was noted in the study published in 2018 [15]:

$$\mathcal{L}_{\text{DIP-Lite}} = \|C_\mu - \text{diag}(C_\mu)\|_F^2 + \frac{1}{2} \|\text{diag}(C_\mu) - \mathbf{1}\|_F^2. \quad (19)$$

The supervised contrastive term $\mathcal{L}_{\text{SupCon}}$ pulls same-satellite class samples together, while pushing apart those of different classes sharing a bus-type as they share attributes, making them more difficult to discriminate. With a greater weight attributed to these "hard negatives" the latent space becomes more robust to these samples. Let $s_{ij} = \mu_i^\top \mu_j / \tau$. For anchor i , positives $\mathcal{P}(i) = \{j : y_j^{\text{sat}} = y_i^{\text{sat}}, j \neq i\}$, and negatives $\mathcal{N}(i) = \{j : y_j^{\text{sat}} \neq y_i^{\text{sat}}\}$, weight negatives that share the same bus by $(1 + \alpha)$:

$$\mathcal{L}_{\text{SupCon}} = - \frac{1}{|\mathcal{P}(i)|} \sum_i \sum_{p \in \mathcal{P}(i)} \log \frac{\exp(s_{ip})}{\sum_{a \in \mathcal{P}(i) \cup \mathcal{N}(i)} w_{ia} \exp(s_{ia})}, \quad w_{ia} = \begin{cases} 1 + \alpha, & y_a^{\text{bus}} = y_i^{\text{bus}}, a \in \mathcal{N}(i) \\ 1, & \text{otherwise.} \end{cases} \quad (20)$$

After training, all training samples are encoded and we use Borderline-SMOTE on the latent space in order to specifically target decision boundary regions. A small "bus-boundary" nudge is applied in the latent space such that the sample is pushed to the nearest satellite centroid within a bus-type other than its current satellite class. This is done to diversify intra-bus modes before the decoding and filtering steps discussed above.

3. RESULTS

We will discuss the training and outcome of the CVAEs. After discussing this, we will briefly introduce how we trained the CNNs and compare the accuracy of our CNNs trained on various data types.

3.1 CVAEs

All the CVAEs used a 80/20 split for the original data. The training portion was then split 78/22, and the 78% for training had random augmentations done to them, leading to the use of 13,727 augmented samples and 371 original samples. 104 original samples were set aside for validation, and 119 were used for testing. We have set aside this test set of samples to be used for testing the CNNs as well. The 1D CVAE for time-series data was validated with the non-augmented data, leading to better generalization over all class labels. After fifteen epochs of training, the test results showed that our model had a well-structured latent space and learned how to reconstruct the data. Fig. 3 shows examples of the original and generated data per class label. Most generated classes carried extra noise from visual inspection. Upon inspecting the histograms of the flattened latent space, including their mean and standard deviations for the original and generated data, one synthetic label had a standard deviation slightly less than its original counterpart and many were roughly equal, as Fig. 4 shows. This could mean it missed capturing the variability of the underlying structure for that label, or that it has a well-regulated latent space. Upon seeing the CNN results, it would appear to point towards the former of those two theories. Being that these were the best results we had for the 1D version, we moved forward using this to generate data for the CNN. The VGGCVAE was the first CVAE we created for the GAF images. Our

