Technische Hochschule Deggendorf
Fakultät Angewandte Informatik

Studiengang Master Artificial Intelligence and Data Science

Vorhersage von Spektralkanälen im Zusammenhang mit der Mineralogie aus CRISM-Spektralbändern mittels Deep Learning

Predicting spectral channels related to mineralogy from CRISM spectral bands using deep learning

Masterarbeit zur Erlangung des akademischen Grades:

Master of Science (M.Sc.)

an der Technischen Hochschule Deggendorf

Vorgelegt von:
Aman Sahani
Matrikelnummer: 12201412

Am: 30. April 2024

Prüfungsleitung:
Prof. Dr. Benedikt Elser

Ergänzende Prüfende:
Prof. Dr. rer. nat. Florian Wahl

Abstract

Remote sensing for planetary exploration involves the use of instruments and technologies to collect data from distant celestial bodies without direct physical contact. This field plays a crucial role in understanding the composition, geology, and atmosphere of planets, moons, and other celestial bodies. With the rapid development in computer vision and generative image models the analysis and generation of data have also become possible.

The Mars Reconnaissance Orbiter (MRO) is equipped with a suite of instruments that collectively generate a vast and diverse array of imaging data for comprehensive investigations of the Martian surface. Notably, the hyper-spectral Compact Reconnaissance Imaging Spectrometer for Mars (CRISM) captures reflective information across multiple spectral bands, spanning the visible to infrared range. However, due to the failure of instrument sensors in CRISM, it is no longer operational and its coverage is partial, limiting the availability of full spectral and spatial resolution data. In contrast, the Colour and Stereo Surface Imaging System (CaSSIS), characterized by high spatial resolution and diverse color capabilities, covers larger areas of the Martian surface.

Stepcenkov et al., 2022, successfully predicted CRISM bands from a Context Camera (CTX) onboard the MRO using traditional image models within the visible spectrum. However, they achieved inferior results when extending this task to bands outside of the visible spectrum, especially in the Near-Infrared (NIR) Channel. Therefore, the motivation of this project is to improve the prediction of CRISM bands outside of the visible range using a diffusion model.

Furthermore, the Mars rover is also currently deployed at the Jezero crater to examine mineralogical deposits in an ancient lake (such as carbonates) which can benefit from this project.

The Stable Diffusion Generative Model (SDGM) presents an innovative methodology for the prediction of spectral signatures. This is achieved through a process of perturbing (noising) and subsequently restoring (denoising) image data, resulting in enhanced precision. In contrast to conventional regression models and Convolutional Neural Network (CNN) image models, SDGM exhibits superior generative capabilities.

This research aims to enhance the accuracy of predicting specific spectral signatures, particularly carbonates in CRISM dataset by using generative capabilities of diffusion models, specifically stable diffusion, and improve interplanetary exploration.

List of Figures

2.1 Unet Architecture . . . 6
2.2 Diffusion model gradually adds Gaussian noise and then reverses the process . 7
2.3 The joint distribution of the latent variables can be expressed as the multiplication of the Gaussian conditional chain transitions. . . . 8
2.4 Stable Diffusion Architecture . . . 9
2.5 Variational autoencoder architecture . . . 10
2.6 Unet training . . . 11
2.7 Contrastive pre-training of CLIP text encoder . . . 11
2.8 Encoding image into latent space and then decoding it using encoder decoder 12
2.9 forward and reverse diffusion process . . . 13
2.10 Conditioning mechanism in detail . . . 13
2.11 Detailed stable diffusion architecture . . . 14

3.1 Crism MTRDR products . . . 18
3.2 CaSSIS response filters against wavelengths . . . 18
3.3 4 channel frt000047a3_07_if166j formed by applying CaSSIS filters on CRISM IF data . . . 19
3.4 3 channel frt000047a3_07_if166j formed by applying CaSSIS filters pan, nir,red on CRISM IF data . . . 19
3.5 PCA converted image of frt000047a3_07_if166j . . . 21
3.6 Rectangular filters . . . 22
3.8 False color image of frt000047a3_07_if166j, visible green patches are carbonate signatures . . . 22
3.7 frt000047a3_07_if166j after applying rectangular filters . . . 23
3.9 Model architecture of InstructPix2Pix optimized for the task, text encoder is later fed with metadata . . . 26

5.1 Unet training loss per epochs . . . 33
5.2 Input Image, Ground truth (image with carbonate signatures) and unet predicted image shows missing carbonate signature in unet predicted image . . . 34
5.3 Input Image, Ground truth (image with carbonate signatures) and unet predicted image shows if carbonates are missing unet is producing images very similar to ground truth . . . 35
5.4 Unet not producing good results on PCA dataset . . . 36
5.5 Unet not producing good results on PCA dataset . . . 37
5.6 Stable Diffusion predicting carbonates . . . 38
5.7 Stable Diffusion predicting carbonates . . . 39

5.8 Stable Diffusion predicting carbonates even if they are not present 39
5.9 Stable Diffusion not predicting carbonates if they are not present 41
5.10 Stable diffusion with reduced bias 42
5.11 Stable Diffusion cannot pinpoint the carbonates pixels 43
5.12 Stable Diffusion cannot predict carbonates 44
5.13 Stable Diffusion correctly predicting carbonate signatures. 45
5.14 Stable Diffusion performing well but occasionally missing carbonate pixels. . . 46

Contents

Abstract iii

List of Figures v

1 Introduction 1
- 1.1 Motivation . . . 2

2 Literature Review 3
- 2.1 Related Work . . . 3
- 2.2 CRISM and CASSIS dataset . . . 4
  - 2.2.1 CRISM . . . 4
  - 2.2.2 CASSIS . . . 5
- 2.3 Unet Model . . . 5
- 2.4 Introduction to Diffusion models . . . 7
- 2.5 Stable Diffusion . . . 8
  - 2.5.1 Working of stable diffusion model . . . 12
- 2.6 Instruct pix2pix . . . 14

3 Methodology 17
- 3.1 Dataset Collection and Creation . . . 17
  - 3.1.1 CRISM dataset creation . . . 17
  - 3.1.2 Data Augmentation . . . 22
- 3.2 Baseline Model: Unet . . . 23
- 3.3 InstructPix2Pix Diffusion Model . . . 24

4 Experimental Setup 28
- 4.1 Hardware and Software Requirements . . . 28
- 4.2 Performance Metrics . . . 29

5 Experiments and Results 33
- 5.1 Unet model . . . 33
  - 5.1.1 PAN,NIR,RED Dataset . . . 33
  - 5.1.2 Data Skewness . . . 34
  - 5.1.3 PCA Dataset . . . 36
- 5.2 Stable Diffusion . . . 37
  - 5.2.1 PAN, NIR, RED Dataset . . . 37
  - 5.2.2 Detecting Bias and adding metadata . . . 38
  - 5.2.3 Performance of Stable Diffusion on PCA Dataset . . . 40

5.3 Comparison of model performance 43
5.4 Performance of Stable Diffusion on Rectangular Filter Dataset 45

6 Conclusion and Future Work **48**
6.1 Conclusion 48
6.2 Future Work 49

1 Introduction

Hyperspectral imaging data holds significant importance in exploring the composition of distant planets and comprehending their surface characteristics. Utilizing reflectance spectroscopy, minerals can be identified by comparing their recorded reflectance spectra with laboratory data. This has found significant applications in various fields, including geology, agriculture, environmental monitoring, and, notably, planetary science. The resulting insights into the mineralogical composition of these bodies enable scientists to conduct detailed studies of regions that pique scientific curiosity.

In order to create RGB browse products, which provide a qualitative representation of the spectral variability on the surface within the image, specific spectral bands are combined to calculate spectral parameters. However, the pursuit of broad bandwidth and high spectral resolution often necessitates compromises in other desirable attributes, such as spatial coverage. As a result, the design of imaging instruments is often tailored to meet specific research goals and applications.

NASA's Mars Reconnaissance Orbiter (MRO) [1] houses a variety of instruments onboard which provide a large quantity and variety of imagining data for investigations of the Martian surface. A prominent instrument to acquire spectral data from orbit to determine mineralogical properties of specific areas is the Compact Reconnaissance Imaging Spectrometer for Mars or CRISM [2], in contrast, The Colour and Stereo Surface Imaging System (CaSSIS) (Thomas et al., 2017 [3]) on the European Space Agency's (ESA) provide high spatial resolution and high color diversity.

CRISM, boasting over 400 spectral bands ranging from visible to near-infrared, enables targeted observations in areas of scientific interest. Nonetheless, the limited size of imaged areas and spatial resolution prompts a strategic selection of observation sites. Targeted observations are therefore conducted in areas of scientific interest, which were discovered using untargeted observations or other instruments. A vast number of photos were taken with high resolution in both spectral and spatial aspects, encompassing less than one percent of the Martian surface [4]. CaSSIS on the other hand is a high-resolution camera system onboard the European Space Agency's (ESA) ExoMars Trace Gas Orbiter (TGO). It is a scientific instrument that captures images of the Martian surface in color and stereo. This allows scientists to study the geology, morphology, and composition of the Martian surface in great detail however, it lacks the bands present in CRISM which contain only 4 bands.

1.1 Motivation

The motivation for this thesis stems from the inherent dichotomy between the depth of information embedded in CRISM's multitude of bands and the broader spatial coverage facilitated by CaSSIS. While CRISM's high spectral resolution enables detailed analyses of specific regions, CaSSIS, with its reduced set of bands, captures a more expansive view of the Martian surface. The synthesis of these datasets offers a unique opportunity to harness the spectral richness of CRISM while leveraging the broader spatial coverage afforded by CaSSIS. The specific objective of this thesis is to take images from CRISM, chop the special information down to the 4 bands, that we find in CaSSIS as well, and predict spectral channels related to mineralogy (which are available in the original CRISM data) thus enhancing the predictive capabilities of Martian mineralogical signatures. The choice of employing a Diffusion Generative Model for this task is rooted in the independence of the spectral bands within the CRISM dataset. The diffusion model might be better than the traditional image model due to the following points :

1. **Independence of Spectral Channels:** Traditional image models often assume pixel-wise independence, while the Diffusion Generative Model leverages the inherent independence of spectral channels.
2. **Non-Linearity in Spectral Signatures:** Spectral signatures, especially those associated with minerals, can exhibit non-linear and intricate patterns. Diffusion models can better handle these non-linear relationships.
3. **Robust Handling of Noise and Uncertainty:** The Stable Diffusion Generative Model inherently accounts for uncertainty in data, providing robust predictions in the presence of noise and uncertainties.

2 Literature Review

In recent years, diffusion models have garnered significant attention as powerful tools for various tasks in machine learning and computer vision, demonstrating their efficacy in capturing complex data distributions. The diffusion model, particularly exemplified by the success of generative models like the Stable Diffusion Generative Model, has shown promise in applications beyond traditional areas. Machine learning has also become an important tool in the analysis of remotely sensed data and planetary science in general.

2.1 Related Work

Hyperspectral images, which capture bands over hundreds of wavelengths of the electromagnetic spectrum, have piqued researchers' curiosity in the last two decades. These images have been used for various applications, including land cover classification, anomaly detection, plant classification, etc. The process of analyzing hyperspectral images has become more efficient and accurate with the advent of deep learning techniques [5]. Hyperspectral imaging has been developed for mineral exploration, with a focus on imaging the Earth's surface using the visible (0.4 μm) to near-infrared (2.5 μm) part of the electromagnetic spectrum to map various mineral species [6].

Machine learning has become an essential technique for interpreting remotely sensed data in Earth and planetary sciences over the last few years. Machine Learning and artificial intelligence have many applications for Earth as well as other planets, On Earth, these applications encompass tasks like categorizing land cover, identifying targets, separating mixed data, and estimating physical and chemical parameters. These applications employ a wide range of methods and model architectures [7, 8]. Deep learning techniques have been used for detecting exoplanets by combining real and synthetic data[9]. These techniques have been used to analyze large volumes of data from various space missions and telescopes, providing valuable insights into the characteristics of celestial bodies [10]. Notably, studies involving CRISM data primarily focus on surface mineralogy identification and classification [11].

In the realm of machine learning, this work centers predominantly on image-to-image translation, a field crucial for generating output images based on input photos. By utilizing coregistered pairs of input and output images for training, a generalized mapping is acquired, enabling the transformation of novel images. Once a generalized mapping is learned, it can be applied to transform new images, such as converting aerial photographs into maps or altering day-to-night images [12].

Many segmentation architectures rely on effective encoding networks that are interconnected in various ways. VGG [13] stands as an early example of such an encoder, utilizing convolutional layers, pooling operations, and non-linearities. Later advancements introduced the concept of residual blocks with skip connections, making deep networks easier to train by learning residual functions differing from identity [14]. Dense blocks were further developed, where each layer incorporates all preceding feature maps of the same size, addressing issues like vanishing gradients and promoting feature propagation and reuse [15].

Stepcenkov et al. 2022 [16] showed that segmentation models, specifically U-Net can achieve the prediction of continuous spectral reflectance and can produce CRISM and CTX image pairs therefore UNet model is used as a baseline model for this task however he achieved inferior results when predicting channels in NIR bands, therefore I explored diffusion model.

