Advancing Geosynchronous Satellite Classification Utilizing Spectral Data via Fine-Tuned Pretrained Deep Learning Models

Chad A. Mello, Matthew Mendoza, Leonardo Camacho, Damien Eberhardt
United States Air Force Academy

ABSTRACT

A recent study utilizing data collected from the Falcon Telescope Network for space situational awareness has demonstrated the potential of non-resolved optical signatures for identifying geosynchronous orbit satellites using traditional approaches like principal component analysis and support vector machines. While these methods can be effective, especially for relatively small datasets, they face limitations such as sensitivity to data variance and dependence on manually selected features, which can hinder scalability and accuracy. To better cope with these challenges, we propose leveraging modern machine learning techniques, specifically fine-tuned convolutional neural networks pretrained over large spectral datasets. Our approach may offer significant advantages, including enhanced feature extraction and robustness to spectral variances.

In our experiments, we developed and utilized a hybrid Inception-ResNet deep learning model. We trained one instance against a 600,000-plus star spectra dataset from the Sloan Digital Sky Survey and another instance on satellite spectra generated by a custom conditional variational autoencoder. We then fine-tuned these instances using augmented satellite data to better capture the unique characteristics of optical-based satellite spectra, aiming to improve classification accuracy and efficiency. By automatically learning features directly from the data, our model is able to identify complex patterns in satellite spectra that traditional methods might miss.

Additionally, the use of generative models and transfer learning allows us to leverage prelearned features from large datasets, reducing the requirement for extensive labeled satellite data and addressing the challenges associated with small sample sizes typical in space situational awareness research. The results demonstrate that our proposed method achieves over 83% accuracy across 20 GEO classes using only 86 *complete samples*, showing the potential of our models as more data is collected via the Falcon Telescope Network. This study underscores the potential of neural network architectures in enhancing the accuracy and efficiency of GEO classification, contributing to the greater goal of improved situational awareness in space.

1. INTRODUCTION

Due to the growth in the number of low-Earth orbit (LEO) and geosynchronous orbit (GEO) satellites, as well as the accumulation of space debris orbiting Earth, the need for continual and timely survey data of these objects is reaching a critical juncture. Resolved imagery can be an excellent tool for satellite identification and tracking purposes, yet it is often expensive and technically challenging [10] [3]. Capturing resolved imagery of satellites from the ground, particularly those in high orbits like GEOs, requires high-resolution telescopes and sensors with precision guiding and tracking. Ground-based telescopes must overcome atmospheric distortion and often require adaptive optics or space-based platforms to obtain clear images. Even then, the images may not be detailed enough to provide definitive identification, especially for smaller or more distant objects. There is an available system of ground-based telescopes that may offer a more practical approach, the Falcon Telescope Network (FTN).

The FTN was developed over a decade ago by the Department of Physics at the United States Air Force Academy (USAFA). Conceived as a global network of relatively small aperture optical telescopes, the FTN was designed to support research into space situational awareness and to provide USAFA cadets and faculty with the capability to conduct research in satellite characterization and astronomy [1]. Currently, the network includes 12 observatories spread across various regions, including the United States, Chile, Germany, and Australia, with the potential for additional sites in the future [2]. It is worth mentioning here that the FTN has evolved into a legitimate global network; however, it currently lacks a comprehensive and centralized repository of regularly collected observational data. This

situation is likely to soon change, as observatories within the network are being upgraded with new optics, sensors, and modernized automation technology [7]. These enhancements are expected to result in more consistent and regular data collection from FTN instruments

While these scopes are not equipped to reliably guide-capture resolved imagery of objects orbiting our planet, they are well-suited for *slitless spectroscopy*, a form of low-resolution imagery useful for observing and characterizing objects in space, particularly Earth-orbiting satellites [5], [6], [7]. Refer to Fig.1 for a visual explanation. These images might appear as fuzzy, out-of-focus balls or streaks of light to the human eye, but our interest lies not in their shape or size, but in the unique light signatures these images hold about each object. Slitless spectrographs produced by using low-cost charged coupling devices (CCDs) offer a cost-effective and efficient method for capturing light from unresolved objects across a wide field of view, enabling the capture of multiple objects simultaneously within a large area of the sky.

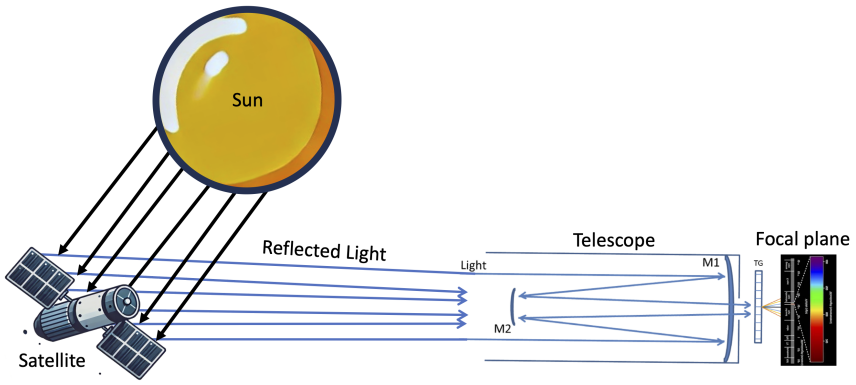


Fig. 1: An overly simplistic illustration depicting how the data used in our experiments was collected. Light from the sun reflects off the surfaces of the satellite. Ground-based telescopes capture this reflected light and direct it through sensors used for the diffraction grating process. The light is then spread over a sensor's surface where it is measured at different wavelengths, covering the visual spectrum from approximately 3800Å to 8800Å.

Recently, a small number of GEO samples were collected using two FTN instruments. These samples have been the subject of two recent studies that utilized Principal Component Analysis (PCA) and Support Vector Machines (SVMs). The first study [3] showed promising results via dimensionality reduction and data synthesis, achieving up to 99.49% accuracy across three satellite classes with a limited dataset. However another study performed by the same authors [10], which included a broader number of satellite classes and experiments using PCA, SVMs, and K-nearest neighbors (KNN) classification, observed diminished results with up to 49% accuracy across 20 satellite classes.

Using the exploratory work performed in aforementioned papers as a precedent, our research focuses on applying advanced deep learning models that may better generalize across GEO spectral data. In preparation for the expected increase in data from the FTN, we aim to develop machine learning (ML) models capable of effectively processing and making sense of this data. By demonstrating the capability of these models with the currently available data, we also highlight the potential for greater performance over larger datasets that may contribute valuable insights to space situational awareness efforts.

1.1 Research Objective

As space situational awareness becomes increasingly important for both military and private space activities, the ability to develop robust machine learning models that can effectively generalize to spectral data related to objects in Earth's orbit is of keen interest to us. With the Falcon Telescope Network (FTN) expected to provide a significant influx of data in the near future, we feel it is important to develop models that work against this data now.

Currently, we face the challenge of data scarcity, which limits our ability to conduct effective research and develop these models. Overcoming this scarcity is our first goal. Additionally, working with data from slitless spectrographs presents unique challenges: (1) The potential overlap of spectra from different objects in the same field makes it diffi-

cult to distinguish between nearby objects (a topic not explored in this paper), and (2) objects in space can change their orientation, altering their reflective light signatures and causing inconsistencies when recording their measurements.

This paper presents an approach designed to address these specific challenges, demonstrating potential for good performance and robustness with satellite spectral data. Our aim is to have effective working models ready in anticipation of the increased data that will eventually be collected by the FTN, particularly in the context of slitless spectroscopy for satellites in geosynchronous orbit (GEO).