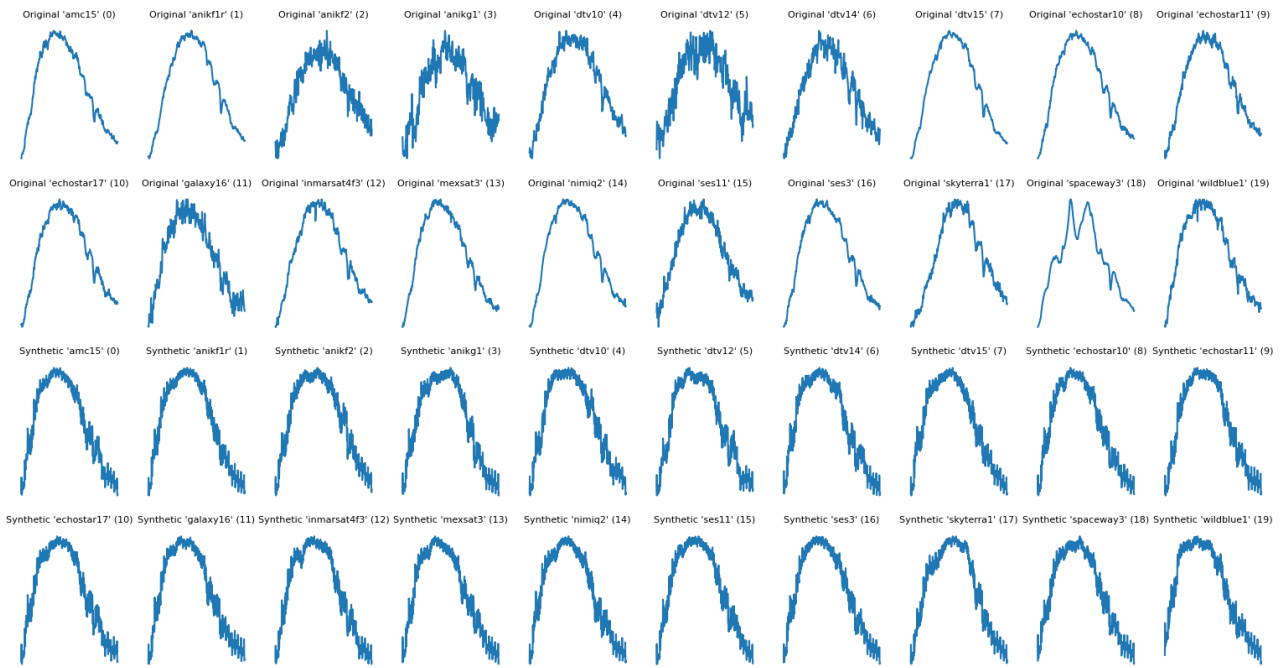


Figure 3: TS Original and Generated samples

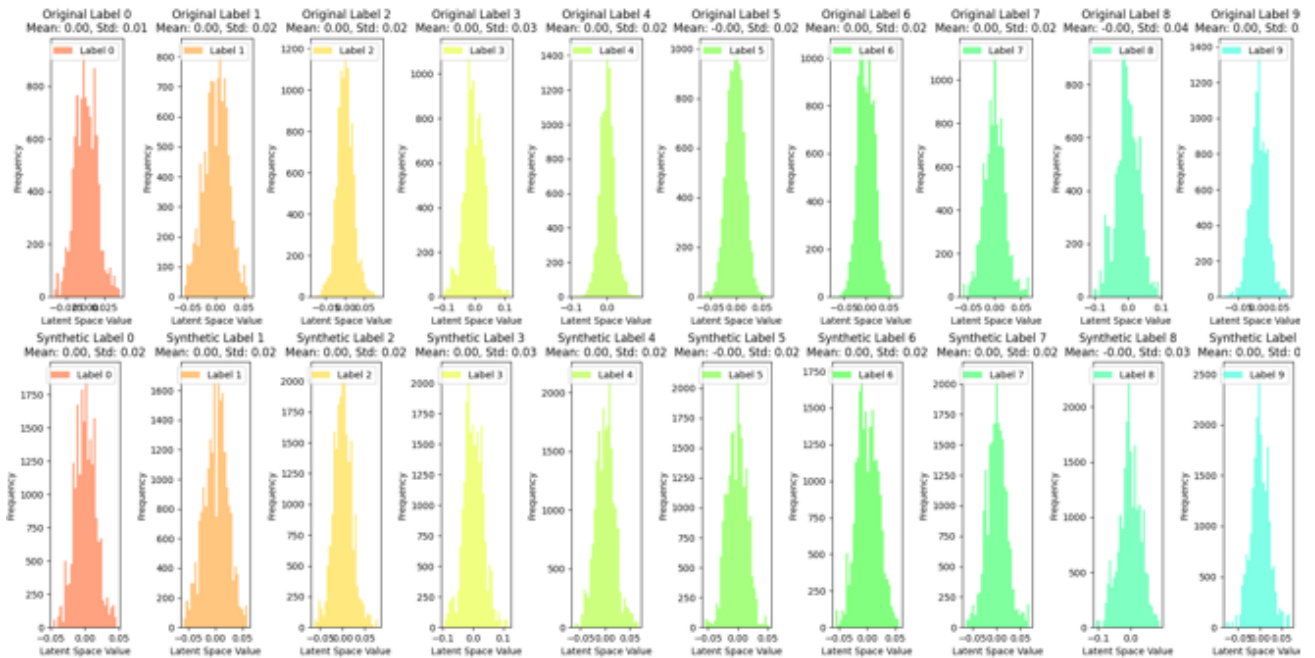


Figure 4: TS Original and Generated samples latent space flattened histograms from labels 0-9