Recent advancements in machine learning have introduced diffusion models as powerful tools for capturing complex patterns within data. One notable success story of these models is their application in image colorization [17].

2.2 CRISM and CASSIS dataset

2.2.1 CRISM

The Compact Reconnaissance Imaging Spectrometer for Mars (CRISM) located on the Mars Reconnaissance Orbiter (MRO), is a crucial tool used to analyze the mineral composition of the Martian surface using spectroscopic techniques. Images from the CRISM instrument are the primary focus of this work. CRISM employs a comprehensive scanning configuration to methodically investigate extensive landscapes with reduced spatial and spectral resolutions, thereby finding potential areas for further detailed examination. The initial phase of reconnaissance is followed by focused observations, which provide improved spatial resolution and spectral accuracy, enabling comprehensive mineralogical analysis. Such locations are then mapped with the full spatial resolution of 15 mm pixel-1 pixel-1 to 19 mm pixel-1 pixel-1 and full spectral resolution of 362 nm–3920 nm at 6.556.55 nm per channel in targeted mode. The goal of CRISM's mission is to describe the crustal mineralogy on the whole surface. [2]. By analyzing the spectral images captured by CRISM, scientists have been able to identify key absorptions at infrared wavelengths, which correspond to different minerals. This has provided invaluable insights into the geological history of Mars and the role of water in altering its landscape [18]. Various techniques have been particularly effective in identifying silicate and carbonate minerals, which are major components in certain regions of Mars[19]. Hyperspectral imaging data, such as those captured by CRISM, are the basis of visible-near infrared reflectance spectroscopy.

CRISM utilizes spectral reflectance qualities to identify mineral elements within a range of 0.4 to 4 microns where each pixel represents a reflectance spectrum with characteristic absorption bands depending on the materials. It is worth mentioning that iron oxide manifestations,

such as rust, display noticeable red shades in CRISM images. In addition, the infrared sensitivity of CRISM reveals supplementary mineral signals, such as those that are suggestive of sulfate, carbonate, hydroxyl, and water-bearing minerals.

In order to visually highlight or distinguish minerals within an image or scene, browse products are created. These are synthesized RGB color images, where each channel is assigned to a summary parameter, such as band depth or some other calculated measure of spectral variability, and afterward stretched to a specific range.

2.2.2 CASSIS

Colour and Stereo Surface Imaging System (CaSSIS) was developed by the University of Bern [3]. CaSSIS is a high-resolution imaging system that works alongside other instruments on EMTGO to provide additional data about the surface of Mars. It builds upon the observations made by the High-Resolution Imaging Science Experiment (HiRISE), which is currently onboard NASA's Mars Reconnaissance Orbiter (MRO) and orbiting Mars. CaSSIS consists of two major units: The rotation Unit and Electronics Unit (ELU). The telescope and focal plane are on this rotation unit so that they can be swiveled and rotated to compensate for nadir direction and orbiting speed. The ELU houses the circuit boards necessary for the camera's operation.

CaSSIS takes high-resolution stereo images in 4 colours or channels namely Blue channel (BLU), Panchromatic Channel (Pan), Near-Infrared (NIR) Channel, and Red Channel (RED), of the Martian surface. The instrument focuses on areas that have been identified as possible producers of trace gases. With its observation of a 9 km broad swath, CaSSIS offers the greatest color imaging of Mars to date[3].

The CaSSIS telescope was originally conceived as a three-mirror anastigmat system (off-axis) with a fold mirror. The primary mirror is around 13.5 cm in diameter. The mirrors are held in a carbon fiber-reinforced polymer (CFRP) structure with a focal length of 875 mm. The focal plane will comprise a single silicon hybrid detector with 4 colour filters mounted on it. [3]

CaSSIS is intended to acquire moderately high-resolution (4.6 m/pixel) targeted images of Mars at a rate of 10–20 images per day from a roughly circular orbit 400 km above the surface2. A typical product from one image acquisition will be a 9.5 km x 45 km swath in full colour and stereo in one over-flight of the target. This reduces atmospheric influences inherent in stereo and colour products from previous high-resolution imagers [3].

2.3 Unet Model

The U-Net is a convolutional neural network (CNN) architecture designed for semantic image segmentation, which is the task of classifying each pixel in an image into a specific class. It was introduced by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015 [20] and has since become a popular choice for medical image segmentation and other related tasks.

The U-Net architecture, represented in Fig. 2.1 [20], is characterized by a U-shaped structure, consisting of a contracting path (encoder) and an expansive path (decoder). The contracting path captures context and reduces the spatial resolution, while the expansive path restores the resolution and refines the segmentation.

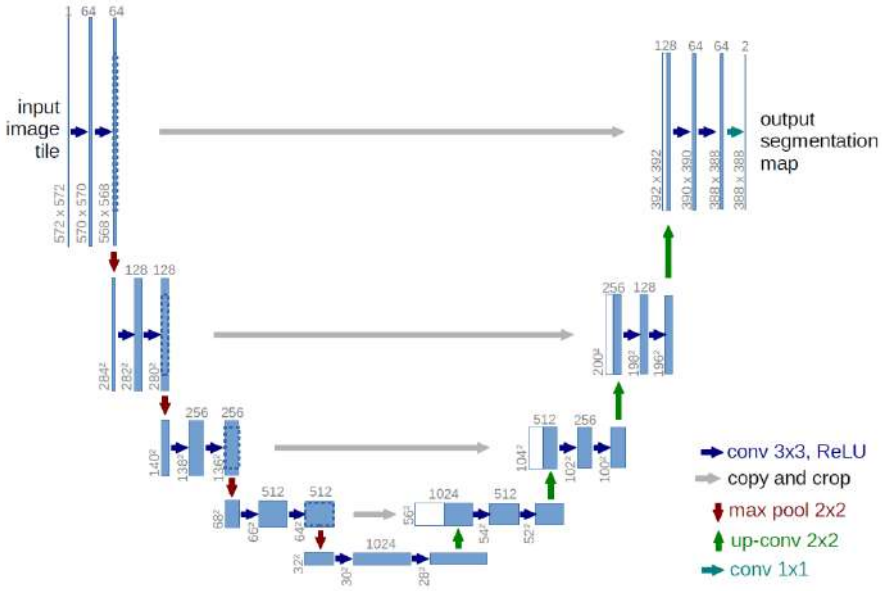

Figure 2.1: Unet Architecture

1. **Contracting Path (Encoder):**
   - The contracting path starts with the input layer representing the original image.
   - Convolutional layers with small 3×3 filters are commonly used for feature extraction. Each convolutional layer is followed by a rectified linear unit (ReLU) activation function to introduce non-linearity.
   - After each convolutional layer, max-pooling operations with 2×2 kernels are applied to reduce spatial dimensions and increase the receptive field. Max pooling helps in capturing the most important features while discarding spatial information.
2. **Bottleneck:**
   - The bottleneck connects the contracting and expansive paths.
   - It typically consists of multiple convolutional layers with ReLU activation functions. These layers capture the most essential features of the input data.
   - Dropout or batch normalization may be used in the bottleneck to improve generalization and prevent overfitting.

3. **Expansive Path (Decoder):**
   - The expansive path is the mirror image of the contracting path.
   - Up-sampling layers are used to increase the spatial resolution of the feature maps. Transposed convolutions (also known as deconvolutions) with 2×2 kernels are commonly employed for this purpose. Up-sampling helps in restoring the lost spatial information during the contracting path.
   - Concatenation is performed with the corresponding feature maps from the contracting path to provide detailed information at higher resolutions. This is a crucial aspect of the U-Net architecture and is achieved through skip connections.
   - Convolutional layers with 3×3 filters and ReLU activation functions are used for feature refinement in the expansive path.
4. **Output Layer:**
   - The final layer uses a 1×1 convolution to map the features to the desired number of output channels/classes.

2.4 Introduction to Diffusion models

Diffusion models have emerged as a significant development in the field of machine learning especially in the field of computer vision. These generative models have attracted significant attention because of their remarkable capabilities, including outperforming Generative Adversarial Networks (GANs) [21] in the synthesis of images

Diffusion models [22] are a class of generative models that learn to produce data similar to the data they are trained on by reversing a process of adding noise to the data, figure 2.2 represents a sample flow of the diffusion process.

Figure 2.2: Diffusion model gradually adds Gaussian noise and then reverses the process

Diffusion models consist of three main components: the forward process, the reverse process, and the sampling procedure.

1. **Forward diffusion process:** The forward diffusion process is the Markov chain of diffusion steps in which we slowly and randomly add Gaussian noise to the data until it becomes indistinguishable from random noise.

$$x_t = x_{t-1} + \epsilon_t \tag{2.1}$$

where ϵ_t is Gaussian noise with zero mean and unit variance at time step t. This process can be used to generate new data from random noise by applying a reverse process.

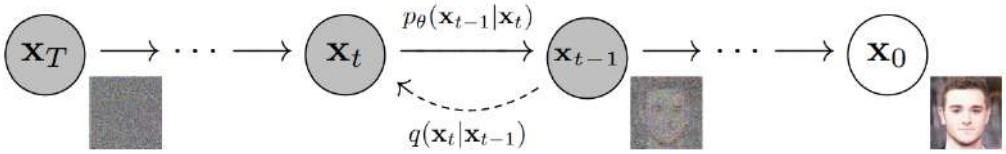

Figure 2.3: The joint distribution of the latent variables can be expressed as the multiplication of the Gaussian conditional chain transitions.

2. **Reverse diffusion process:**The reverse process learns to remove the noise and recover the original data by following a Markov chain that depends on the latent variables of the data.

   The objective here is to learn the reverse process i.e. training p so new data can be generated by traversing backward along this chain.

$$p(x_t|x_{t-1}, \epsilon_t) = \frac{\exp(-\beta(x_t - x_{t-1}))}{\sum_{s=0}^{\infty} \exp(-\beta(x_s - x_{s-1}))} \tag{2.2}$$

   where β is a hyperparameter controlling the strength of the reverse process, and ϵ_t and x_t are defined as before.

   Figure 2.3 shows the forward and reverse diffusion process across a Markov chain [23]

3. **Sampling procedure:** The sampling procedure uses this learned Markov chain to generate new data from random noise

$$y = \sigma(\mu + \epsilon) \tag{2.3}$$

   where μ and σ are learned parameters defining a Gaussian distribution over y, and ϵ is Gaussian noise with zero mean and unit variance.

2.5 Stable Diffusion

Stable Diffusion is a latent text-to-image diffusion model, as a Latent diffusion Model (LDM) it operates by repeatedly reducing noise into a latent space and then denoising that representation into an image. It is primarily used to generate detailed images conditioned on text descriptions, though it can also be applied to other tasks such as inpainting, outpainting, and generating image-to-image translations guided by a text prompt.

The procedure of generating an image by Stable Diffusion involves an iterative diffusion process to the first noise and then denoise the image. The diffusion coefficient is computed by the

algorithm in each iteration, taking into account local picture properties such as gradients and edges4. The proposed approach utilizes a diffusion model that has been trained to effectively remove Gaussian noise from photos with blurriness. This model iteratively refines the images until a sharp and clear outcome is attained.

The researchers from the CompVis developed stable diffusion based on the paper High-Resolution Image Synthesis with Latent Diffusion Models [24]. The model is trained on 512x512 images from a subset of the LAION-5B, the code and model weights of the software have been made available to the public, and it is compatible with a wide range of consumer hardware that possesses a moderate GPU.

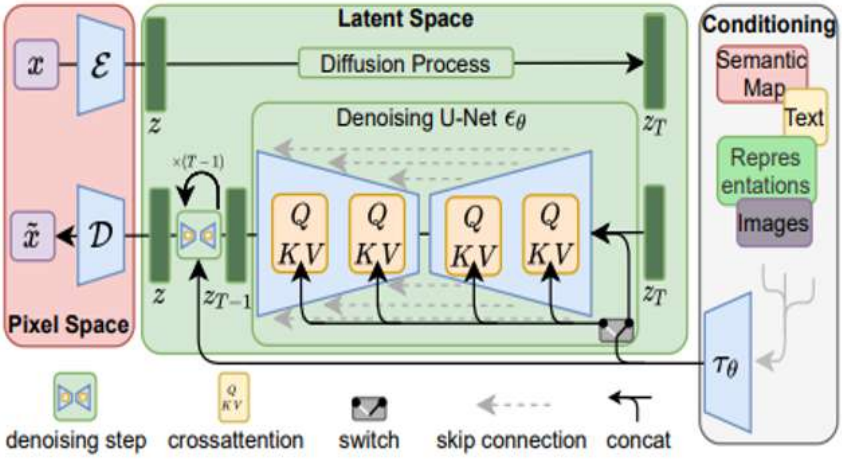

Figure 2.4: Stable Diffusion Architecture

The Stable Diffusion model is a type of diffusion model that decomposes the image formation process into a sequential application of denoising autoencoders. Figure 2.4 shows stable diffusion architecture[24], it has three primary components

1. **Variational Autoencoder-Decoder**: This component generates the image by applying the predicted noise to the previous image in the sequence.
2. **Text Encoder**: This component encodes the input text prompt into a latent space representation.
3. **UNet Noise Predictor**: This component predicts the noise to be added to the image at each diffusion step.