2. THE DATA

The satellite spectra used in our experiments was collected via a limited experiment using slitless spectroscopy with diffraction gratings to capture the spectral signatures of satellites using 2 optical telescope systems from the FTN system: a 16-inch Ritchey-Chrétien at USAFA and a 20-inch Ritchey-Chrétien located at Northeastern Junior College in Sterling, Colorado. Of note, the USAFA instrument has since been replaced with a new 20-inch Ritchey-Chrétien as part of the ongoing FTN upgrade efforts. These telescopes were retrofitted with instruments that provided nearly identical performance, with the same 100-lines-per-mm transmission grating and the same charge-coupled device (CCD) sensors [10].

For a total of 5 evenings spectral data was collected for 20 different satellites. Each satellite was observed between 4 and 10 times per evening. All observations for a particular evening were stored in a single file per satellite. The unprocessed data was stored in MATLAB format and contained spectral flux measurements ranging between 3800Å and 8800Å. For our experiments we were provided with a total 86 files; they covered observations across all 20 satellites.

2.1 Data Preparation

Even though each file contained multiple observations, we treated each file as a single sample by producing a single sample that contained the average across all the observations in the file; this yielded a total of 86 satellite samples to work with. Averaging across the samples effectively acts as a noise filter, smoothing out and stabilizing the signals. We suspect that this technique may also help to cope with the variations in the light signatures due to angular variance between satellite measurements (solar panel angles) and the location of telescope sensors on the earth. In a future paper we may experiment with training over each observation independently to see if the models better learn distributions of spectral features.

We applied several preprocessing steps to the spectra after averaging across observations. First, we resolved the signal to a specific target resolution of 400Å from 20Å, as similarly performed in [3] to reduce the effects between data collected using sensors with varying resolutions across the wavelength spectrum:

$$F_{\text{deg}}(w_i) = \sum_{j=-N}^{N} F(w_i - w_j) \cdot G(w_j; \sigma)\, \Delta w \tag{1}$$

where:

- $F(w_i)$ is the original flux at discrete wavelength w_i,
- $F_{\text{deg}}(w_i)$ is the degraded flux at the discrete wavelength w_i,
- $G(w_j;\sigma)$ is the Gaussian function defined as $G(w_j;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{w_j^2}{2\sigma^2}\right)$
- $\sigma = \frac{\text{target\_resolution}}{2.355}$, relating the full width at half maximum (FWHM) to the standard deviation of the Gaussian distribution,
- w_j represents the discrete deviation from the central wavelength w_i,
- Δw is the wavelength step size in the discretized space,

- N is the number of points on either side of the central wavelength to include in the summation, depending on how far out the Gaussian kernel is considered.

After which we resampled the signals to expand across 400 points from the original 367 points using a 3rd order polynomial. This better aligned the data for fine tuning with using a model pretrained on another spectral dataset, as explained in Section 3.4.

$$F_{\text{res}}(w) = S(w) \tag{2}$$

where:

- w ranges from new_wavelength_min to new_wavelength_max,
- $S(w)$ is the cubic spline interpolation function, which interpolates the flux values F_{deg} from the degraded spectrum,
- $F_{\text{res}}(w)$ is the resampled flux at wavelength w,
- The new wavelength grid is defined as:

$$w_i = \text{new\_wavelength\_min} + i \cdot \frac{\text{new\_wavelength\_max} - \text{new\_wavelength\_min}}{\text{num\_points} - 1} \quad \text{for } i = 0, 1, 2, \ldots, \text{num\_points} - 1$$

We then normalized the flux values across the spectrum to values between 0 and 1 using min-max normalization:

$$F_{\text{norm}}(i) = \frac{F(i) - F_{\text{min}}}{F_{\text{max}} - F_{\text{min}}} \tag{3}$$

where:

- $F(i)$ is the original flux value at index i,
- $F_{\text{norm}}(i)$ is the normalized flux value at index i,
- F_{min} and F_{max} are the minimum and maximum flux values in the spectrum, respectively.

Finally, we mapped integer values to all the sample names, assigning them each an integer class label. Table 1 details the class mapping.

Index	**Name**	**Index**	**Name**
0	amc15	10	echostar17
1	anikf1r	11	galaxy16
2	anikf2	12	inmarsat4f3
3	anikg1	13	mexsat3
4	dtv10	14	nimiq2
5	dtv12	15	ses3
6	dtv14	16	ses11
7	dtv15	17	skyterra1
8	echostar10	18	spaceway3
9	echostar11	19	wildblue1

Table 1: Mapping of integer labels to satellite names

Refer to Fig.2 to see the final data product.

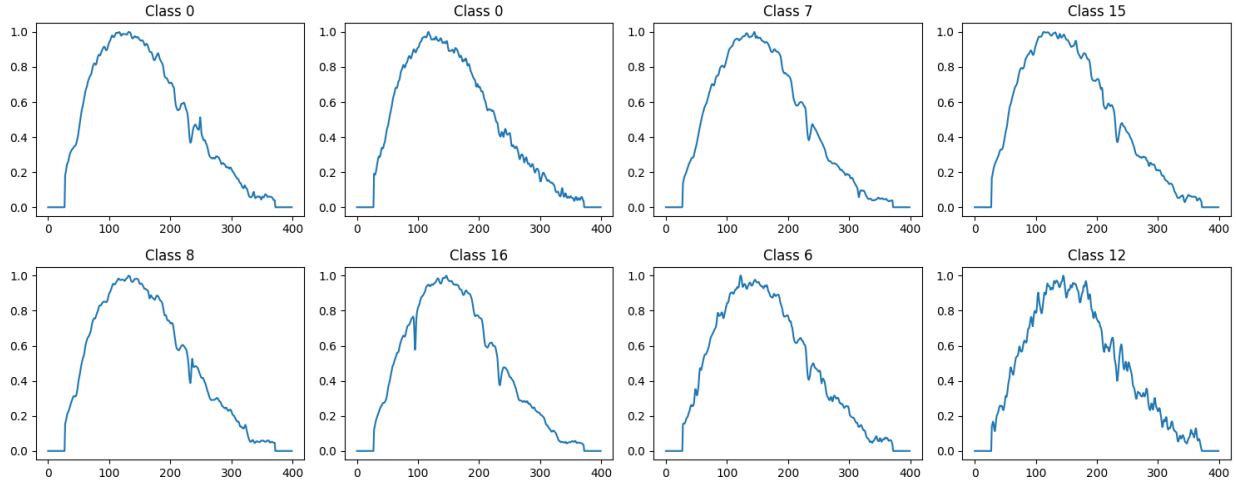


Fig. 2: Several randomly selected labeled satellite spectral samples after preprocessing: degrading to 1/4 wavelength, polynomial resampling, and min-max normalization of flux values.

3. METHODOLOGY

Since we had very few original data samples to work with, 86 total samples, we were left with just 4 representations per class on average throughout the entire training set - not a lot; typically, deep learning models require many thousands or millions of data samples to train against. We did not wish to resort to traditional learning methods or few-shot learning methods for this paper, as we wanted to experiment with deep learning models we believed might generalize better over nonstationary and noisy data. Therefore, we performed several experiments in an attempt to overcome challenges related to data scarcity and potential overfitting to present evidence of a model with potential for good performance over the data.

We developed and utilized a hybrid Inception-ResNet (IRN) model and then trained one instance against a 600,000-plus star spectra dataset from the Sloan Digital Sky Survey. We trained another instance of the same model on satellite spectra generated by a conditional variational autoencoder (CVAE) to learn embeddings that contain the important features that differentiate the various satellites. See Section 3.2 for more details on the model and Section 3.3 for details on how it was trained. We chose a CVAE because, once trained, these models are easy to use to generate data of a specific class. We then fine-tuned both IRN instances using augmented FTN satellite data to better capture the unique characteristics of optical-based satellite spectra, aiming to improve classification accuracy and efficiency.