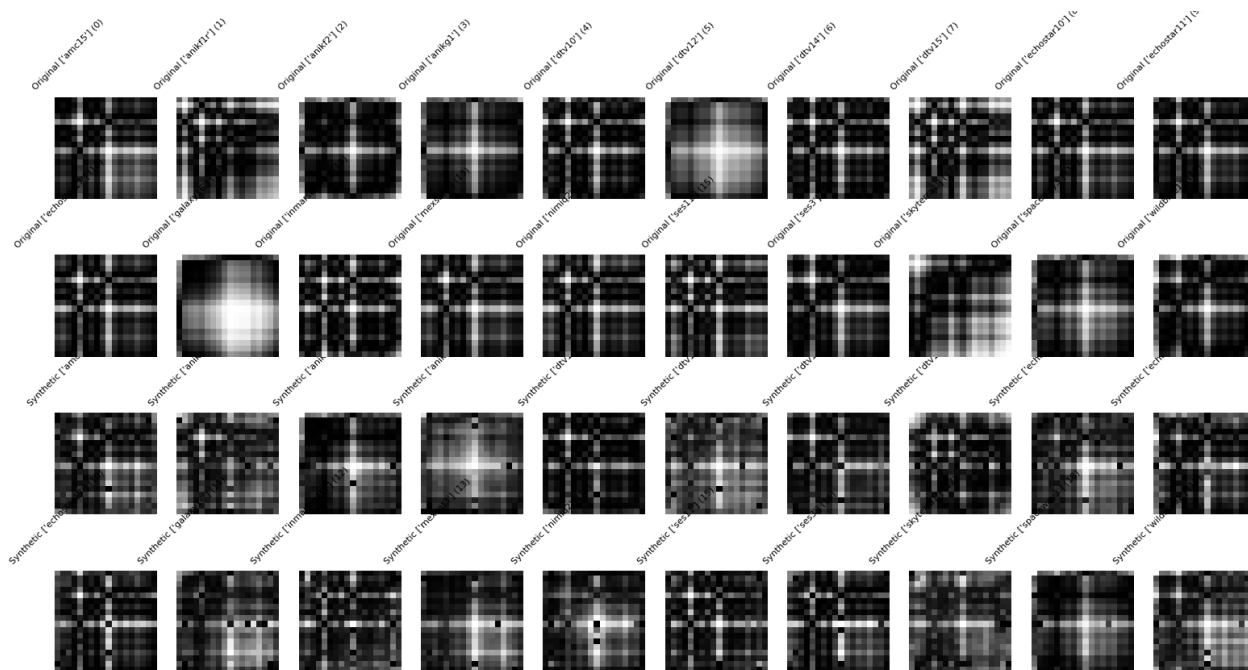


Figure 5: GAF Original and Generated samples of VGGCVAE data. Note: Labels going across images are not part of the data

results from this CVAE indicated that our model may be overfitting to the training data and not learning the latent space distribution as well as it should, which was to be expected given that we included a small KLD weight. This KLD weight reduced the importance of regularization, allowing our model to generate realistic samples while still allowing the latent space to be reasonably structured. If the KLD weight was increased, we observed that the synthetic samples tended to be of extremely poor quality, rendering them useless for the CNN. Upon investigating the generated samples and the flattened latent space histograms, we noted that the generated samples retain more augmentations and still looked visually closer to their original counterparts than the 1D samples. The generated class label histograms had larger standard deviations than the originals while generally sharing the same mean values. This suggests that the model captured a more diverse latent space than the original data structure, or that the model put too much importance on outliers and other samples that lie outside of the original class boundaries. The results from the CNN would have us believe it to be the latter for this case. The results are in Fig. 5 and Fig. 6 below.

As previously discussed, more tricks were introduced in our BaDSCVAE, and the results show that they seemingly paid off. The latent space histograms are similar to those from the 1D CVAE, but as results will show, these are well regularized latents. The resulting synthetic samples also looked visually closer to the originals (The figures of the BaDSCVAE histograms and images are not shown here for conciseness. Please contact the author if you would like to see them).