Variational Autoencoders

Autoencoders are neural networks that learn the best encoding-decoding scheme using an iterative optimization process. The search for an encoder and decoder that minimize reconstruction error is done by gradient descent over the parameters of these networks. The limitation of autoencoders is that they cannot create new data.

Figure 2.5 shows a variational autoencoder introduced in 2013 by Diederik P. Kingma and Max Welling [25] is an architecture composed of both an encoder and a decoder trained to minimize the reconstruction error between the encoded-decoded data and the initial data. To introduce regularization, the model is modified by encoding an input as a distribution over the latent space, sampling a point from that distribution, decoding the sampled point, and backpropagating the reconstruction error through the network. Thus, instead of producing a solitary output value, the encoder generates a probability distribution in the bottleneck layer. Nevertheless, VAEs are not without their limitations, including the ability to generate data with smoothness, the limited representation of data distribution, and the occurrence of mode collapse.

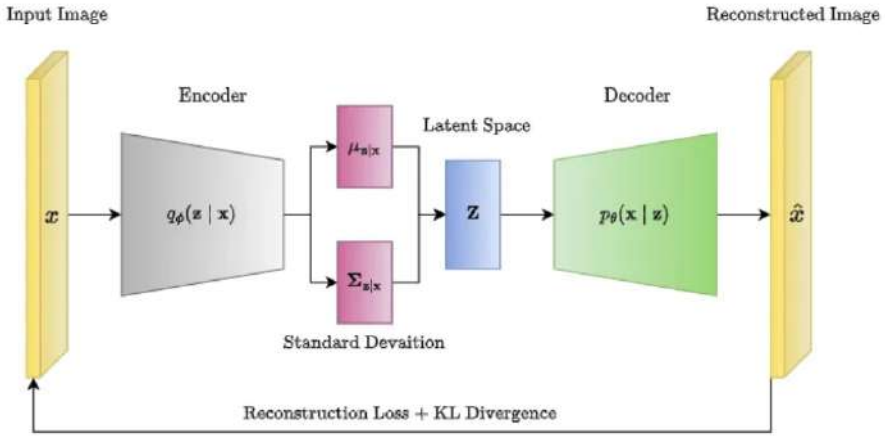

Figure 2.5: Variational autoencoder architecture

Unet Noise predictor

The Unet noise predictor has the same architecture as explained above, the model predicts a denoised picture representation of noisy latent variables. In the context of stable diffusion, the model takes noise latent as input and produces noise in the latent as output, and learns the noise thus by subtracting the noise from the noise latent we can regenerate the image. Figure 2.6 shows a visual representation of Unet training on noise latent [26].

Text Encoder

The text encoder converts the input prompt into an embedding space, which is then used as input for the U-Net. This serves as a reference for handling noisy latent variables during the training of Unet for its denoising procedure. Stable diffusion uses a pre-trained text encoder

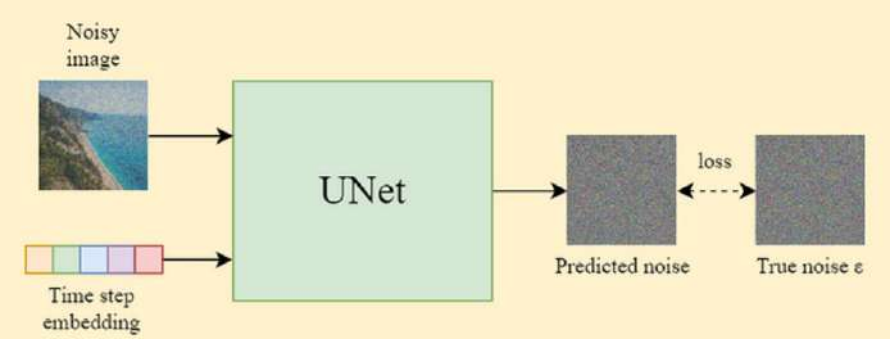

Figure 2.6: Unet training

called CLIP. Contrastive Language-Image Pre-Training (CLIP) model introduced by Radford in 2021 [27] is a neural network trained on image-text pairs. It operates on the principle of learning a joint encoding space for both the image and its corresponding text caption as well as to minimize the similarity between pairs that do not correspond to each other. The model tries to bring the representations of the image and its corresponding caption closer together in the encoding space but if an image and a caption that are not related, the model will adjust its parameters to push their representations further apart in the encoding space, this is achieved through a process known as contrastive learning. Figure 2.7 shows the working of CLIP text encoder with an image [27].

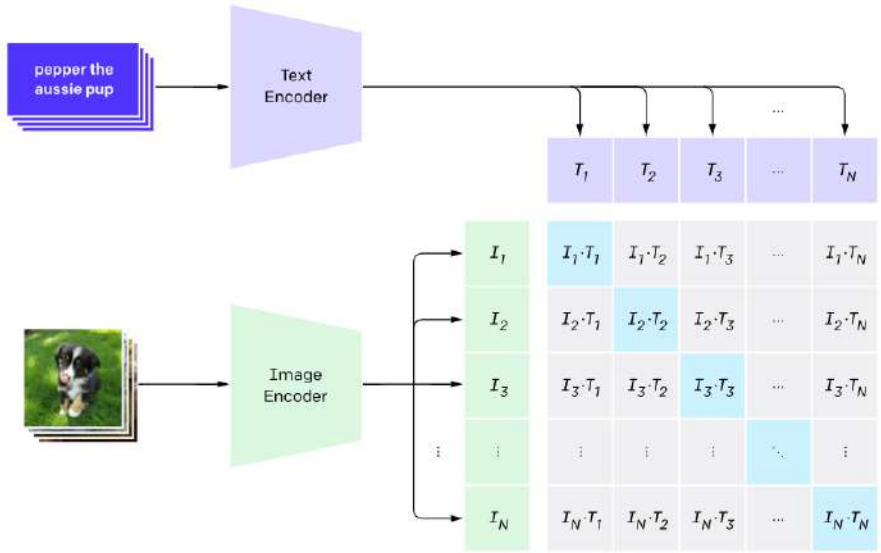

Figure 2.7: Contrastive pre-training of CLIP text encoder

2.5.1 Working of stable diffusion model

Encoding images into latent space
The first step is to train an autoencoder to effectively compress images into lower-dimensional representations that is latent space. This entails two key steps: firstly, utilizing the trained encoder (E) to encode the complete image into a compressed, lower-dimensional latent data format. Subsequently, employing the trained decoder (D) to decode this latent data, thus reconstructing the original image. Figure 2.8 shows the architecture of encoding and decoding images into latent space [26].

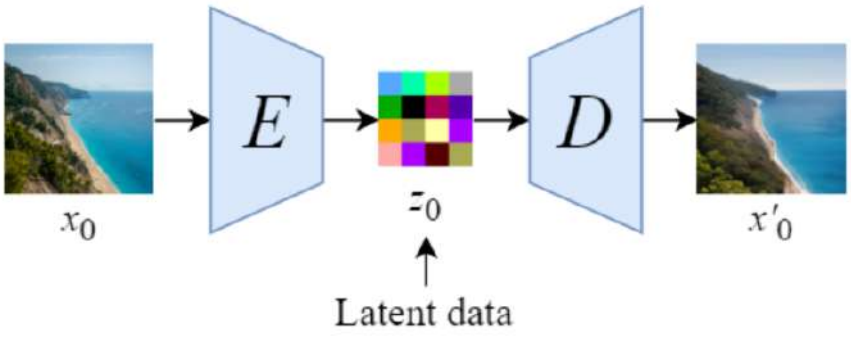

Figure 2.8: Encoding image into latent space and then decoding it using encoder decoder

Adding noise to the image
During forward diffusion, a random timestamp is selected and then the noise is added to the latent data iteratively until the selected timestep, while during reverse diffusion, the noise is removed from the latent data. Figure 2.9 shows the forward and reverse diffusion process for latents [26].

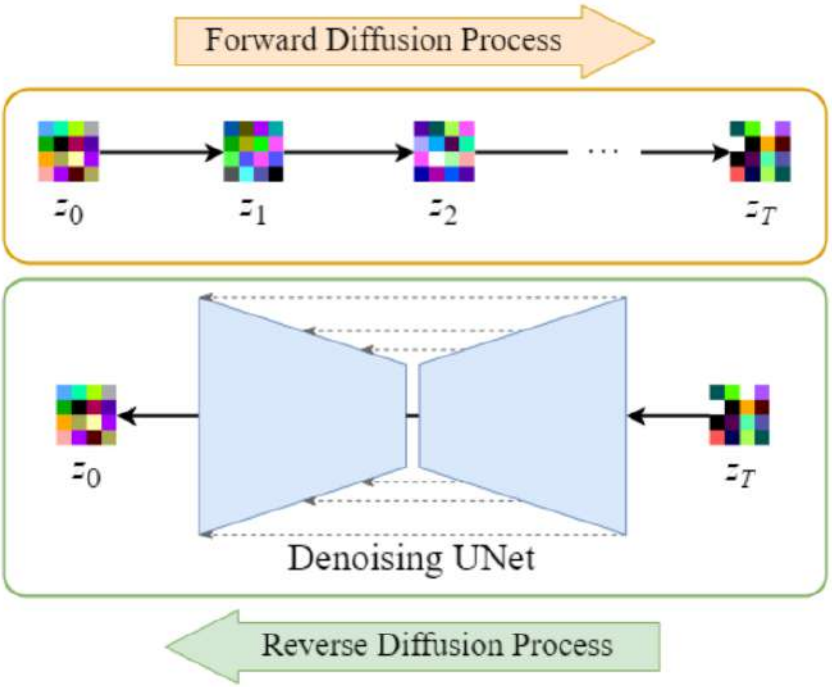

Figure 2.9: forward and reverse diffusion process

Conditioning

The text inputs are converted to vector embeddings using CLIP text encoder and then these embeddings are mapped into the U-Net through the attention (Q, K, V) mechanism. Figure 2.10 [26] shows the conditioning mechanism in detail.

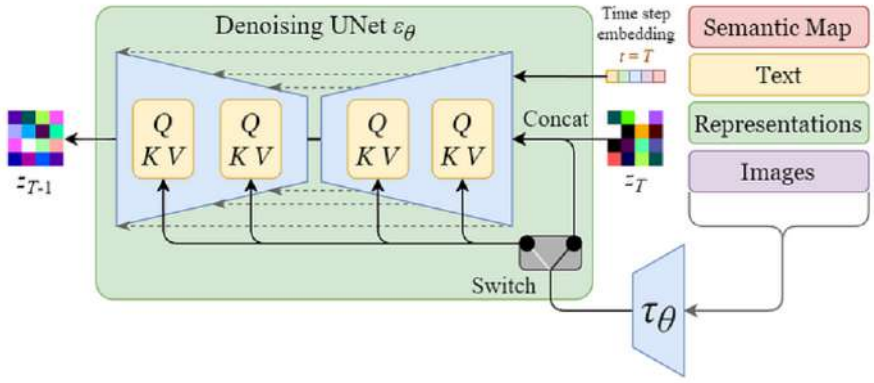

Figure 2.10: Conditioning mechanism in detail

Training

Based on the image-conditioning pairs, the training objective or the loss function is :

$$L = \mathbb{E}_{E(x),y,\epsilon\sim N(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta)\|_2^2\right]$$

where τ_θ (added conditioning input to the U-net) and ϵ_θ are jointly optimized, where z_t is the input latent data.

The detailed architecture combining all the above methods is depicted in Figure 2.11 [26]

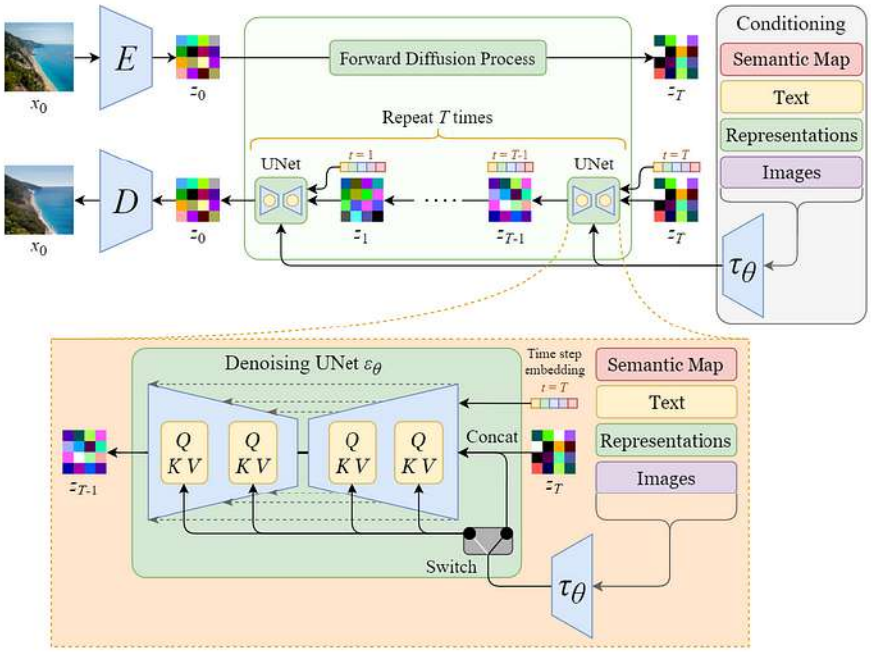

Figure 2.11: Detailed stable diffusion architecture

2.6 Instruct pix2pix

InstructPix2Pix, developed by Tim Brooks, Aleksander Holynski, Alexei A. Efros in 2024 [28], is a sophisticated machine-learning model designed for image editing based on human instructions to perform a diverse collection of edits such as replacing objects, changing the style of an image, changing the setting, and the artistic medium, among others. It operates on the principle of conditional diffusion and is trained on generated data, allowing it to generalize to real images and user-written instructions at inference time.