3.1 A Hybrid 1D Inception-ResNet Model

IRNs tend to be robust for noisy data [9]. The spectral dataset we were attempting to learn over contains subtle differences between the various classes, is noisy, and has a nonstationary nature to it. While typically used in improved computer vision models, we decided to create our own hybrid IRN model to train over FTN data. Our model combines the strengths of Inception modules, which capture features at multiple scales, with Residual connections, which help stabilize and improve the training of deep networks. The Inception-like blocks in our model allow it to capture features at multiple scales simultaneously. Each branch (1x1, 3x3, 5x5 convolutions, and max pooling) captures different levels of detail in the data; this is particularly useful when working with complex signals like spectral data. A simpler CNN typically uses only one kernel size per layer and may overlook some details that our multi-scale approach captures. Our model effectively increases the representation power of each layer by combining outputs from multiple convolutional layers. This allows the network to learn more complex patterns in the data, which can lead to better performance and scalability. This could be very useful should data with higher resolution become available in the near future. Below, we express our approach to building the model step-by-step by combining simple mathematical expressions to build a complete Inception-ResNet block. The completed model is then constructed from *stacking* these blocks together.

The first equation below represents the operation of a basic **Inception module**:

$$\mathbf{y}_{\text{Inception}} = \text{concat}\left(\text{Conv1D}_{1\times1}(\mathbf{x}), \text{Conv1D}_{3\times3}(\mathbf{x}), \text{Conv1D}_{5\times5}(\mathbf{x}), \text{Conv1D}_{1\times1}(\text{MaxPool1D}(\mathbf{x}))\right) \tag{4}$$

$\mathbf{x}$ denotes the input data, which in our case is a sequence of 1D data points related to the flux values. The functions $\text{Conv1D}_{k\times k}(\cdot)$ represent convolution operations with different filter sizes k, allowing the model to extract features of varying sizes from the input data. The function $\text{MaxPool1D}(\cdot)$ performs a pooling operation that reduces the spatial dimensions of the input while retaining the most significant information. The function $\text{concat}(\cdot)$ concatenates the outputs of these parallel operations, combining the multi-scale features into a single tensor.

$$\mathbf{y}_{\text{Residual}} = \mathbf{x} + \mathbf{y}_{\text{Inception}} \tag{5}$$

The result of the Inception module, $\mathbf{y}_{\text{Inception}}$, is then combined with the original input $\mathbf{x}$ through an **addition operation**, forming the **residual connection**: This connection helps preserve the original information from the input while allowing the network to learn additional features. This additive merge ensures that even as the network gets deeper, the original data characteristics are not lost, making the model easier to train.

$$\mathbf{y} = \sigma(\mathbf{y}_{\text{Residual}}) \tag{6}$$

After the residual connection, as is the case in all deep models, an **activation function** $\sigma(\cdot)$, typically a Rectified Linear Unit (ReLU), is applied to introduce non-linearity. The activation function modifies the combined output, $\mathbf{y}_{\text{Residual}}$, to produce the final output of the block, $\mathbf{y}$.

$$\mathbf{y} = \sigma\left(\mathbf{x} + \text{Conv1D}_{1\times1}\left(\mathbf{y}_{\text{Inception}}\right)\right) \tag{7}$$

The equation above combines the entire **Inception-ResNet block** into a single expression: It starts with an input $\mathbf{x}$ and passes it through an Inception module. The output of the Inception module is then added back to the original input $\mathbf{x}$ via a residual connection. The combined result is passed through a ReLU activation function to introduce non-linearity, yielding the final output of the block.

$$\mathbf{y}_n = \sigma\left(\mathbf{y}_{n-1} + \text{Block}_n(\mathbf{y}_{n-1})\right) \tag{8}$$

In a complete IRN network, multiple such blocks are stacked together: Here, $\mathbf{y}_{n-1}$ is the output of the $(n-1)$-th block and serves as the input to the n-th block. Each block applies the IRN operations described above, with the final output $\mathbf{y}_n$ being the result after the last block. Our model in total consisted of 5 IRN blocks with each convolutional layer receiving 32 filters. Our IRN graph can be seen in Appendix A.

3.2 Conditional Variational Autoencoder

We built a CVAE specifically to use as a form of data augmentation; we used it to generate realistic data samples across all class representations with which to train our ML models against. A CVAE extends from the basic concepts held in the Variation Autoencoder (VAE)[8]. It is well known that the VAE aims to learn a generative model by encoding the input data x into a lower-dimensional latent space z and then decoding it back to reconstruct x. The VAE is optimized by minimizing a combination of some sort of reconstruction loss with KL divergence loss.

$$\mathscr{L}_{\text{VAE}}(x;\theta,\phi) = \underbrace{-\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{Reconstruction Loss}} + \underbrace{\text{KL}(q_\phi(z|x)\|p(z))}_{\text{KL Divergence Loss}} \tag{9}$$

where:

- $q_\phi(z|x)$ is the encoder that approximates the posterior distribution of the latent variable z given the input x.
- $p_\theta(x|z)$ is the decoder that reconstructs x from z.
- $p(z)$ is the prior distribution of the latent variable, typically assumed to be a standard Gaussian $\mathscr{N}(0,I)$.

Now, a CVAE simply extends the VAE by incorporating an additional condition c into both the encoder and decoder. This condition represents relevant information, allowing the model to generate data conditioned on c. For our purposes c represents a conditional over the class labels; we want to learn some distribution over some $p(z|c)$.

$$\mathscr{L}_{\text{CVAE}}(x,c;\theta,\phi) = \underbrace{-\mathbb{E}_{q_\phi(z|x,c)}[\log p_\theta(x|z,c)]}_{\text{Conditional Reconstruction Loss}} + \underbrace{\text{KL}(q_\phi(z|x,c)\|p(z|c))}_{\text{Conditional KL Divergence Loss}} \tag{10}$$

where:

- $q_\phi(z|x,c)$ is the encoder that approximates the posterior distribution of the latent variable z given both the input x and the condition c.
- $p_\theta(x|z,c)$ is the decoder that reconstructs x from both z and c.
- $p(z|c)$ is the prior distribution of the latent variable conditioned on c.

As is evident here, the key difference between the VAE and CVAE lies in the conditioning on c, which forces the model to learn to generate data consistent with the condition, thereby allowing for more controlled and structured data generation. We note here that it is also possible to use this CVAE to study the latent space of the spectral data more closely to gain insights into the underlying structure and relationships within it[4]. This future study could aid in understanding the variability and distributions of different earth-orbiting classes.

3.3 Data Augmentation Methods

Data augmentation was employed to overcome the data scarcity issue with the FTN data and enhance the robustness of our machine learning models by artificially increasing the diversity of the spectral data. The first approach to augmentation was through the construction and training of our CVAE described in section 3.2. The CVAE was trained using the data augmentation technique describe in the next paragraph. It was trained for 35 epochs on over 5,000 augmentations per sample taken from the original dataset of 86 samples, totaling 430,000 augmented samples. Note that the CVAE was never exposed to any original samples; it was not trained or validated against any original samples. This was done so that the CVAE would gradually converge on important features from embedding through noisy data, essentially regularizing the model as it trained. As it turns out, the model training was fairly robust, and the CVAE discovered good embeddings that allowed it to generate realistic data with marginal variance across all of the classes in the original data. Fig 3 shows what some of these samples look like.