3.2 Simple CNN Accuracies

CNNs accuracies: The CNNs were trained in the same approach, with the only difference being how the data was fed into them and the number of epochs. We generated 100 samples per label for a total of 2,000 samples each. We combined the original data with the 2,000 generated samples. These were divided into 80/20 splits during the five k-fold training and validation processes. The graphical TS data differed, in that PCA was used on the augmented data and finding that the eigenvalues trended to hold less variance than the original. We chose the same number of principal components for our data as the original. We also increased the amount of epochs from 100 (as it was for the GAF images created using the VGGCVAE) to 150, finding that it took slightly more for the model to converge. Despite both the 1D and VGGCVAE sample fed CNNs overfitting during training/validation,

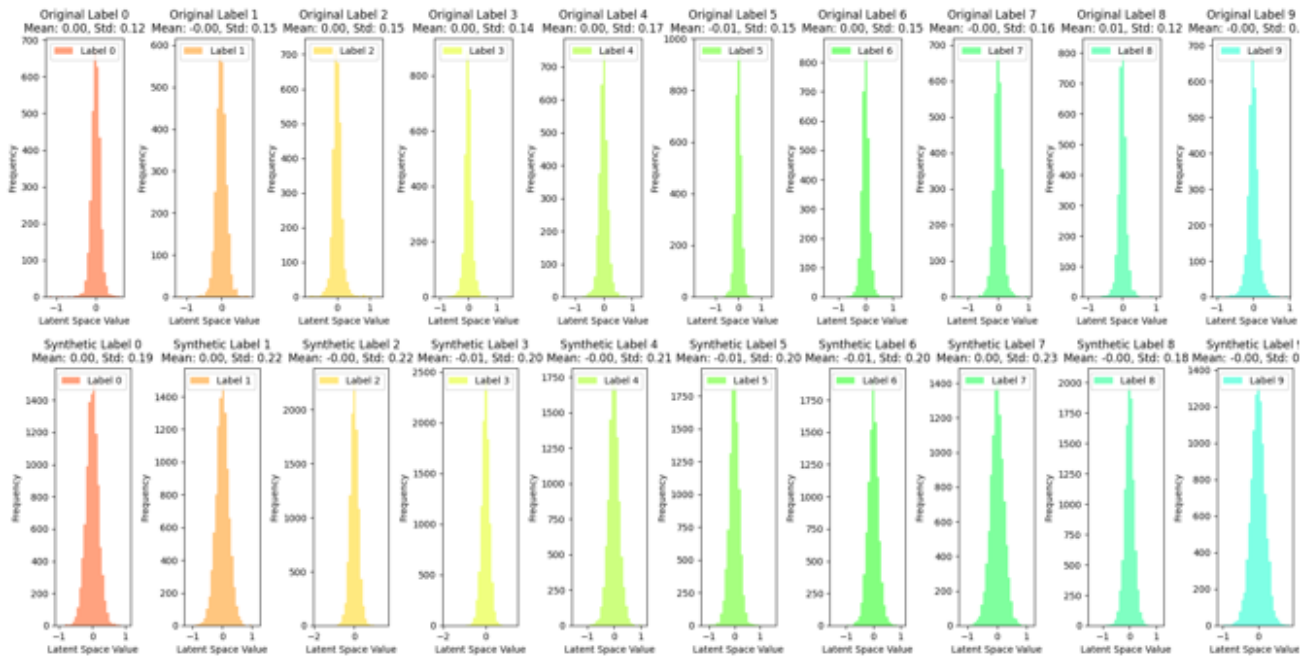


Figure 6: GAF Original and Generated samples latent space for VGGCVAE flattened histograms from labels 0-9

the GAF model consistently classified unseen data with higher F1 accuracies, as seen in the following results table. The BaDSCVAE samples not only outperform on accuracy metrics but also converge faster, at only 50 epochs, and are well fit with stable training. Per_samp references using the data normalized per sample instead of globally for the GASF transforms. While the training graphs for the VGGCVAE CNN show severe overfitting