InstructPix2Pix uses the knowledge of two large pre-trained models - a language model (GPT-3) and a text-to-image model (Stable Diffusion) - to generate a large dataset of image editing examples. The authors generated the dataset using Prompt-to-Prompt, a recent method

aimed at encouraging multiple generations from a text-to-image diffusion model to be similar to generate paired images from paired captions for training of the model.

InstructPix2Pix is based upon stable diffusion and an additional input channel is added to the first convolution layer to accept an image with its image condition and then is fine-tuned using existing weights from stable diffusion to achieve best results.

3 Methodology

In this section, I will discuss a detailed description of the methods and procedures followed to prepare datasets and implement unet and stable diffusion to predict CRISM band images. This section serves as a roadmap for how this research is conducted.

3.1 Dataset Collection and Creation

All CRISM images were acquired at the most recent, publicly available, calibration level, called Map-projected Targeted Reduced Data Record (MTRDR). The CRISM MTRDR products are sophisticated empirical and statistically corrected sets to remove spikes, rectify for imaging geometry and gimbal motion, and remove atmospheric contamination to obtain approximate surface reflectance [11]. It integrates a series of standard and empirical spectral corrections, spatial transforms, and parameter calculations, and renders good-quality visualization products. It provides 2D spatially resolved spectra over a wavelength range of 362 nm to 3920 nm at 6.55 nm/channel. The spatial resolution is typically around 18 m/px. Pelkey et al. [4] and Viviano et al. [29] the feature set of "image products" derived from CRISM spectra, which exhibit a high correlation with the geochemical composition of the Martian surface. Available CRISM products can be searched using the Orbital Data Explorer (ODE) REST API (**https://oderest.rsl.wustl.edu/#ODERESTInterface**) and can be downloaded. All the files and related data can be found at the Geosciences Node of NASA's Planetary Data System (PDS) (**https://pds-geosciences.wustl.edu/**). I already had the CRISM dataset in MTRDR format available in the hub.

3.1.1 CRISM dataset creation

The CRISM .hdr files can be read using the spectral library in Python. The data also include a .txt file that lists wavelength information for the spectral image cube as well as a list of summary band products. The CRISM MTRDR format contains several products[30] explained in figure 3.1.

In the context of this research, we will be working on correct I/F or IF and refined spectral summary parameters or SR to generate input and output images for the model respectively.

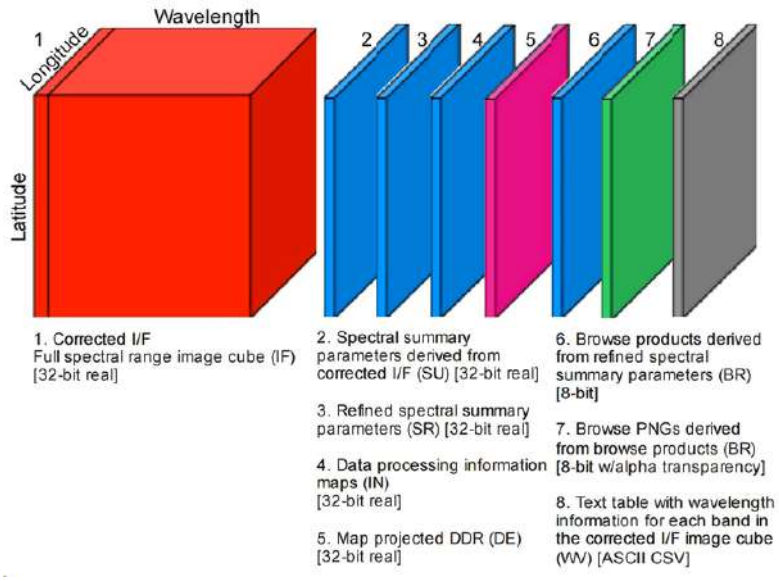

Figure 3.1: Crism MTRDR products

Applying CASSiS Filters

The CaSSIS filters can be obtained from the hub in .txt format, they contain normalized responses along with wavelengths for the 4 channels that are - Blue channel (BLU), Panchromatic Channel (Pan), Near-Infrared (NIR) Channel, and Red Channel (RED). The response filters can be better visualized in Figure 3.2.

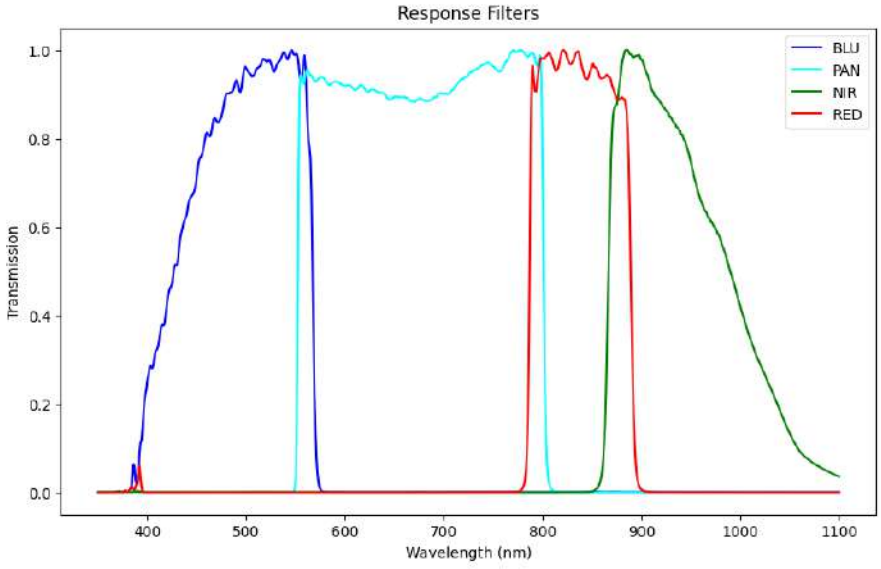

Figure 3.2: CaSSIS response filters against wavelengths

To get the input image I applied these filters to the IF images and then stacked them together to create a 4-dimensional array, I then reshaped and normalized the array in the form of an image with shape (512,512,4). Figure 3.3 displays a CRISM data frt000047a3_07_if166j converted into a 4-dimensional image.

Figure 3.3: 4 channel frt000047a3_07_if166j formed by applying CaSSIS filters on CRISM IF data

Since the stable diffusion model is trained on 3-dimensional images, I need to reshape them into three channels to utilize the trained weights of stable diffusion. Selected 3 channels: PAN, NIR, and RED as these 3 channels contain important information. Figure 3.4 shows the image formed by these three channels and converted to RGB.

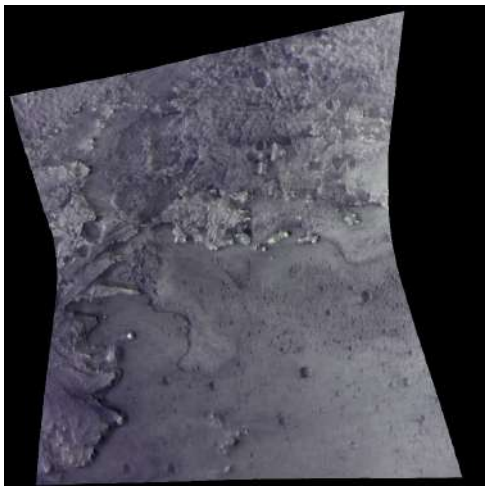

Figure 3.4: 3 channel frt000047a3_07_if166j formed by applying CaSSIS filters pan, nir,red on CRISM IF data

PCA for dimension reduction

Principal Component Analysis (PCA) [31] is a widely used technique in data analysis and image processing for reducing the dimensionality of a dataset while retaining its essential information. This method is particularly useful for handling high-dimensional data, such as images, by transforming the original features into a new set of uncorrelated variables called principal components.

Formula:

For a given dataset with n observations and p features, the principal components are obtained through the eigendecomposition of the covariance matrix. The formula for PCA can be expressed as follows:

$$\begin{aligned} \text{Covariance Matrix:} \quad & \Sigma = \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X}) \\ \text{Eigenvalue Decomposition:} \quad & \Sigma = VDV^T \end{aligned}$$

Where:

X : Centered data matrix with dimensions $n \times p$
$\bar{X}$: Column-wise mean vector of X
Σ : Covariance matrix of X
V : Matrix of eigenvectors (principal components)
D : Diagonal matrix of eigenvalues

The principal components are then selected based on the eigenvalues, with the higher eigenvalues corresponding to the directions of maximum variance in the data.

In image processing, PCA can be applied to reduce the dimensionality of image data. Consider an image represented as a matrix I with dimensions $m \times n$, where each pixel corresponds to a feature. Applying PCA involves reshaping the image matrix into a vectorized form, performing the PCA transformation, and then reconstructing the image using a reduced set of principal components.

This process results in a compressed representation of the image, where the most important features are retained. The reconstructed image can be obtained by linearly combining the selected principal components.

I then applied PCA to reduce the dimensions from 4 to three resulting in the image of shape (512,512,3). Figure 3.5 shows the PCA converted image.

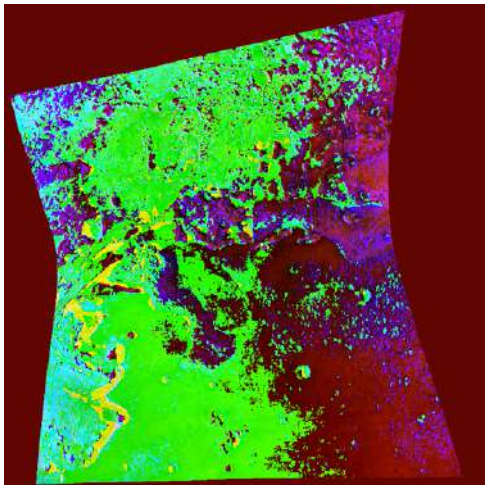

Figure 3.5: PCA converted image of frt000047a3_07_if166j

Applying rectangular filter

To experiment with the detection of carbonates I am also applying a rectangular filter to obtain the to include the channels PAN, NIR, and a separate channel containing wavelength between 1550 - 1990 as this wavelength range contains more information about the carbonates signatures even though these are not present in the CASSiS channels.

To apply the rectangular filter, I chose the wavelength values -

1. **Filter 1:** 780nm - 900nm
2. **Filter 2:** 860nm - 1150nm
3. **Filter 3:** 1550nm to 1990nm

Then I chose the wavelength corresponding to the nearest of these range averages from the data spectral cube values of CRISM. Figure 3.7 shows data frt000047a3_07_if166j after applying rectangular filters.

Image with carbonate signature channels

To generate the output images containing spectral signatures of carbonates we need to work with the refined spectral summary parameters or SR images. I chose the standard IR "FAL" false-color RGB image (R2529, R1506, R1080), they contain carbonate signatures:

1. **R1080:** 1.08-micron reflectance
2. **R1506:** 1.51-micron reflectance
3. **R2529:** 2.53-micron reflectance

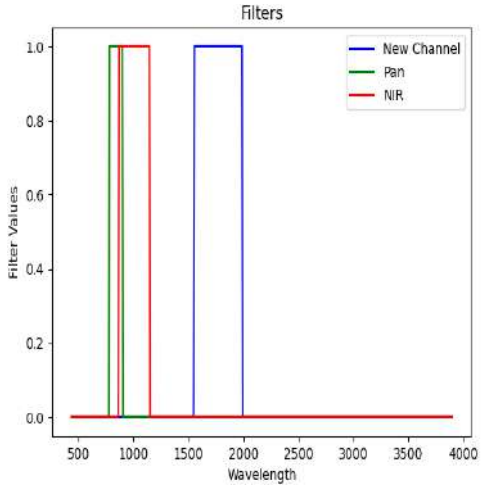

Figure 3.6: Rectangular filters

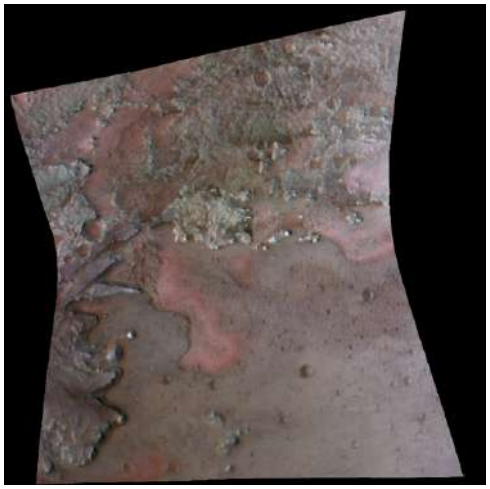

Figure 3.8: False color image of frt000047a3_07_if166j, visible green patches are carbonate signatures

Figure 23.8 shows an image made by the above 3 channels and there is a visible green color patch in the false-color image indicating the presence of carbonate signatures.