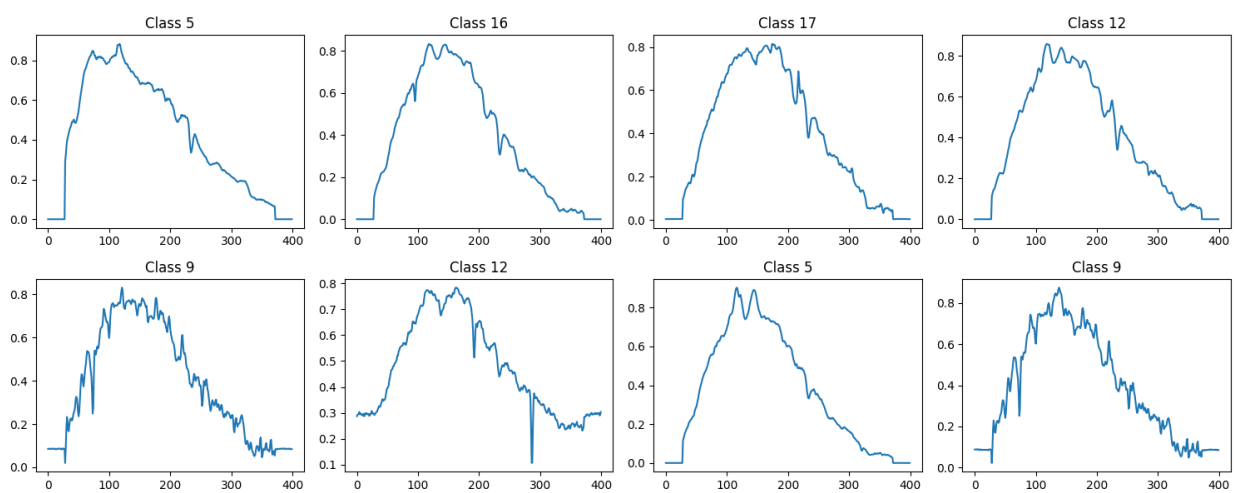


Fig. 3: Several randomly selected labeled satellite spectral samples generated from our CVAE model.

Our second data replication and augmentation approach combined two techniques with data replication: noise injection and random masking. Firstly, Gaussian noise was added to each spectral sample to simulate the effect of varying noise conditions that might be encountered in practical scenarios. The noise was characterized by a mean of zero and a

standard deviation governed by a predetermined noise level. Secondly, a specified percentage of the spectral data points in each sample was randomly masked, setting these points to the minimum value observed in the spectrum, thus simulating the effect of dropout due to sensor errors or occlusions. As it did for training our CVAE, this technique acted as a good regularization technique when training our models. Please refer to Fig. 4 for a visual illustration of this data.

$$\tilde{s} = s + n + m \tag{11}$$

where n represents the noise vector added to the spectrum. The noise vector is generated from a Gaussian distribution:

$$n \sim \mathcal{N}(0, \sigma^2) \tag{12}$$

with σ set to the predefined noise level. The masking operation m is defined as:

$$m_i = \begin{cases} 0 & \text{if } i \notin M \\ \min(s) - s_i & \text{if } i \in M \end{cases} \tag{13}$$

Here, M is the set of indices chosen randomly from the spectrum, representing $p\%$ of the total number of data points in s, where p is the mask percentage. This operation effectively simulates dropout by setting the selected spectral values to the minimum observed value in the spectrum.

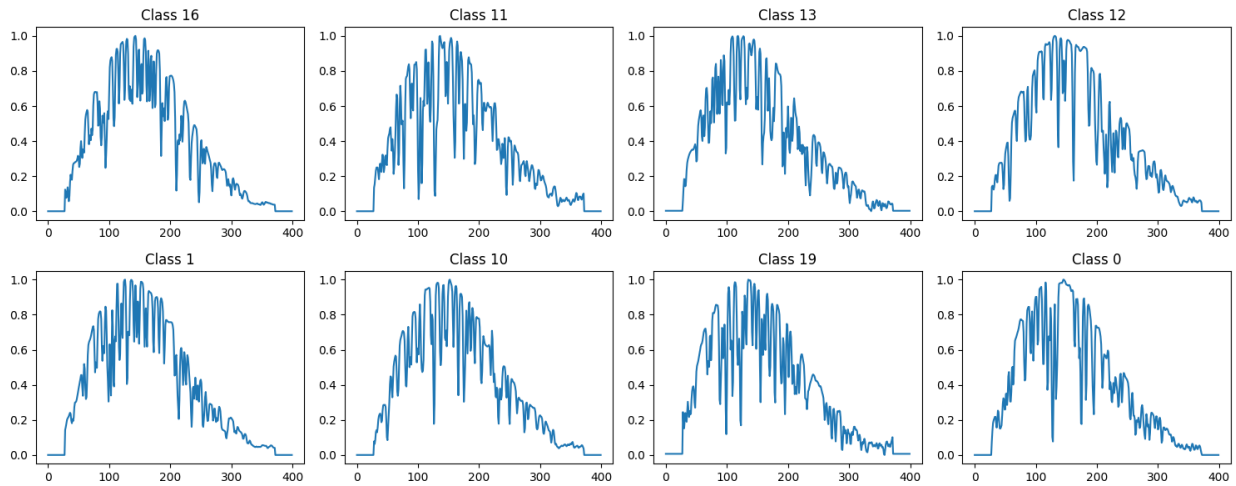


Fig. 4: Several randomly selected labeled augmented samples produced by data replication and augmentation of the original FTN samples as described.

3.4 Training Process

We began our experiments using two pretrained IRN instances: (1) an IRN pretrained using a 600,000-plus star spectra dataset we pulled together from a larger astronomical spectra survey curated through the Sloan Digital Sky Survey (SDSS) [11], and (2) an IRN pretrained on completely generated data from the CVAE. These models were never exposed to any of the original FTN samples.

3.4.1 Pretrained SDSS IRN Instance

The SDSS instance was trained using only the broader 7-class categories for stars; we did not train using any subclass labels. See Table 2 for class descriptions and label mapping. The SDSS is divided into multiple phases and includes several distinct surveys and has several datasets that include stellar spectra and photometric data, curated using different instruments over multiple phases of the project. Each phase had its own goals and instruments, contributing to different aspects of astronomical data collection. Unlike the FTN slitless method used to collect the satellite (see1), the SDSS used fiber-fed slit spectrographs, where light from different objects (stars, galaxies, quasars, etc.) is directed into fibers and fed into a spectrograph for spectral analysis.

The dataset from the SDSS we created covers a visual range between 3600Å and 10400Å with 5,400 data points per sample. The equipment in the SDSS employed various sensors, ranging from older, lower-resolution sensors to newer, higher-resolution ones. Our dataset includes samples from the Baryon Oscillation Spectroscopic Survey (BOSS) and the extended Baryon Oscillation Spectroscopic Survey (eBOSS) from the Sloan Digital Sky Survey III (SDSS-III) and Sloan Digital Sky Survey IV (SDSS-IV), the original Sloan Digital Sky Survey (SDSS) and Sloan Digital Sky Survey II (SDSS-II), and the Sloan Extension for Galactic Understanding and Exploration 1 and 2 (SEGUE-1 and SEGUE-2, part of SDSS-II and SDSS-III) data from the Data Release 18 (DR18) dataset.

The spectral resolution varies across the wavelength range, as does the range itself. In the original SDSS spectrograph, the resolution was $R \sim 1800$ to 2000, while in the BOSS spectrograph, it is $R \sim 1500$ to 2600, where R is the resolving power ($\lambda/\Delta\lambda$). To standardize the data and make it more suitable for use as a pretrained model with the goal of fine-tuning over FTN data, we degraded all of our data samples to match the lower resolution, closer to that of the original SDSS resolution. We resampled to 400 points using a 3rd order polynomial as demonstrated in Equation 2 and then normalized the flux values via min-max normalization referenced in Equation 3. The final data product is visualized in Figure 5.

For training we used a 70/20/10 data split, allocating 70% of the data for training, 20 for validation, and 10% for testing, to ensure a balanced approach to model training and evaluation. We extended training out to 30 epochs, where overfitting was evident after 16 epochs. We assessed and utilized the model trained at 16 epochs; this model was used in our experiments with fine-tuning (see Section 4). The model showed a validation accuracy of 91.50% and testing accuracy of 81.25% We suspect that the model was fitting too closely to the validation samples. We could have improved this initial model by adjusting hyperparameters and ensuring a more consistent sampling between test and validation data; however, our goal was not to optimize a model to SDSS data. We deemed this model's performance adequate for experimentation purposes.