Table 2: CNN Results on 119 Original Samples

Model	Val Acc (Mean \pm Std) [%]	F1 Score [%]
TS_Original	71.40 \pm 6.10	74.14
TS_CVAE	78.18 \pm 0.50	76.66
GAF_PerSamp	49.46 \pm 2.21	60.73
GAF_Orig	66.67 \pm 0.63	83.25
GAF_VGGCVAE	59.96 \pm 1.50	88.31
GAF_BaDSCVAE	99.33 \pm 0.25	93.48

to the training data as shown in Fig. 7, those for the BaDSCVAE CNN are fit well, even across 10 folds, as seen in Fig. 8. As previously discussed, we ensured there was no data leakage as far as direct copies go amongst the training/validation and test samples. This suggests that, despite the overfitting, the VGG-backed model can still capture some of the important structures of the original data, but not as well as the BaDS-backed model. As a final check, we split the original data into a 50/50 split and combined it with the generated data from the VGG-backed one first. We increased the number of epochs to 150 and the fold count to ten. During this training and testing phase, our model achieved an impressive F1 accuracy of 84% on 297 unseen original data samples. Worried there may have been data leakage allowing for this impressive result, the BaDSCVAE to CNN pipeline was called upon. We split the original data in half and held the test half out to be used with the CNN only. Upon doing this, using the newly created samples and the training half of the original data, we once again did ten k-fold cross-validation with 150 epochs to match the previous test accurately. Unfortunately, only an F1 accuracy of 80% was found on the 297 unseen originals. This is a great score for the dataset itself, but this most

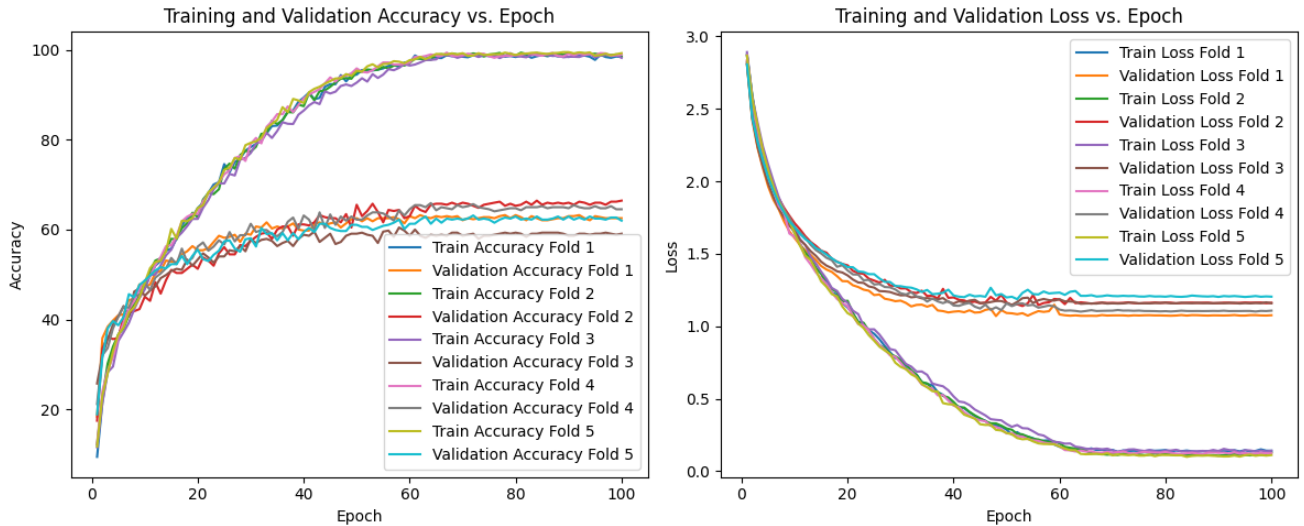


Figure 7: Training and loss graphs of the 5 k-fold VGG-backed GAF CNN with synthetic and original data

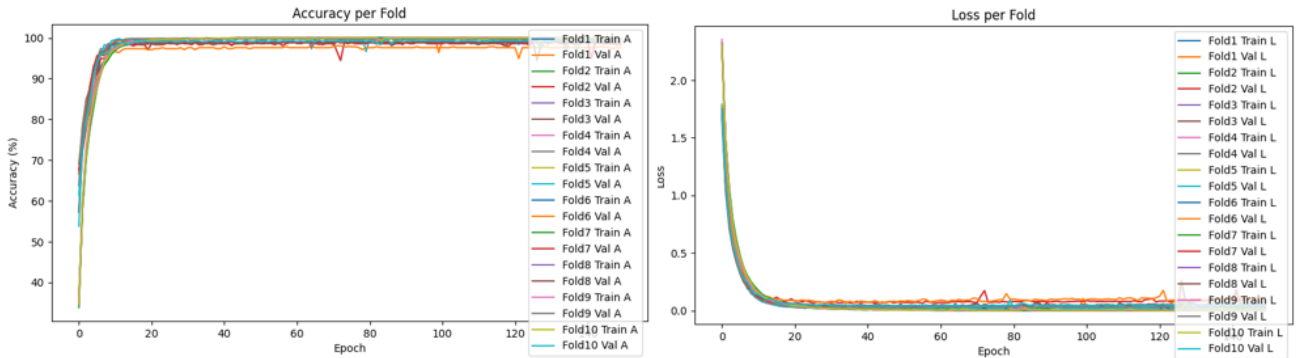


Figure 8: Training and loss graphs of the 10 k-fold BaDS-backed GAF CNN with synthetic and original data