3.1.2 Data Augmentation

Data augmentation is a technique widely used in machine learning and computer vision to artificially increase the size of a dataset by applying various transformations to the existing data. The goal of data augmentation is to improve the generalization and robustness of machine learning models, particularly in scenarios where the available training data is limited.

Since the goal of this work is to accurately predict spectral bands, no changes to ground-truth CRISM pixel values such as brightness or gamma are considered.

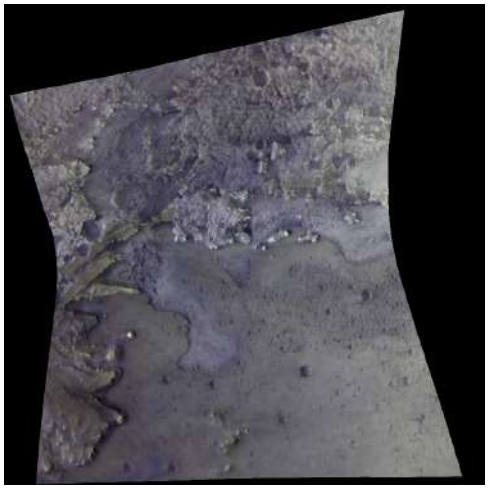

Figure 3.7: frt000047a3_07_if166j after applying rectangular filters

The augmentations examined in this work are rotation, flipping, and cropping. The images are cropped into 4 equal parts, and the degree of rotation is randomly chosen from the interval $[-45°, 45°]$. Additionally, the images are flipped vertically. The images are cropped into 4 equal parts

3.2 Baseline Model: Unet

The U-Net design plays a crucial role in our research when evaluating the effectiveness of diffusion models compared to classic image models. The U-Net architecture is widely recognized for its exceptional performance in picture segmentation tasks. It provides a strong foundation for conducting comparative analysis with diffusion models, specifically in the field of spectral signature prediction.

U-Net was initially developed for the purpose of biomedical picture segmentation. Its name is derived from its unique U-shaped structure, which enables the effective extraction of features and retention of spatial information. The overwhelming appreciation of this technology stems from its versatility and ability to effectively capture complicated patterns and characteristics present in photos. Stepcenkov et al. 2022 proved the efficiency of the method in spectral signature prediction tasks using the CRISM dataset, extending its use beyond medicinal applications [16]. Their research highlighted the adaptability of U-Net in other image modalities, confirming its significance beyond its initial field.

Furthermore, in addition to its architectural sophistication, the success of U-Net may be ascribed to the incorporation of skip connections, which facilitate the smooth integration of high and low-level elements, consequently augmenting the performance of the model. Furthermore, the widespread acceptance of this technology among researchers has facilitated the

development of a diverse range of pre-trained models and transfer learning techniques, hence enhancing its practicality and simplicity of use.

The unet architecture is referenced from Segmentation Models by PyTorch, the architecture is as follows -

- Encoder : Resnet34
- Weights: Imagenet
- Batch size: 8
- Data normalization method: image-wise
- Optimizer: Adam
- Learning rate: 10^{-3}
- Loss function: MSE
- Device: CUDA

The model is trained on different epochs and the best model according to the validation set is saved for evaluation.

3.3 InstructPix2Pix Diffusion Model

The diffusion model is a framework that represents the stochastic progression of a probability distribution as it changes over a period of time. Image denoising, image inpainting, and picture production have been among the several applications in which it has been employed.

The Hugging face diffusers library (https://github.com/huggingface/diffusers) is widely recognized as a prominent repository of advanced pre-trained diffusion models. These models are utilized for the purpose of generating molecular visuals, audio, and even three-dimensional structures. InstructPix2Pix suits our work and its architecture is comprised of several key components: the Variational Auto-Encoder (VAE) model (AutoencoderKL) for encoding and decoding images to and from latent representations, the text-encoder from CLIP for encoding the textual instructions, and a conditional U-Net for denoising the encoded image latent [28].

Diffusion models [32] utilize denoising autoencoders to generate data by estimating the score of data distribution. Latent diffusion enhances the effectiveness and excellence of diffusion models by functioning in the latent space of a pre-trained variational autoencoder equipped with an encoder and a decoder.

Data Preprocessing
I first apply the cassis filter on the CRISM MTRDR dataset available on the hub to obtain the

feature and target image from IF and SR products, and then store the pair together in separate folders. The prompt for the first part is fixed "Predict carbonates" and then create a dataset with the input image, output image, and prompt. The images are then converted to numpy

arrays, concatenated, and converted to tensors. The images are then converted to latent space using a pre-trained VAE from the stable diffusion model.

The prompt is converted to vector space using the pre-trained CLIP ViT-L/14 text encoder, the text encoder is frozen for the first step as the prompt is the same and thus it is only required to convert the text to vector encodings, however, it is unfrozen later and loaded with pre-trained weight to learn metadata of the dataset.

Forward Diffusion
Let the image be x, the noise is then added to the encoded latent from the encoder E is

$$z = E(x) \text{ over noisy latent } z_t$$

the noise levels are increased over timesteps t. Classifier-free guidance[3] is then implemented for two conditions which are input image (c_I) and text instruction (c_T)

Reverse Diffusion
The decoder then predicts the image from the noisy latent and tries to learn the data distribution by gradually denoising a normally distributed variable. The network ϵ_θ
learns the noise added to the latent from the input image and text prompt. The latent diffusion objective helps in aligning the latent space distributions of the generator and the discriminator, which leads to more stable and improved training, thus the latent diffusion objective to minimize [28] is:

$$L = \mathbb{E}_{E(x),E(c_I),c_T\in\sim N(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, E(c_I), c_T)\|_2^2\right]$$

Architecture

According to Wang [33], fine-tuning a model is more effective than training a model for image translation tasks, especially when there is a limited amount of paired training data.

The VAE and CLIP focused on encoding the image and instruction into latent space therefore I initialize the pre-trained instructPix2Pix weights for both VAE and text encoder and the Unet, which is in charge of denoising the noisy latent into the desired distribution of carbonate images is fine-tuned in the process with existing weights

For the later stage, I unfroze and fine-tuned the CLIP Text encoder as well to include metadata.

- Resolution: 512 pixels by 512 pixels
- Batch size: 8
- Optimizer: Adam

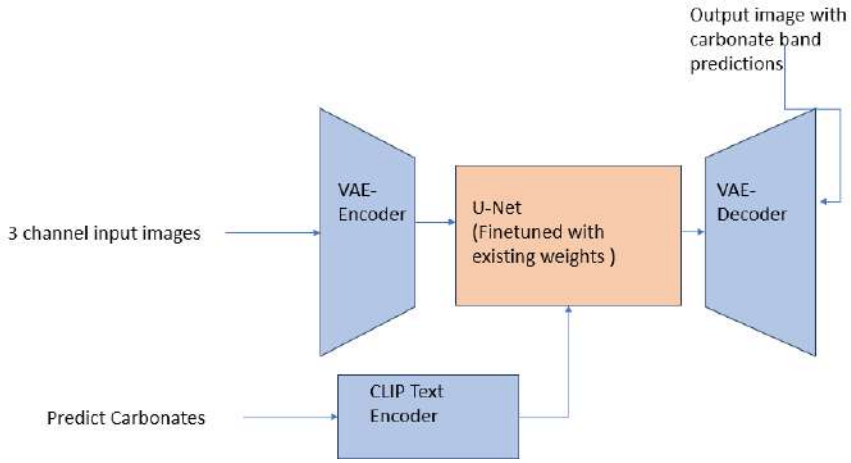

Figure 3.9: Model architecture of InstructPix2Pix optimized for the task, text encoder is later fed with metadata

- Learning rate: 5×10^{-5}
- Device: CUDA

4 Experimental Setup

4.1 Hardware and Software Requirements

The hardware and software environment for running and training the Diffusion model is a critical consideration to ensure efficient and effective performance. High-quality and high-memory GPU is required for stable diffusion.

Below are the key aspects of the hardware and software requirements -

1. **Hardware:**
    - CPU : Intel Xeon W-1390P with 16 cores and 128GB RAM.
    - GPU: NVIDIA RTX A5000 24564MiB
    - RAM: 64GB
2. **Software:**
    - **Operating System:** Linux - Ubuntu 22.04
    - **Programming Language:** Python3.9
    - **Dependencies:**
        a) **NumPy** is a fundamental package for scientific computing in Python used for numerical operations.
        b) **TorchVision** is a library consists of popular datasets, model architectures, and image transformations for computer vision. It consists of: Training recipes for object detection, image classification, instance segmentation, video classification, and semantic segmentation.
        c) **Pandas** is a powerful data manipulation library for Python used for handling structured data.
        d) **Diffusers** is the go-to library by HuggingFace for state-of-the-art pretrained diffusion models
        e) **Transformers** is a library for natural language understanding and generation, and it provides pre-trained models for various NLP tasks.
        f) **Cudatoolkit** The NVIDIA CUDA Toolkit provides a development environment for creating high-performance GPU-accelerated applications.

4.2 Performance Metrics

Evaluating the performance of image-to-image translation models is crucial for assessing the quality of generated images. Several metrics are commonly used for this purpose, each capturing different aspects of the translation process. Some commonly used metrics include Root Mean Squared Error (RMSE), Structural Similarity Index (SSIM), and perceptual metrics like Learned Perceptual Image Patch Similarity (LPIPS).

To measure the performance of the model using these metrics I predicted all images for test set and then calculated scores against each ground truth and prediction pair and averaged them to create final average scores against each metric for the model.

Details of Selected Metrics:

1. **Root Mean Squared Error (RMSE):**
   Root Mean Square Error (RMSE) is a commonly used metric in image processing and computer vision to quantify the difference between two images. It measures the average difference between corresponding pixel intensities of two images. The formula for RMSE is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(I_1(i) - I_2(i))^2}{n}}$$

   Where:

   - $I_1(i)$ is the intensity of the i-th pixel in the first image,
   - $I_2(i)$ is the intensity of the i-th pixel in the second image,
   - n is the total number of pixels in the images.

   RMSE provides a single numerical value representing the overall difference between the two images. A lower RMSE indicates a smaller difference between the images, implying higher similarity.

   RMSE can be used to compare similarity between two images by computing the RMSE between them. The lower the RMSE value, the more similar the images are. However, it's important to note that RMSE doesn't take into account perceptual differences that humans may consider significant even if the pixel-wise difference is small.

2. **Pearson Correlation Coefficient (PCC):**

   The Pearson correlation coefficient (PCC) is a measure of the linear correlation between two variables, in this case, two images. It quantifies the strength and direction of the linear relationship between the pixel intensities of the images. The formula for Pearson correlation coefficient is:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Where:

- x_i and y_i are the intensities of the i-th pixels in the two images, respectively.
- $\bar{x}$ and $\bar{y}$ are the mean intensities of the pixels in the two images, respectively.
- n is the total number of pixels in the images.

The Pearson correlation coefficient ranges from -1 to 1 where -1 implies perfect negative and 1 implies perfect positive relation.

PCC can be used to measure the similarity between two images. A higher absolute value of PCC indicates a stronger linear relationship between the images, suggesting higher similarity. However, PCC only captures linear relationships and may not capture non-linear dependencies between pixel intensities.

3. **Cosine Similarity:**
Cosine similarity is a measure used to determine how similar two vectors are, often used in the context of comparing documents or, in this case, images. It calculates the cosine of the angle between two vectors, where the vectors represent the pixel intensities of the images. The formula for cosine similarity is:

$$\text{cosine\_similarity} = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$

Where:

- x_i and y_i are the intensities of the i-th pixels in the two images, respectively.
- n is the total number of pixels in the images.

Cosine similarity ranges from -1 to 1 where -1 indicates that images are completely different and 1 indicates images are perfectly similar.

4. **Learned Perceptual Image Patch Similarity (LPIPS):**
Learned Perceptual Image Patch Similarity (LPIPS), proposed by Zhang et al., 2021 [34] is a deep neural network-based perceptual similarity metric. It learns from labeled data and predicts similarity between images based on human perception.

   LPIPS is based on a deep convolutional neural network (CNN) trained on large-scale image datasets. The network learns to extract features that are relevant for human perception, such as texture, color, and structure. The LPIPS metric computes the Euclidean distance between the feature representations of corresponding patches in the images so the lower the LPIPS score the more similar the images are.

   LPIPS has been shown to outperform traditional metrics in capturing perceptual differences between images, aligning more closely with human perceptual judgments. It is

widely used in tasks such as image quality assessment, image retrieval, and image synthesis.

5. **Structural Similarity Index:**
   Structural Similarity Index (SSIM) is a metric used to quantify the similarity between two images. Unlike simple pixel-based metrics like Mean Squared Error (MSE), SSIM takes into account the structural information and luminance of the images, which are more closely related to human perception.