Class	Map	Description
O	0	Hot, blue stars with temperatures above 30,000 K, often found in regions of star formation.
B	1	Blue-white stars with temperatures between 10,000 and 30,000 K, often very luminous.
A	2	White or bluish-white stars with temperatures between 7,500 and 10,000 K, including many of the visible stars.
F	3	Yellow-white stars with temperatures between 6,000 and 7,500 K, including many main-sequence stars.
G	4	Yellow stars with temperatures between 5,200 and 6,000 K, similar to our Sun.
K	5	Orange stars with temperatures between 3,700 and 5,200 K, often older stars.
M	6	Red stars with temperatures below 3,700 K, typically the most common type of stars in the galaxy.

Table 2: Spectral Classes, Integer Mappings, and Brief Descriptions of SDSS dataset.

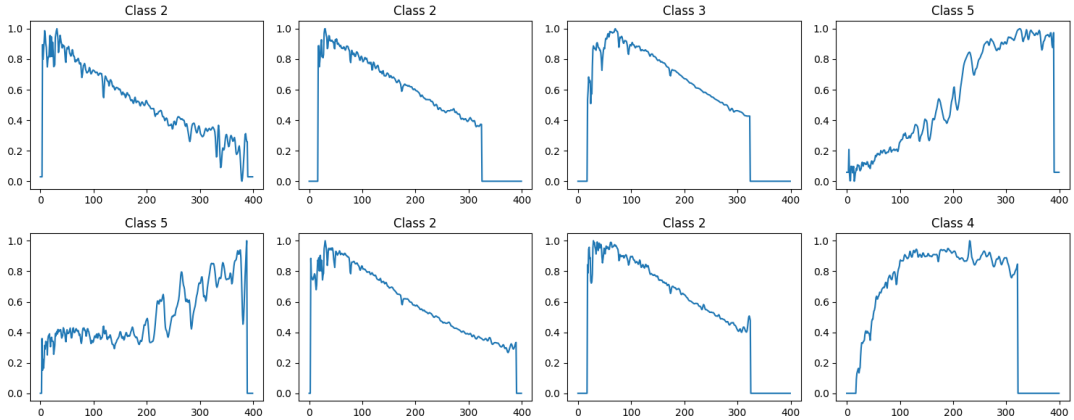
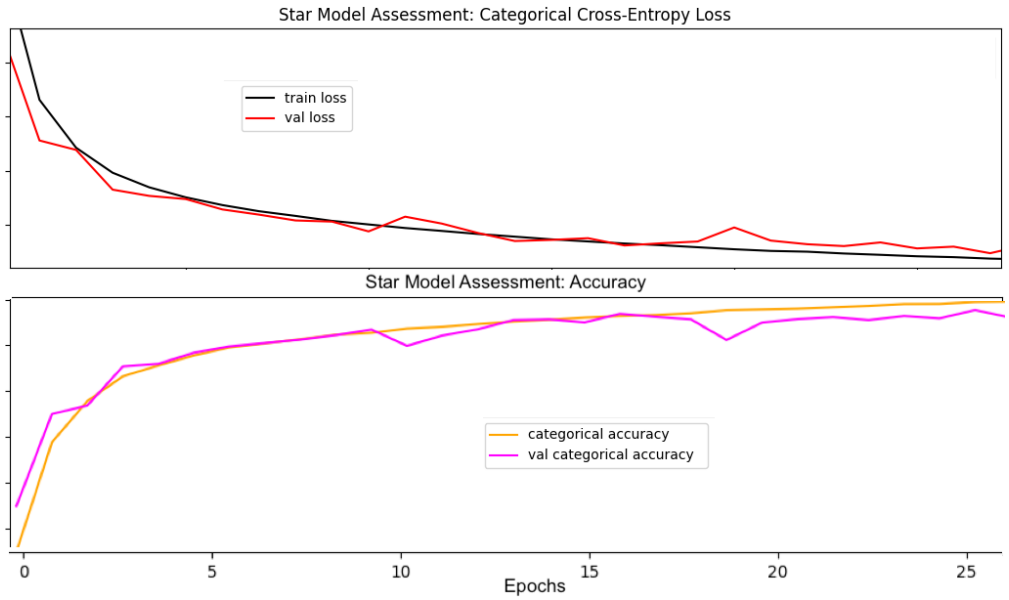
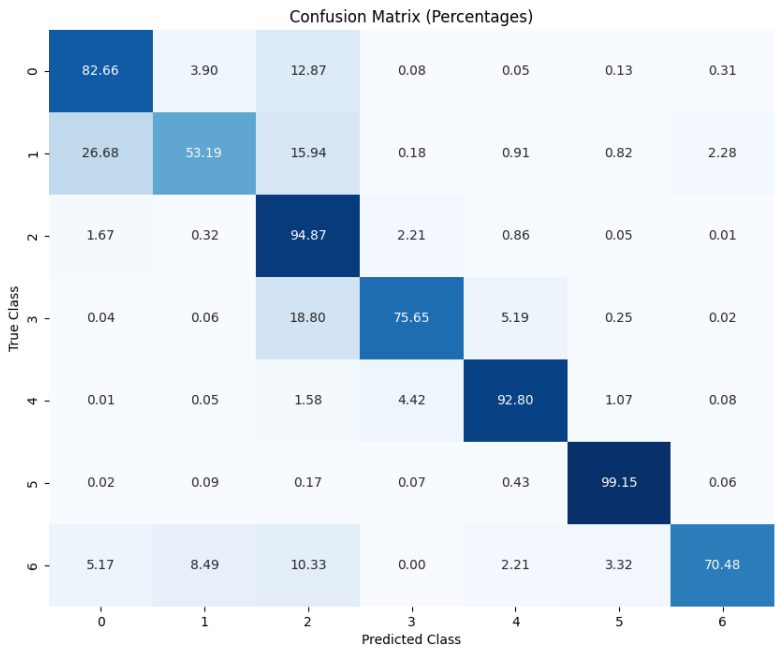


Fig. 5: Several randomly selected preprocessed labeled SDSS samples used to pretrain our first IRN model instance.



(a) Learning curves while training base model over SDSS. Notice tight fitting against validation data until after epoch 16. This could have been remedied by tuning hyperparameters and/or training out for more epochs.



(b) Confusion matrix for base model pretrained over SDSS for 16 epochs. The confusion matrix is normalized by row and shows accuracy percentages over each class. 82.25% overall test accuracy.

Fig. 6: Training results and confusion matrix for the base model trained over SDSS.

3.4.2 Pretrained CVAE-Generated Inception-ResNet Instance

This model was trained over data generated by our CVAE model. It was never exposed to the original FTN samples. The model was trained for 15 epochs before overfitting became obvious. The trained model showed a 99.9% validation accuracy score, but only reaching 75.6% accuracy. This result was not a bad result considering the model was never exposed to actual data during training. It is likely that it had overfit to the generated data.

4. EXPERIMENTATION

With our 2 pretrained models in hand we experimented with fine-tuning for each model hoping to create improved models that can learn well over the FTN data. Fine-tuning is a popular form of domain adaptation, especially in deep learning, where a model trained on a large dataset in the source domain is adapted to perform well on a related but distinct target domain. We adapted each pretrained model from its original source domain $\mathscr{D}_S$ with a distribution $P(X_S)$, to a target domain $\mathscr{D}_T$ with a distribution $P(X_T)$, where generally $P(X_S) \neq P(X_T)$.