likely means that even though there was no direct data leakage from the VGGCVAE CNN using the 50/50 split above, there were some synthetic samples that were very similar to the originals tested on and so we cannot take its pipeline accuracy seriously. We infer this from the significant improvement in performance of the BaDSCVAE CNN on 119 held-out samples compared to the VGGCVAE CNN, and we will be looking further into this in the future.

We can also surmise that the VGGCVAE pipeline was weaker overall when compared to our BaDSCVAE pipeline. This can be verified by a mixture of the CNNs’ performances, and by discussing the 2D t-SNE graphs. Fig. 9, below, shows that, at first glance, the raw pixel-space data reveals significant overlap between classes. This overlap reinforces what we already know about the data, indicating these pixel-level differences are not discriminative enough. Looking at the latent space representation of the t-SNE graphs allows us to observe the level of regularization our models are bringing to the table. The VGGCVAE was able to create good enough samples to increase performance over the original data, but its latent space was poorly regularized, with quite a lot of inter-class mixing. In contrast, the BaDSCVAE produced a well-structured latent space. This structure can be seen as classes formed tighter, distinguishable clusters. The synthetic samples created by the BaDSCVAE also embedded within their corresponding real distributions.

4. DISCUSSION

This research highlights a key finding: the impact of transforming unresolved GEO satellite spectra into GAF images on deep learning models, particularly their ability to recognize important features and structures within