   The formula for SSIM is:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

   Where:

   - x and y are the two images being compared.
   - μ_x and μ_y are the means of x and y, respectively.
   - σ_x and σ_y are the standard deviations of x and y, respectively.
   - σ_{xy} is the covariance of x and y.
   - c_1 and c_2 are small constants added to avoid division by zero.

   SSIM ranges from -1 to 1, where 1 indicates perfect similarity between the images.

   SSIM is widely used in image quality assessment, it has been shown to better correlate with human perception of image quality compared to traditional metrics like PSNR (Peak Signal-to-Noise Ratio).

These metrics provide quantitative measures for different aspects of image-to-image translation model performance, including pixel-level accuracy, structural similarity, and perceptual similarity.

5 Experiments and Results

In this section, methods and tools that have been presented and created are applied, and their results are compared. The number of epochs and image-wise normalization are chosen based on prior experiments, while a batch size of 8 guarantees that each model fits into memory. The patch size and learning rate are common default values, whereas MSE and Adam are robust first choices.

5.1 Unet model

5.1.1 PAN,NIR,RED Dataset

The unet model is trained properly and the loss per epoch as shown in Figure 5.1 displays the model is able to learn and generalize with loss decreasing per epoch.

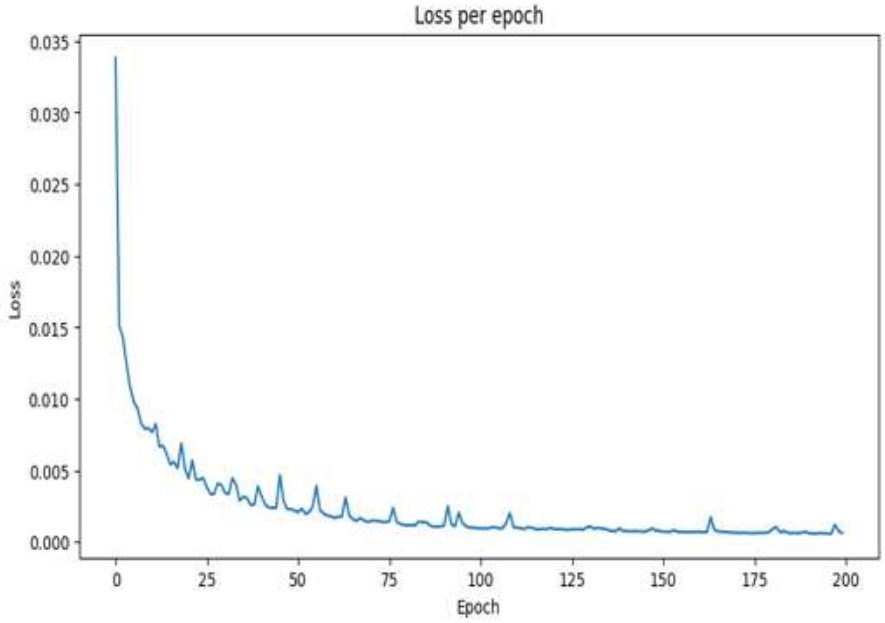

Figure 5.1: Unet training loss per epochs

As expected the Unet model performed well however it is only able to capture the structure of the input image and is unable to capture the carbonate segments. I have used various metrics to quantify the difference between the predicted image and the ground truth image which

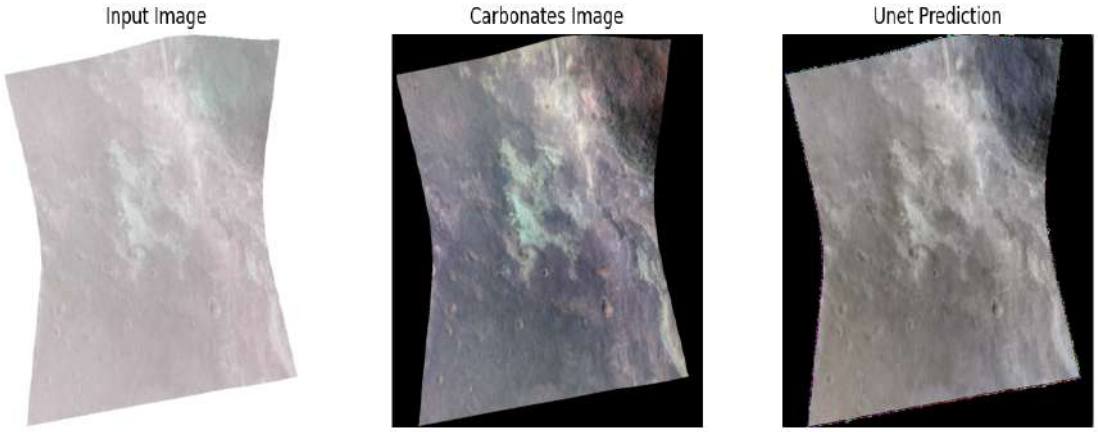

Figure 5.2: Input Image, Ground truth (image with carbonate signatures) and unet predicted image shows missing carbonate signature in unet predicted image

were calculated entirely on the test set. The root mean square error give me an overview

of pixel-wise accuracy between the two images, the Pearson Correlation Coefficient (PCC) measures the linear relationship between the images while the cosine similarity measures the similarity in orientation between the images. The Structural Similarity Index (SSIM) is used to measure the structural similarity between the images while the Learned Perceptual Image Patch Similarity (LPIPS) is trained using a neural network and is learned from human judgments of image similarity.

Metric	Score
Average RMSE	6.967
Average PCC	0.961
Average Cosine Similarity	0.986
Average SSIM	0.915
Average LPIPS	0.128

Table 5.1: Performance Metrics for Unet model

5.1.2 Data Skewness

Table 5.1 shows the score of the above metrics for the unet model, at first glance it appears that the model is performing well and it's producing images similar to the ground truth however upon closer inspection of the predictions and dataset I came across skewness in the dataset.

The carbonates are not abundant on Mars and it has an even lower presence in the data

captured by CRISM therefore there were a lot of images where carbonates were not present, we have already seen that unet was not able to capture carbonates however it was able to capture the structure therefore if there are no carbonates present in the image the unet is able to produce images very similar to the ground truth thus improving the scores.

Figure 5.3: Input Image, Ground truth (image with carbonate signatures) and unet predicted image shows if carbonates are missing unet is producing images very similar to ground truth

To overcome this data skewness I went through the images separately and created a new smaller test set with a balanced mix of images containing both carbonate and non-carbonate images.

I then measured each metric again on the smaller test dataset and Table 5.2 displays the metrics and their respective scores. Here we can clearly see that all the metrics point to a lower performance than the previous score, the RMSE, and LPIPS show major variations as they are measuring pixel-wise differences while structurally the images are still the same so PCC, and cosine similarity and SSIM shows minor changes.

Metric	Score
Average RMSE	7.466
Average PCC	0.931
Average Cosine Similarity	0.975
Average SSIM	0.882
Average LPIPS	0.169

Table 5.2: Performance Metrics of Unet model on smaller balanced dataset

I have used this smaller balanced test dataset to calculate further metrics.

5.1.3 PCA Dataset

Unet performed badly on the PCA dataset, The unet model is designed to excel in image segmentation tasks but it encountered challenges when applied to the PCA dataset derived from reducing 4 channels (PAN, NIR, RED, and BLU) to 3.

Figure 5.4 shows that unet is not able to capture the structure as well as the pixels correctly.

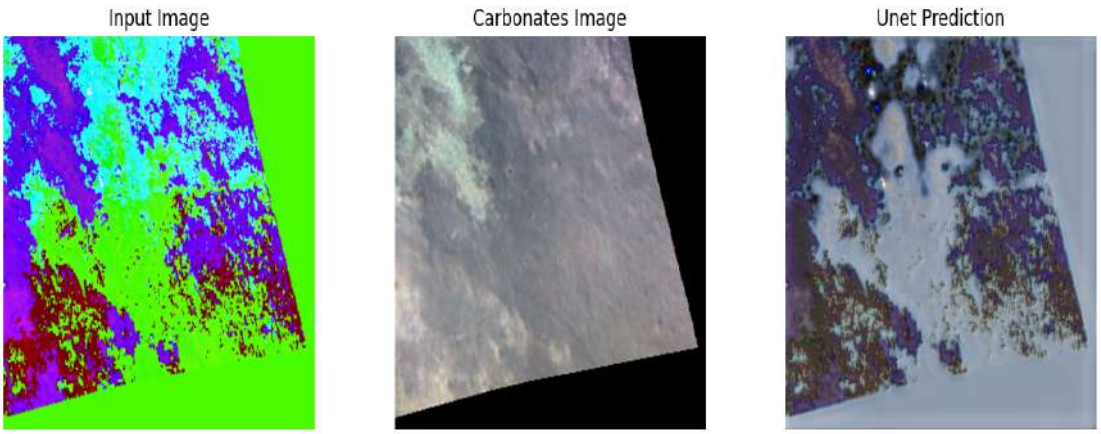

Figure 5.4: Unet not producing good results on PCA dataset

Possible explanations for poor performance

1. **Data Mismatch:**
   - The inherent characteristics of PCA-transformed data may not align with the assumptions or features that U-Net is adept at capturing. The model might struggle to effectively interpret and reconstruct the data in the transformed space.
2. **Complex Spectral Information:**
   - CRISM's spectral data is known for its complexity, and the transformation through PCA may not have retained critical spectral features essential for accurate segmentation. U-Net's architecture may not be well-suited for capturing nuanced spectral information.
3. **Unet might not suited for non-RGB images:**
   - PCA transformation to reduce four channels (blue, pan, nir, and red) to three might impact the way the information is represented in the dataset. PCA aims to retain

the most significant features of the original data, but it might not fully preserve the spatial relationships present in RGB images.

Table 5.3 quantifies the poor performance of unet on PCA dataset

Metric	Score
Average RMSE	10.421
Average PCC	0.135
Average Cosine Similarity	0.755
Average SSIM	0.262
Average LPIPS	0.422

Table 5.3: Performance Metrics of Unet on PCA dataset

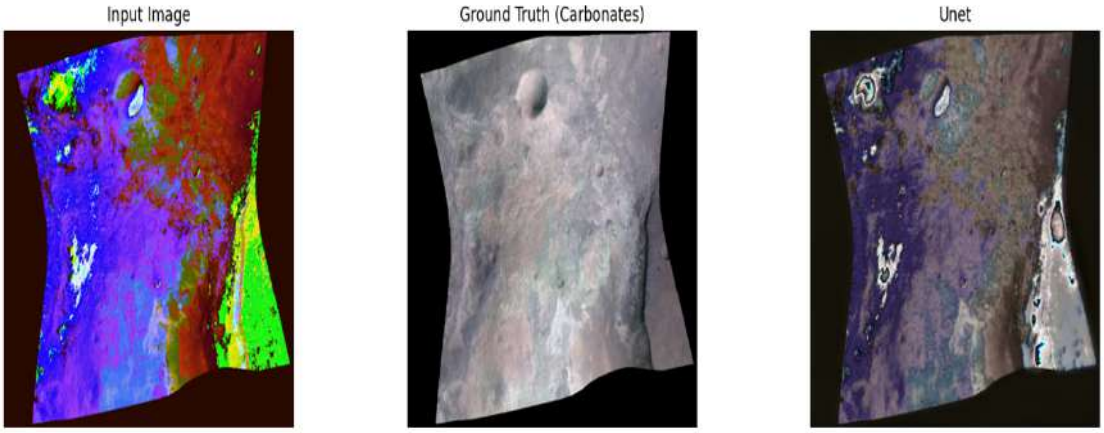

Figure 5.5: Unet not producing good results on PCA dataset

5.2 Stable Diffusion

I fine-tuned my model with existing weights os stable diffusion since the dataset provided was small and the stable diffusion model has been trained on a much larger dataset with high GPU specs therefore fine tuning the model was much more effective than training from scratch.

5.2.1 PAN, NIR, RED Dataset

The model performs well on the PNR dataset, as is shown by the metrics in Table 5.4. Almost all the metrics are better than Unet except structural similarity which is expected as Unet is better at capturing the segments and structure of an image.

Metric	Score
Average RMSE	7.097
Average PCC	0.979
Average Cosine Similarity	0.980
Average SSIM	0.826
Average LPIPS	0.071

Table 5.4: Performance Metrics of diffusion model on PNR dataset

Figure 5.6: Stable Diffusion predicting carbonates

Figure 5.6 shows that stable diffusion is able to capture the structure as well as predict carbonates in specific pixels, however, I can notice that the model is a bit biased and is predicting carbonates in areas where it isn't present while Figure 5.7 shows that stable diffusion is not predicting enough carbonates that are present.

5.2.2 Detecting Bias and adding metadata

Upon closer inspection of the predictions generated by the Stable Diffusion model, notable instances were identified where carbonates were predicted even in the absence of ground truth representation. Additionally, there were cases of both over-prediction and under-prediction of carbonate quantities compared to the actual ground truth. The existence of such biases prompts a deeper investigation into potential explanations.