Assume a pre-trained model with parameters θ, trained on the source domain $\mathscr{D}_S$. The goal of domain adaptation is to adjust θ to θ' such that the model performs well on the target domain $\mathscr{D}_T$.

The adaptation can be expressed as an optimization problem:

$$\theta' = \arg\min_{\theta} \mathscr{L}_{\mathscr{D}_T}(f(x;\theta), y) \tag{14}$$

where $f(x;\theta)$ represents the model predictions with parameters θ, $\mathscr{L}_{\mathscr{D}_T}$ is the loss function quantifying the error in the target domain, and y are the true labels in $\mathscr{D}_T$.

4.1 Fine-Tuning Procedure

Our fine-tuning approach involved the following steps:

1. Start with the pretrained models, each with parameters θ obtained from pretraining against their respective data source $\mathscr{D}_S$.
2. For the pretrained SDSS instance, add 2 new dense layers and replace the final output layer in the model with a new softmax layer to expand the number of classes the model can learn to predict.
3. For the pretrained CVAE-trained instance, simply add 2 new dense layers between the softmax output layer.
4. For each instance, continue training the model on $\mathscr{D}_T$, with a reduced learning rate, to avoid overfitting on the target domain.
5. While some layers of the models may be optionally frozen during fine-tuning to retain knowledge from the source domain, we discovered that unfreezing the convolutional layers as well as the dense layers yielded the best performance across both instances.

Both models were trained using augmented data via our replication and augmentation approach described in Section 3.3. The training set was purely augmented data while the validation set used 50% of the original FTN data samples. The other 50% of the FTN samples were set aside as test data for model performance assessment after the fine-tuning process was complete. To be clear, both models were fine-tuned against noisy, augmented data while using actual data samples only for validation. This approach required that the models be trained over hundreds of epochs due to the initial difficulty the models encountered with convergence during early epochs. This approach essentially used the training data itself as a form of regularization so as to avoid overfitting while also learning to filter out noise in the signals. Refer to the learning curve in Fig 7 for an illustration of the training process; curves for both models were similar during training.

4.2 Results

Both models were assessed using 50% of the original FTN data samples as the independent test set. Accuracy for both models was similar, with an average of 83.60% for the CVAE fine-tuned model and 83% for the SDSS fine-tuned model. Note that the CVAE fine-tuned model consistently demonstrated better accuracy across all classes, whereas the SDSS fine-tuned model would consistently miss some classes entirely. Refer to Fig. 8 and Fig. 9 for confusion matrix results.

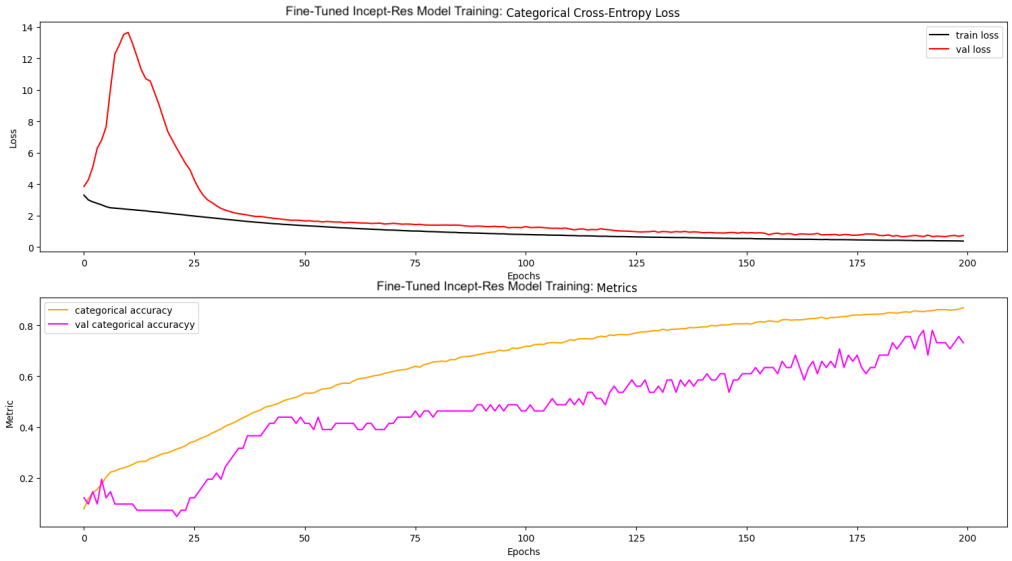


Fig. 7: Learning curves for our fine-tuned CVAE-pretrained instance. The validation loss curve demonstrates difficulties with convergence over the first 30 epochs of training. Notice how validation accuracy, while trending upwards, is choppy and noisy. This is due to the fact that the model is being trained against noisy augmented data while validating against a very small sample set (around 40 samples) of actual FTN data. Note that the training continued out for another 100+ epochs beyond the 200 epochs shown here before overfitting.

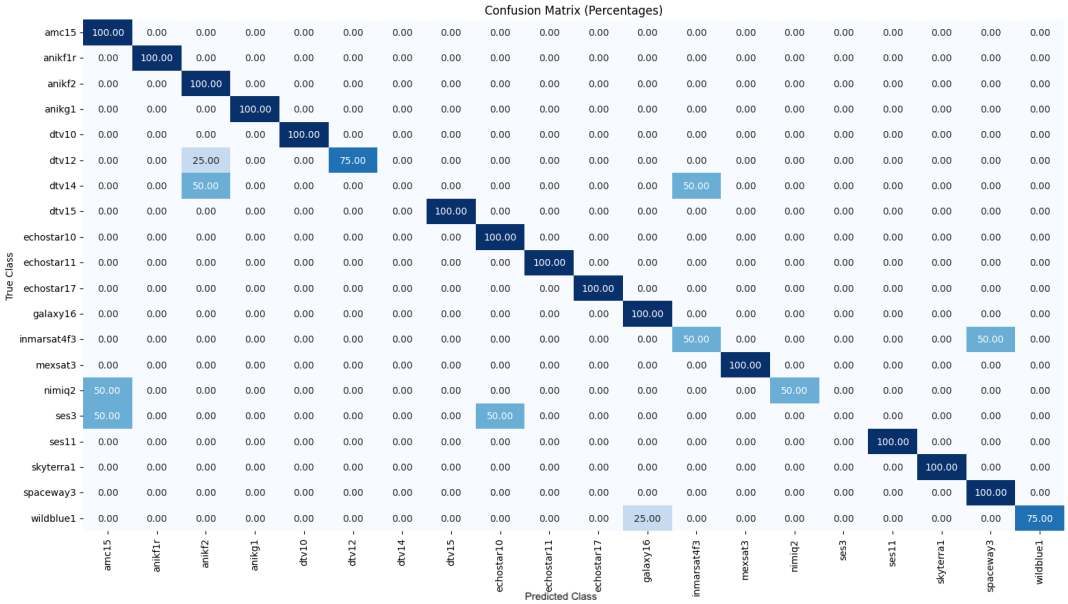


Fig. 8: Confusion matrix for fine-tuned SDSS model. The confusion matrix is normalized by row and shows accuracy percentages over each class. 82.5% overall test accuracy against actual FTN data in this particular instance. Highest accuracy measured was 83.00%

4.3 Discussion

The results from our experiments demonstrate that deep learning models, particularly those based on hybrid architectures like Inception-ResNet, hold significant potential for the classification of GEO satellites using unresolved spectral data. By leveraging pretrained models and applying fine-tuning, we were able to achieve promising results despite the