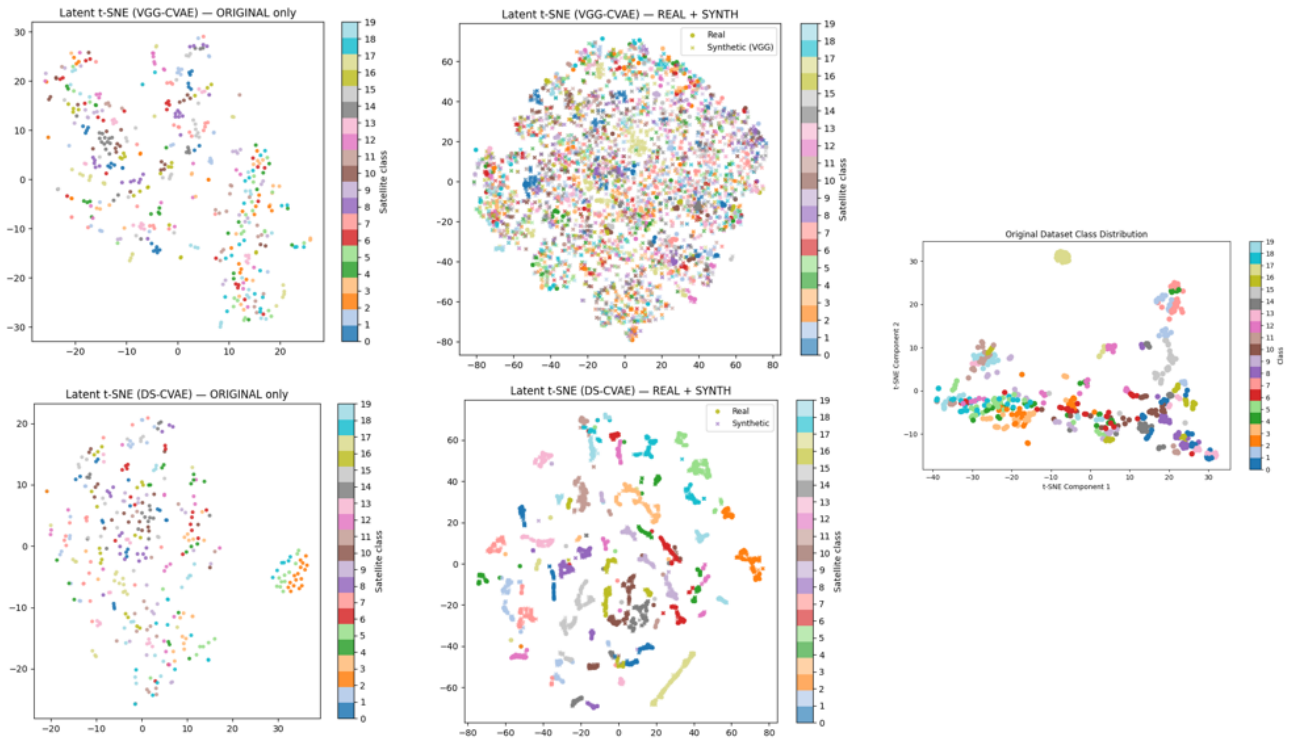


Figure 9: t-SNE graphs from left to right: Original latent dimensions from VGG(top) and BaDS(bottom), Original + Synthetic dimensions from VGG(top) and BaDS(bottom), Original in pixel space. Note: DS == BaDS

the data. Initially, we observed that this transformation provided an immediate boost to classification models when compared to the original samples. When we incorporated our initial CVAEs, we continued to see the GASF images outperform the graphical samples. Starting with the synthetic samples from the VGGCVAE, which was able to reconstruct the images visually, but had a weakly regularized latent space. We wanted to see how samples from a more robust and well-regularized latent space would perform. The BaDSCVAE was developed with more explicit regularization for the latent space through the use of loss functions, penalties, and filters while keeping the bus-type in consideration. This CVAE was able to merge the latent clusters, producing more challenging boundary samples and providing more class-faithful clusters in general. The CNN trained on samples from this dataset was able to perform significantly better, both in terms of accuracy and training stability, than previous pipelines for both 1D and 2D data from various groups. This demonstrates that not only was transforming the data important, but also the choice of a high-quality data generator is crucial when focusing on improving the classification of URSOs, even when using a relatively simple CNN.

4.1 Limitations

Throughout this study, several limitations became apparent. The issues caused by data scarcity have already been mentioned, and previous research discusses how to overcome this, specifically by generating realistic and diverse data and waiting for more to be collected and shared. Even when using a data generator to create realistic and diverse samples, representative of the classes, the method still needs a representative original dataset to base itself on. As robust as our BaDSCVAE is, it cannot provide us with physically meaningful samples based on conditions not seen in the original dataset. Additionally, the trade-off of using the Gramian transformation should be considered. While GAF methods allowed for significant improvements, they still involved compressing the original data of length 501 down to an 18x18 pixel image. Doing so may blur or outright remove finer details and features.

4.2 Future Work

Moving forward, we want to optimize the characterization and identification of URSOs, specifically, the next phase of our research involves the problem of material characterization for GEO satellites. We will continue using GAF representations and increase the resolution to 32x32. This will enable us to capture more details while minimizing noise. Instead of using the non-calibrated, DN flux data, we will be switching to the calibrated solar analog data, which will better reflect the material reflectance, focusing on bus-type groupings as the materials will be easier to separate in that approach. Additionally, we plan to introduce multimodal fusion using various data representations, including complementary ones such as the GADF (Gramian images made using differences instead of summations). The CVAE-Classification pipeline could also be more robust by exploring semi-supervised methods such as those used in research by Cedillo et al. as well as looking into more robust classifiers [16]. These changes will open the approach to explore spectral unmixing and whether/how it can be applied to this data type. Vasile et al. recently investigated spectral unmixing techniques for low Earth orbit (LEO) satellites, including a promising technique to account for space weathering by red-shifting spectra. Before them, Velez-Reyes and Yi investigated using various libraries to assist in spectral unmixing. Still, they noted they didn't use Bi-directional Reflectance Distribution Function (BRDF) measurements, an important aspect to consider. Dianetti and Crassidis mainly discuss using polarized light curves but touch on BRDF measurements and approaches to incorporate them into this type of research. These works show potential in further identification of GEO sats and other RSOs that we hope to expand upon [17][18][19]. The

5. CONCLUSION

This study explored a new method of working with the URSO 1D data from the FTN by transforming it into GAF images. In doing so, it establishes that these samples provide a richer representation for each class than the TS data was able to. As our BaDSCVAE pipeline produced diverse samples from robust latent groupings and allowed even a simple CNN to outperform other models trained on 1D samples from previous research. Two contributions specifically were highlighted in our study: Unlocking more information from a limited data set using GAF methods, and bridging the data-scarcity and variability challenges in URSO classification by using properly regularized domain-specific generators. These contributions and the results of our study show that we can move towards pipelines which move beyond averaging the samples together, and instead utilizing and learning from the diversity of all samples. Setting more of a foundation for future orientation invariant material characterization efforts, aligning with and supporting the goals of SSA.