Possible Reasons for Bias:

1. **Carbonates Quantity on Mars:**
    - Mars is not rich in carbonates, and the CRISM dataset covers only a limited portion of its surface. The scarcity of carbonates in the training dataset may lead to biased

Figure 5.7: Stable Diffusion predicting carbonates

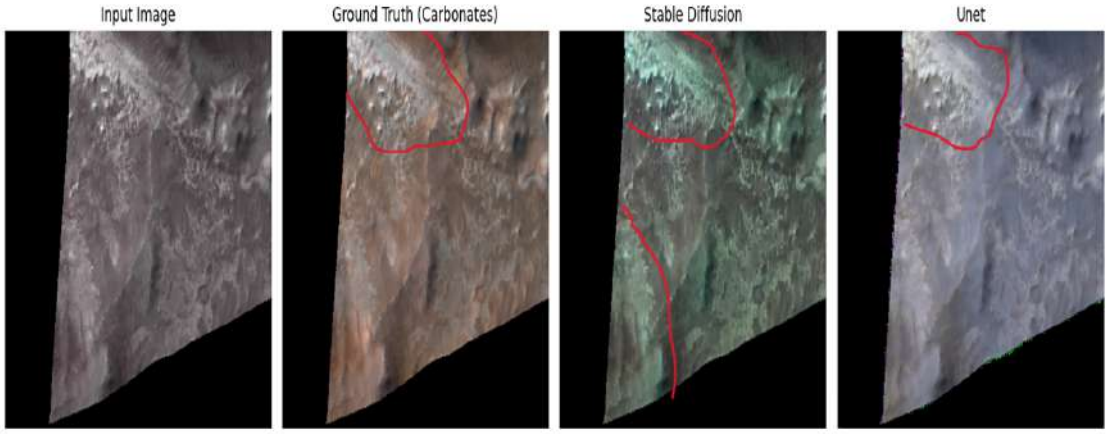

Figure 5.8: Stable Diffusion predicting carbonates even if they are not present

predictions, as the model might not have learned the nuances associated with their limited presence.

2. **Location-Dependent Distribution:**
   - Carbonates on Mars exhibit a location-dependent distribution. Specific regions, such as the Jazero Crater, may have a higher abundance of carbonates compared to others, like the Coprates region. Factors such as atmospheric conditions, mineral composition, and geological history contribute to these variations.
3. **Spatial Variability:**
   - The model may struggle with the spatial variability of carbonate concentrations, especially in regions where carbonates are sparsely distributed. The lack of consistent patterns across the dataset can contribute to prediction inconsistencies.

To tackle bias in the predictions, I took a strategic approach by refining the dataset and

incorporating additional metadata. My primary focus was on creating a more balanced representation of both carbonate and non-carbonate areas within the data.

Dataset Refinement: I thoroughly examined the dataset and created a more balanced version. This involved intentionally including areas with and without carbonates, ensuring a more representative and diverse training set for the model.

Metadata Integration: To enhance the model's predictive capabilities, I selected three key metadata attributes—Longitude, Latitude, and Location Name. These attributes provide valuable location-specific information that the model can learn from, aiding in predicting the presence and quantity of carbonates based on geographic characteristics.

Updated Prompt: I refined the model's prompt to 'Predict carbonates if present,' aligning more closely with the refined dataset and metadata features. This adjustment helps guide the model to focus on the specific task of carbonate prediction.

Text Encoder Modification: To accommodate the new metadata, I unfroze the CLIP text encoder and created vector embeddings for the Longitude, Latitude, and Location Names, enriching the dataset with valuable location-related information for the model to learn during training.

Purpose of Metadata:

- **Longitude and Latitude:** Providing precise geographical coordinates allows the model to learn spatial patterns and correlations related to carbonate presence.
- **Location Name:** Offering contextual information about the specific region helps the model understand the unique geological features associated with different areas.

Impact on Model Training: The integration of these metadata elements contributes to a more holistic learning experience for the model. By incorporating location-related details, the model gains insights into the spatial context of carbonate occurrences, potentially reducing biases associated with insufficient training data.

The results were much better after implementing the above steps, the stable diffusion model was now no longer predicting carbonate if they were not present.

Figure 5.9 shows that stable diffusion does not predict carbonates if it's not present on the same dataset, it removes bias on additional datasets as well as shown in Figure 5.10.

Table 5.5 also show good performance with metadata.

5.2.3 Performance of Stable Diffusion on PCA Dataset

In contrast to its success on other datasets, Stable Diffusion exhibited suboptimal performance when applied to the Principal Component Analysis (PCA) dataset. While the model demonstrated proficiency in predicting the presence of carbonates, its ability to pinpoint specific pix-

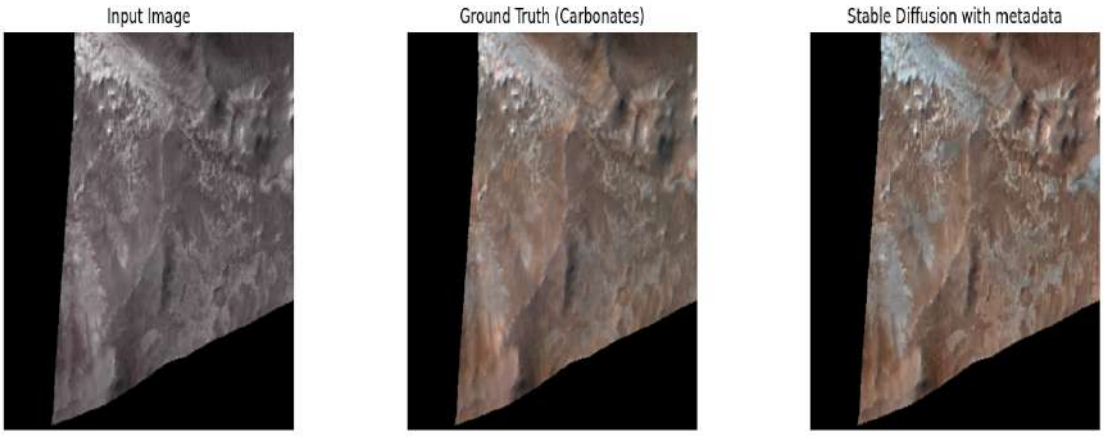

Figure 5.9: Stable Diffusion not predicting carbonates if they are not present

Metric	Score
Average RMSE	6.743
Average PCC	0.974
Average Cosine Similarity	0.983
Average SSIM	0.833
Average LPIPS	0.067

Table 5.5: Performance Metrics of diffusion model on PNR dataset with metadata

els was notably lacking, resulting in predictions resembling generalized areas rather than distinct locations.

Challenges and Observations:

1. **Location ambiguity:**
   - Stable Diffusion struggled to precisely locate specific pixels associated with carbonate presence on the PCA-transformed dataset. The predictions exhibited a more diffused and blob-like pattern, indicating a challenge in spatial localization. Figure 5.11 shows stable diffusion not able to pinpoint pixels
2. **Cannot locate carbonates:**
   - The model can sometimes not even predict the carbonates present in the predictions even with metadata. Figure 5.12 shows one such case

The metric scores also indicate a poor performance with all scores worse than the model trained on the PNR dataset.

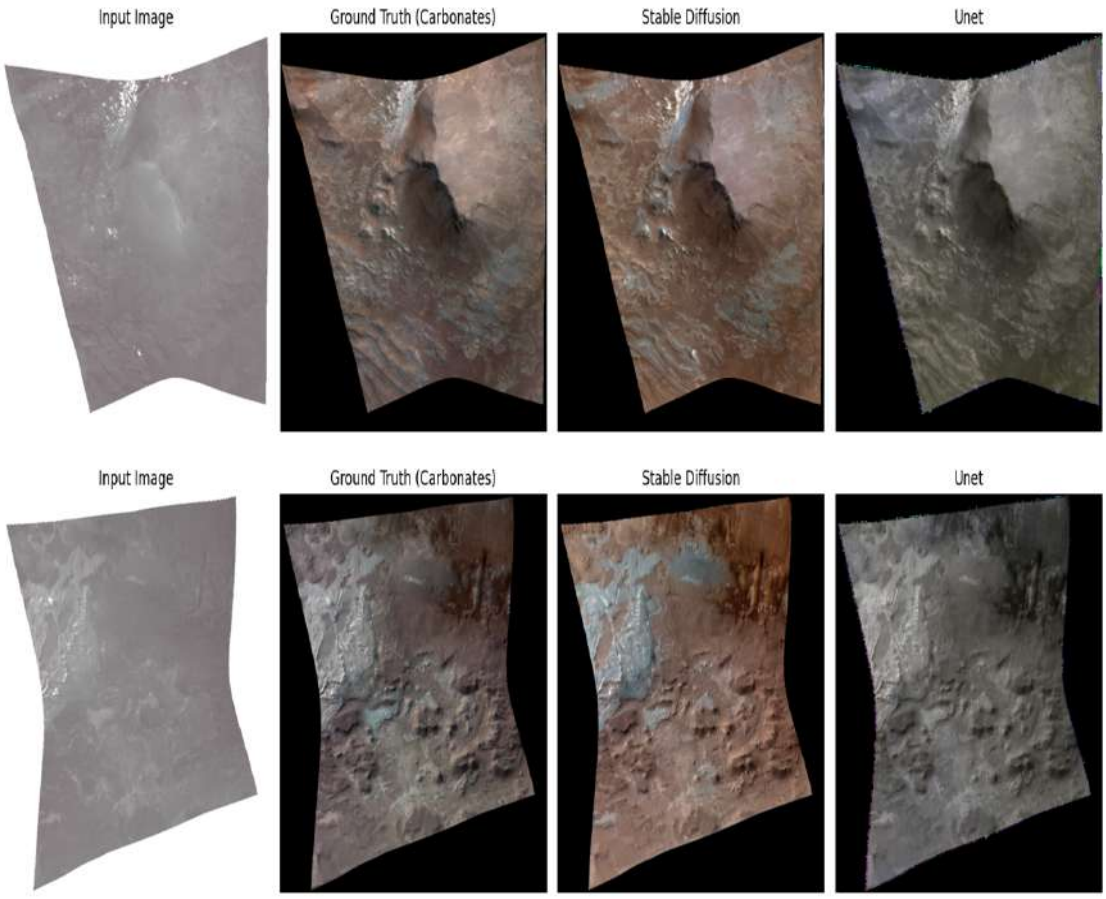

Figure 5.10: Stable diffusion with reduced bias

Possible Reasons for Poor Performance with PCA:

1. **Dimensionality Reduction Impact:**
   - PCA transforms the original 4-dimensional image (pan, nir, red, and blu) into a 3-dimensional representation, possibly resulting in a loss of critical spectral information. Stable Diffusion, designed for higher-dimensional datasets, might struggle to recover intricate details after the dimensionality reduction process.
2. **Loss of Spatial Information:**
   - The reduction of dimensions through PCA may lead to a loss of spatial information crucial for accurate localization. Stable Diffusion relies on detailed spatial relationships in higher-dimensional spaces, and the compression introduced by PCA could hinder the model's ability to discern specific pixel locations.

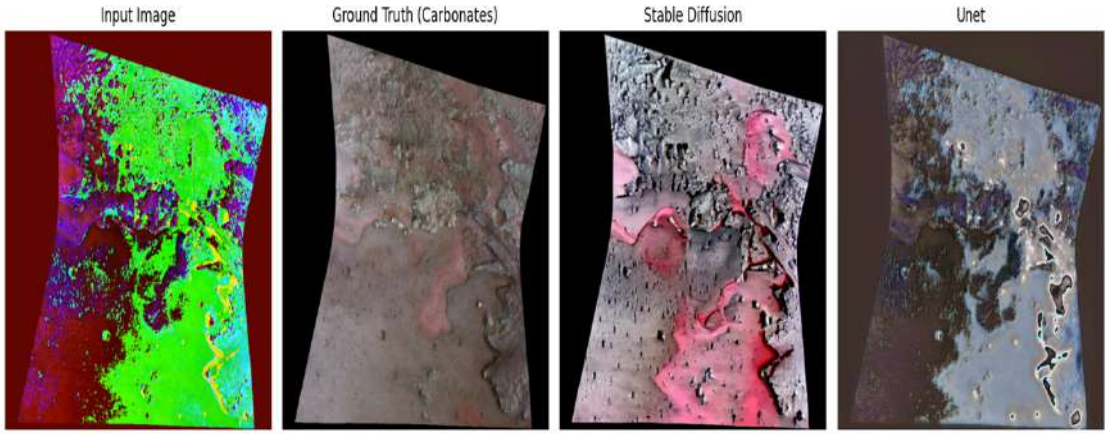

Figure 5.11: Stable Diffusion cannot pinpoint the carbonates pixels

Metric	Score
Average RMSE	8.731
Average PCC	0.880
Average Cosine Similarity	0.952
Average SSIM	0.576
Average LPIPS	0.168

Table 5.6: Performance Metrics for stable diffusion on PCA dataset

3. **Complexity Mismatch:**
    - Stable Diffusion might face challenges in adapting to the altered complexity of the PCA-transformed data. The model's architecture, optimized for certain spectral and spatial intricacies, may not align with the reduced complexity of the transformed dataset.

In conclusion, the challenges encountered by Stable Diffusion on the PCA dataset highlight the nuanced nature of model-data interactions.