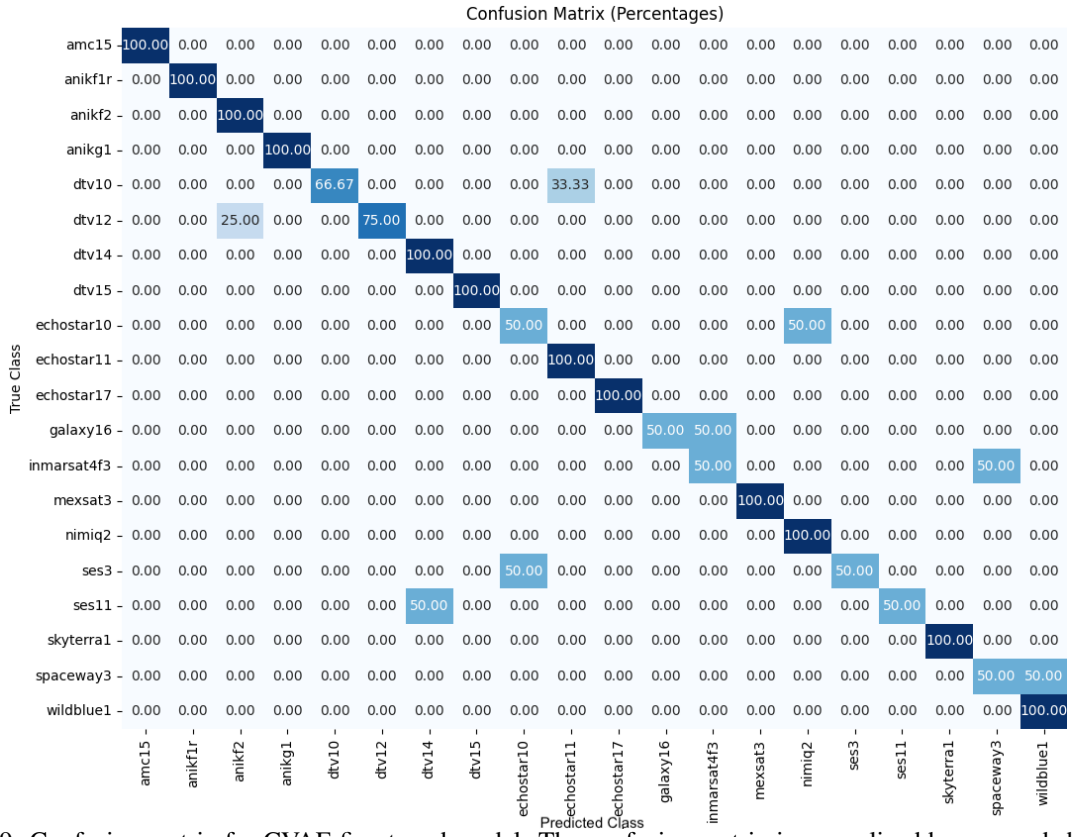


Fig. 9: Confusion matrix for CVAE fine-tuned model. The confusion matrix is normalized by row and shows accuracy percentages over each class. 82% overall test accuracy against actual FTN data in this particular instance. Highest accuracy measured was 83.60%

limited size of our training dataset.

One of the key findings is the superiority of the model fine-tuned using data generated by our CVAE. This model consistently outperformed the model fine-tuned using data from the SDSS. The CVAE-based approach allowed us to mitigate the challenge of data scarcity by generating realistic and diverse satellite spectra, which helped the model generalize better to unseen samples. Additionally, the noisy data augmentation we applied during training proved to be an effective regularization technique, preventing overfitting and enhancing the model's robustness to variations in the data.

However, several limitations remain. The classification accuracy, while promising at 83.6%, clearly indicates that there is room for improvement. Some satellite classes were more difficult for the models to distinguish, as evidenced by the confusion matrices. This is likely due to the inherent similarities between the spectral signatures of certain satellites and the small size of the dataset. The model trained on the SDSS dataset, while effective, struggled with certain classes entirely, highlighting the challenges of applying pretrained models from different domains without sufficient adaptation.

5. FUTURE WORK

Several avenues give way from these findings. First, expanding the dataset will be important to this work. As the Falcon Telescope Network continues to upgrade its instruments and collect more data, we expect that our models can be fine-tuned further, leading to improved performance. Additionally, incorporating more sophisticated data augmentation techniques, such as geometric transformations or more complex noise injection methods, could further improve the model's ability to generalize to new data.

We also plan to experiment with training over individual observations from the FTN dataset rather than averaging them;

this could allow the models to better learn the distributions of spectral features, particularly in cases where satellite orientation or atmospheric conditions cause significant variation in the data. Furthermore, exploring alternative deep learning architectures, such as attention-based models like transformers, may provide a more flexible approach to learning from spectral data.

Lastly, we believe utilizing unsupervised learning techniques that are possible with deep generative models, may help uncover latent patterns in the data that might be overlooked by other methods. Studying the resultant embeddings could lead to better performance in satellite classification as well as enhanced data simulation capabilities.

6. CONCLUSION

This study shows the potential for good performance over spectral data using deep learning architectures like Inception-ResNet, in particular for classifying reflected spectra from GEO satellites. We achieved classification accuracy exceeding 83% across 20 satellite classes, despite using an extremely limited dataset, by fine-tuning pretrained models and leveraging noisy data augmentation with conditional variational autoencoders. As the FTN collects more data we anticipate further improvements in the model's performance with regular fine-tuning or even with periodic complete retraining.

The results of this study underscore the importance of model fine-tuning and data augmentation in overcoming the challenges of small, noisy datasets. Our CVAE-based approach, in particular, offers a scalable solution for generating synthetic data that may enhance model training. While more work remains to be done in expanding the FTN dataset and refining our models, our findings suggest that deep learning holds real promise for contributing to improved situational awareness in space.

[1] Francis K Chun, Roger D Tippets, Michael E Dearborn, Kimberlee C Gresham, Ryan E Freckleton, and Martin W Douglas. The us air force academy falcon telescope network. In *Proceedings of the 2014 AMOS Technical Conference*, 2014.

[2] Francis K Chun, Roger D Tippets, David M Strong, Devin J Della-Rose, Daniel E Polsgrove, Kimberlee C Gresham, Joshua A Reid, Casey P Christy, Mark Korbitz, Joel Gray, et al. A new global array of optical telescopes: The falcon telescope network. *Publications of the Astronomical Society of the Pacific*, 130(991):095003, 2018.

[3] Phan D Dao, Xin C Yee, David M Strong, Benjamin Roth, and Francis K Chun. Multi-geosynchronous satellite classification with spectroscopic signatures. In *Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imaging XXIX*, volume 12519, pages 37–48. SPIE, 2023.

[4] Guoqing Mu and Junghui Chen. Developing a conditional variational autoencoder to guide spectral data augmentation for calibration modeling. *IEEE Transactions on Instrumentation and Measurement*, 71:1–8, 2022.

[5] Markus D Parrish, Joshua A Key, Francis K Chun, David M Strong, Charles J Wetterer, and Philip Castro. Spectral analysis of unresolved satellite imagery. In *Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imaging XXVIII*, volume 12094, pages 120–128. SPIE, 2022.

[6] Qingwei Qiao, Yiding Ping, Jian Chen, Yao Lu, and Chen Zhang. Investigation of factors affecting the reflectance spectra of geo satellites. *Advances in Space Research*, 74(7):3028–3044, 2024.

[7] Ted A Reed, Markus D Parrish, Joshua A Key, Charles J Wetterer, and Philip J Castro. Optimization and automation of the spectroscopy pipeline of the falcon telescope network ted a. reed1, markus d. parrish1, joshua a. key1, charles j. wetterer2, philip j. castro3, david m. strong4, casey p. schuetz-christy1, and francis k. chun1. In *Proceedings of the 2022 AMOS Technical Conference*, 2022.

[8] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[9] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[10] X Yee, P Dao, D Strong, C Wetterer, B Roth, and F Chun. Machine learning classification geos using spectral data. In *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, page 209, 2023.