REFERENCES

1. D. Mehrholz, L. Leushacke, W. Flury, R. Jehn, H. Klinkrad, and M. Landgraf, "Detecting, Tracking and Imaging Space Debris," (pp. 128-134). *rbulletin*, vol. 109, Feb. 2002.
2. F. Massimi, P. Ferrara, and F. Benedetto, "Deep Learning Methods for Space Situational Awareness in Mega-Constellations Satellite-Based Internet of Things Networks," *Sensors*, vol. 23, no. 1, p. 124. 2023, doi:10.3390/s23010124.
3. P. D. Dao, X. C. Yee, D. M. Strong, B. Roth, and F. K. Chun, "Multi-geosynchronous satellite classification with spectroscopic signatures," in *Proc. SPIE 12519, Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imaging XXIX*, 1251907, Jun. 2023, doi: 10.1117/12.2665121.
4. X. C. Yee, P. D. Dao, D. M. Strong, C. J. Wetterer, B. Roth, and F. K. Chun, "Machine learning classification GEOs using spectral data," United States Air Force Academy.
5. C. A. Mello, M. Mendoza, L. Camacho, and D. Eberhardt, "Advancing geosynchronous satellite classification utilizing spectral data via fine-tuned pretrained deep learning models," *AMOS Conference*, 2024.
6. Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," *arXiv:1506.00327v1*, Jun. 2015, doi: arXiv.1506.00327.
7. U. Yokkampon, A. Mowshowitz, S. Chumkamon, and E. Hayashi, "Autoencoder with Gramian Angular Summation Field for Anomaly Detection in Multivariate Time Series Data," *Journal of Advances in Artificial Life Robotics*, vol. 2, no. 4, pp. 206-210, 2021-2022. Released on J-STAGE Oct. 2022, doi: 10.57417/jaalr.2.4_206.

8. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv:1701.07875v3, Dec. 2017, doi: arXiv.1707.07875.
9. S. Sharma, K. Guleria, S. Tiwari, and S. Kumar, "A deep learning based convolutional neural network model with VGG16 feature extractor for the detection of Alzheimer Disease using MRI scans," *Measurement: Sensors*, vol. 24, 2022, Art. no. 100506. doi:10.1016/j.measen.2022.100506.
10. K. He, X. Zhang, and S. Ren, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," arXiv:1502.01852v1, Feb. 2015, doi: 10.48550/arXiv.1502.01852.
11. A. B. Dieng, Y. Kim, A. M. Rush, and D. M. Blei, "Avoiding Latent Variable Collapse with Generative Skip Models," *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, in *Proceedings of Machine Learning Research*, vol. 89, pp. 2397-2405, 2019.
12. O. Sener, S. Savarese, "Active Learning for Convolutional Neural Networks: A Core-Set Approach," *ICLR 2018*.
13. D. Dablain, B. Krawczyk and N. V. Chawla, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6390-6404, Sept. 2023, doi: 10.1109/TNNLS.2021.3136503.
14. R. Müller, S. Kornblith, G. Hinton, "When Does Label Smoothing Help?" *NeurIPS 2019*.
15. A. Kumar, P. Sattigeri, A. Balakrishnan, "Variational Inference of Disentangled Latent Concepts from Unlabeled Observations," *ICLR 2018*.
16. L. R. Cedillo, K. Acosta, M. Velez-Reyes, and D. DeBlasio, "Using neural networks to classify hyperspectral signatures of unresolved resident space objects," *University of Texas at El Paso and Carnegie Mellon University*, Jan. 2024.
17. M. Vasile, L. Walker, A. Campbell, S. Marto, P. Murray, S. Marshall, and V. Savitski, "Space object identification and classification from hyperspectral material analysis," arXiv:2308.07481v1, Aug. 2023, doi: arXiv.2308.07481.
18. M. Velez-Reyes and J. Yi, "Hyperspectral unmixing for remote sensing of unresolved objects," *Sensor and Signal Analytics Laboratory, Department of Electrical and Computer Engineering, The University of Texas at El Paso*, 2020.
19. A. D. Dianetti and J. L. Crassidis, "Resident Space Object Characterization Using Polarized Light Curves," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 2, pp. 246-263, 2023, doi: 10.2514/1.G006847.