[11] Donald G York, J Adelman, John E Anderson Jr, Scott F Anderson, James Annis, Neta A Bahcall, JA Bakken, Robert Barkhouser, Steven Bastian, Eileen Berman, et al. The sloan digital sky survey: Technical summary. *The Astronomical Journal*, 120(3):1579, 2000.

A. INCEPTION-RESNET MODEL ARCHITECTURE

This appendix provides a detailed graphical representation of the Inception-ResNet model used in the study. The architecture is large and is presented below.

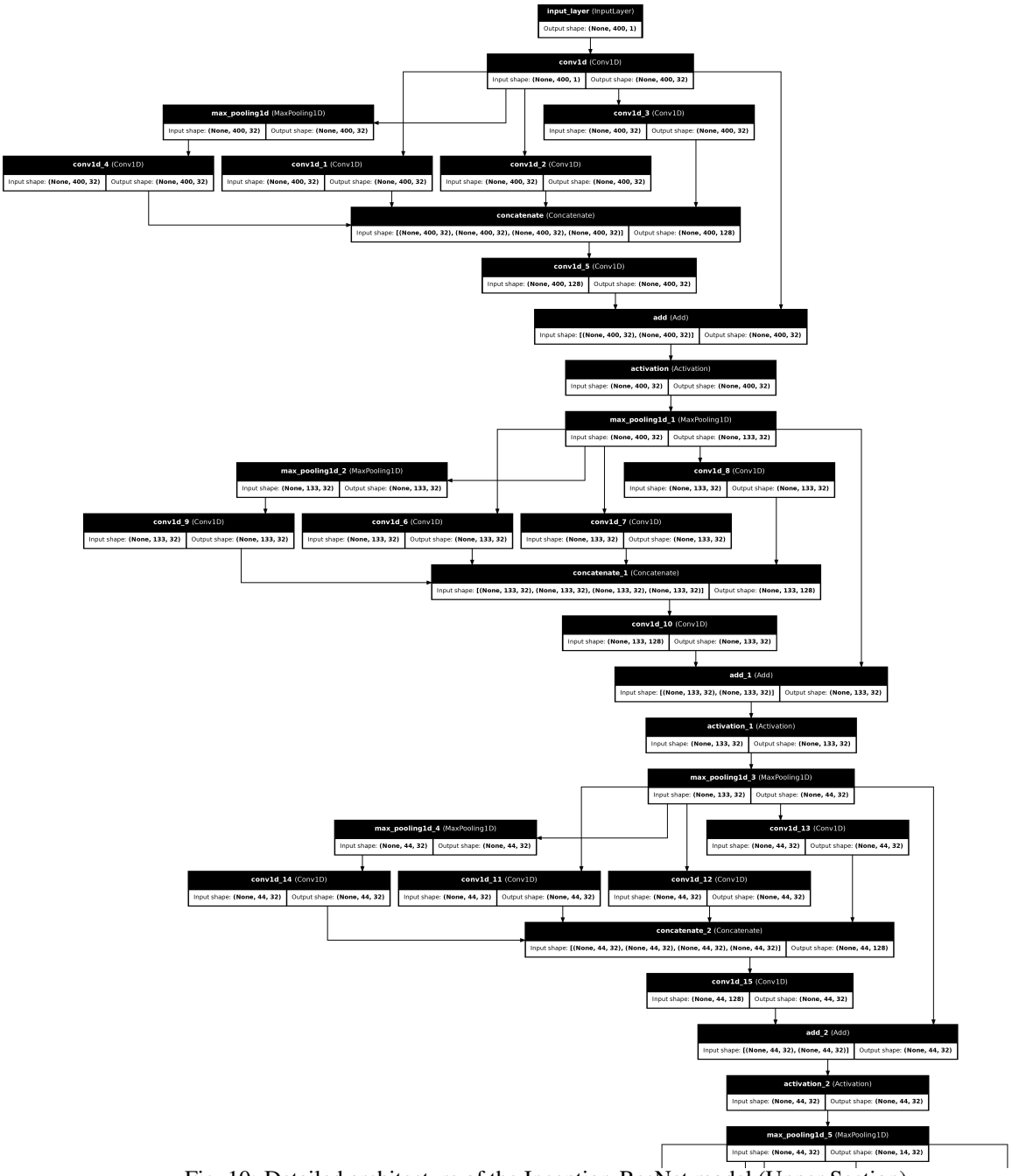


Fig. 10: Detailed architecture of the Inception-ResNet model (Upper Section).

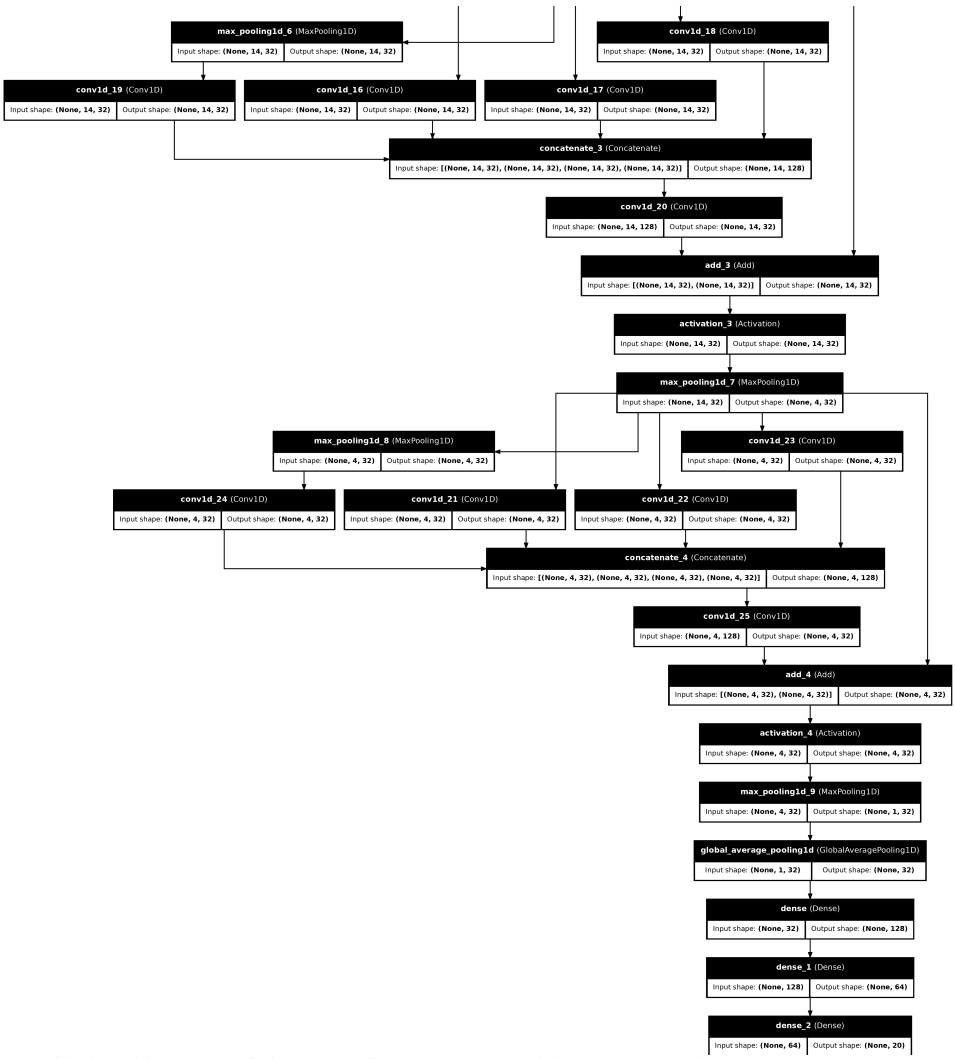


Fig. 11: Detailed architecture of the Inception-ResNet model (Lower Section).