5.3 Comparison of model performance

Upon rigorous experimentation, it became evident that conventional Convolutional Neural Network (CNN) models, such as Unet, faced challenges in capturing the intricate details of spectral channels and struggled to reproduce accurate results in scenarios where no explicit relationship existed between these channels or pixels. In contrast, models based on diffusion, particularly the Stable Diffusion model fine-tuned on the extensive LIAON-2B dataset using high-scale GPU resources, showcased superior performance in predicting carbonates on specific pixels.

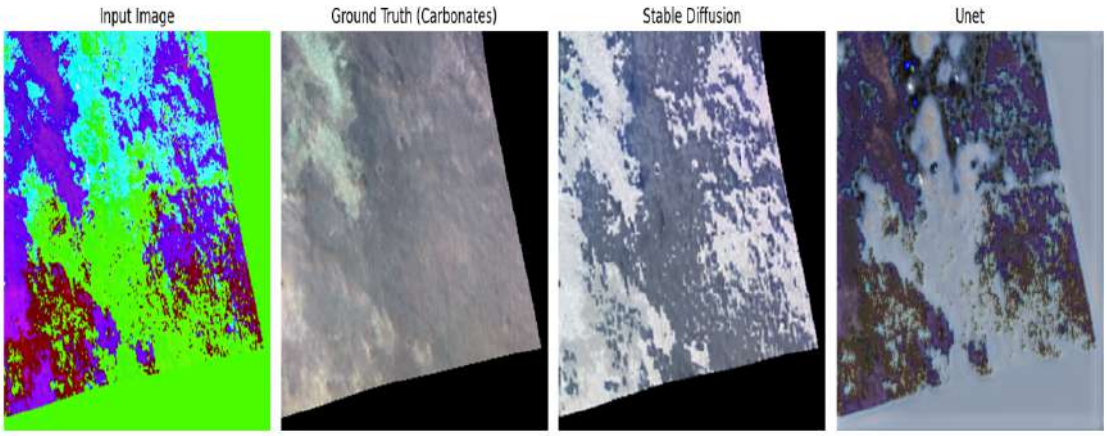

Figure 5.12: Stable Diffusion cannot predict carbonates

Performance Comparison: Unet vs. Stable Diffusion

- **Unet Limitations:** The Unet model, a representative of CNN architectures, demonstrated limitations in capturing nuanced spectral details and struggled in scenarios with no inherent relationships between channels or pixels. Its inability to reproduce accurate results in such contexts underscored the need for alternative approaches.
- **Stable Diffusion Superiority:** Stable Diffusion, having undergone fine-tuning on the expansive LIAON-2B dataset with the aid of high-scale GPU resources, emerged as a standout performer. This model excelled in predicting carbonates on specific pixels, showcasing its adaptability and robustness in handling the complexities of the task.

Dataset-Specific Performance: PNR vs. PCA

- **PNR Dataset:** Both Unet and Stable Diffusion models demonstrated better performance on the Pan, NIR, and Red (PNR) datasets compared to other datasets. However, Stable Diffusion exhibited a notably superior performance, surpassing Unet in terms of both quantitative metrics and qualitative examination of predicted images.
- **PCA Dataset:** While both models struggled on the Principal Component Analysis (PCA) dataset, Stable Diffusion outperformed Unet. This superiority was evident in the metrics as well as the manual inspection of predicted images. Stable Diffusion showcased its adaptability to the challenges introduced by the dimensionality reduction inherent in PCA.

The following table summarizes the quantitative comparison metrics of both Unet and Stable Diffusion models across different datasets:

Metrics	Unet Score	Unet PCA Score	Sd PNR Scores	SD PCA Scores
Average RMSE	7.466	10.421	**6.743**	8.731
Average PCC	0.931	0.135	**0.979**	0.88
Average Cosine Similarity	0.975	0.755	**0.983**	0.952
Average SSIM	**0.882**	0.262	0.833	0.576
Average LPIPS	0.169	0.422	**0.067**	0.168

Table 5.7: Comparison of Performance Metrics

5.4 Performance of Stable Diffusion on Rectangular Filter Dataset

I have included this experiment in a separate section as this section presents an experimental analysis conducted outside the scope of CASSiS filters, aiming to evaluate the potential enhancement in prediction accuracy through selective channel utilization based on mineralogical characteristics. Notably, this exploration underscores the significance of wavelength selection and data curation in optimizing predictions, particularly for carbonate signatures.

The model exhibited superior performance, indicating the efficacy of wavelength selection and data preprocessing in discerning carbonate signatures from spectral data. Two exemplary cases are illustrated in Figures 5.13 and 5.14, demonstrating the Stable Diffusion model's capability in correctly predicting carbonate signatures, albeit with occasional oversights.

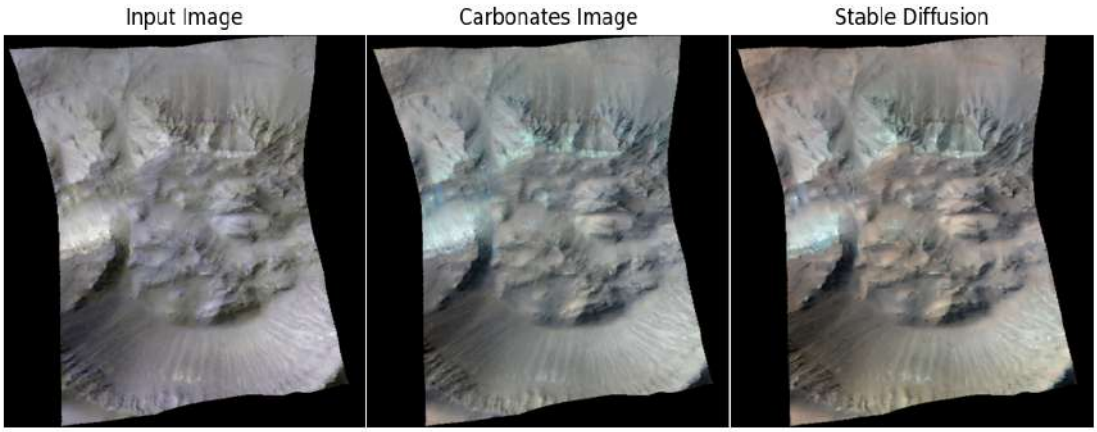

Figure 5.13: Stable Diffusion correctly predicting carbonate signatures.

A comprehensive evaluation of the model's performance across various metrics is presented in Table 5.8. The results indicate notable improvements across most evaluation criteria, underscoring the pivotal role of wavelength selection in mineral detection.

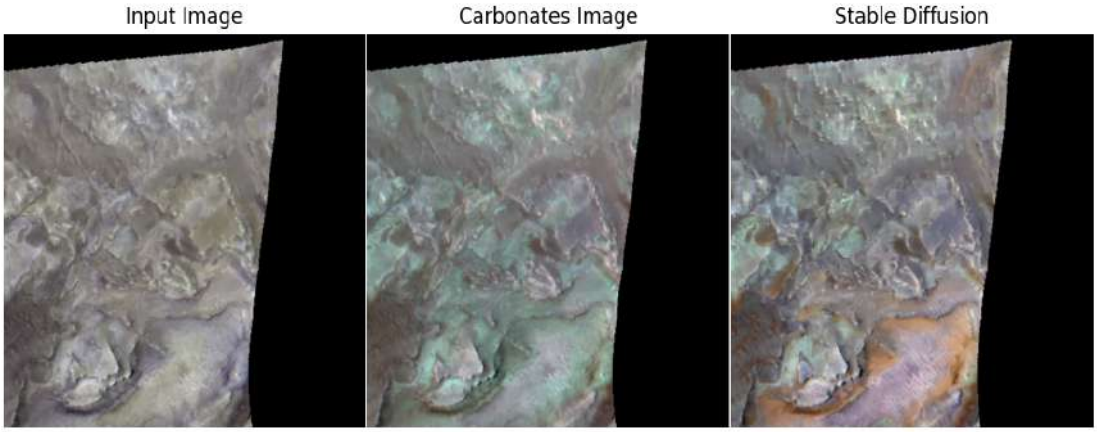

Figure 5.14: Stable Diffusion performing well but occasionally missing carbonate pixels.

Metric	Score
Average RMSE	6.381
Average PCC	0.984
Average Cosine Similarity	0.994
Average SSIM	0.851
Average LPIPS	0.039

Table 5.8: Performance Metrics for stable diffusion on rectangular filter dataset

These findings underscore the potential of spectral wavelength selection as a means to enhance mineral detection accuracy, as evidenced by improved performance across a spectrum of evaluation metrics.

6 Conclusion and Future Work

6.1 Conclusion

The completion of this thesis marks a thorough exploration into the domain of predicting carbonates on Martian surfaces using advanced remote sensing techniques. The journey unfolded through several crucial steps, each contributing to a deeper understanding of the intricacies involved in this challenging task. Traditional neural network models produced inferior results and were not able to predict the spectral channel outside of the visible range, especially near the NIR band while the diffusion model with metadata was able to predict the spectral channel outside of the visible spectrum and even a different channel with no dependencies. They are more accurate if the channel provided contains more information about carbonates as the wavelengths between 1550nm - 1990nm are more related to the carbonates.

I can highlight some important conclusions from each step:

1. **Dataset Creation:** I laid the foundation of my research on the meticulous creation of a dataset using spectral channels, particularly from the Compact Reconnaissance Imaging Spectrometer for Mars (CRISM). This dataset served as the canvas for subsequent model training and evaluation, capturing the nuanced spectral signatures crucial for carbonate prediction.

2. **UNet Model Implementation:** I employed the conventional Convolutional Neural Network (CNN) architecture, UNet, as a benchmark for comparison. Despite its widely acknowledged capabilities, UNet revealed limitations in capturing the intricate details of spectral channels and struggled in scenarios devoid of explicit relationships between channels or pixels.

3. **Exploration of Diffusion Models:** Motivated by the limitations encountered with CNNs, I extended my exploration to diffusion models. Extensive research into the capabilities of diffusion models, especially Stable Diffusion, revealed their potential in capturing subtle spectral nuances and predicting specific pixels, offering a promising alternative to traditional CNN architectures.

4. **Model Fine-Tuning with instructPix2Pix:** Fine-tuning Stable Diffusion became the focal point of my efforts, leveraging the instructPix2Pix method. This involved updating the architecture to align with the intricacies of carbonate prediction on Martian surfaces. The model was trained on the LIAON-2B dataset, demonstrating adaptability and robustness, particularly in scenarios with challenging spectral relationships.

5. **Comprehensive Evaluation:** My thesis journey culminated in a comprehensive evaluation, comparing the performance of UNet and fine-tuned Stable Diffusion on different datasets—Pan, NIR, Red (PNR) and Principal Component Analysis (PCA). Metrics, qualitative analysis, and manual inspection collectively underscored the superior performance of Stable Diffusion, especially in challenging scenarios presented by the PCA dataset.

6. **Key Findings:** The findings not only emphasized the limitations of traditional CNNs in nuanced remote sensing tasks but also highlighted the potential of diffusion models, specifically Stable Diffusion, in capturing intricate spectral details and predicting specific pixels on Martian surfaces.

In conclusion, this thesis contributes to the evolving landscape of remote sensing methodologies. The journey from dataset creation to model exploration and fine-tuning illuminates the potential of diffusion models, laying the groundwork for future advancements in the nuanced realm of planetary surface analysis. The diffusion models along with metadata information are successful in predicting channels out of visible channels, whereas traditional neural network models have provided inferior results.

The results obtained not only address current challenges but also beckon further exploration, inviting the scientific community to delve deeper into the capabilities of advanced models for remote sensing tasks.

6.2 Future Work

As I reflect on the outcomes of my research, several avenues for future exploration and enhancement emerge. The following areas represent potential directions for future work in the realm of predicting carbonates on Martian surfaces using advanced remote sensing techniques:

1. **Increase Dataset Coverage:** Expanding the dataset coverage presents a significant opportunity for further improvement. The CRISM dataset is vast, with individual data files ranging from 1-2GB. My research utilized a subset of this data due to computational constraints. Future work could involve a more comprehensive utilization of the CRISM dataset, potentially incorporating additional spectral channels to enhance model robustness.

2. **Architecture Modification for Stable Diffusion:** Adapting the Stable Diffusion architecture to accommodate 4-band images is a compelling avenue for exploration. This would necessitate retraining the Variational Autoencoder (VAE) from scratch to effectively capture the nuances of the additional spectral information. This modification aims to improve the model's capacity to discern subtle variations in the spectral signatures associated with carbonates.

3. **Incorporate Multi-Sensor Data Fusion:** Combining data from multiple remote sensing instruments beyond CRISM could enhance the predictive capabilities of the model. Fusion with data from instruments like the Mars Orbiter Laser Altimeter (MOLA) or the

Mars Color Imager (MARCI) could provide complementary information, offering a more holistic understanding of the Martian surface.

4. **Investigate Transfer Learning Strategies:** Exploring transfer learning strategies could leverage knowledge gained from related tasks or datasets. Pre-training models on analogous terrestrial mineralogical datasets or tasks before fine-tuning for Martian carbonates might facilitate faster convergence and improved performance.

These suggestions represent potential avenues for future research, each offering unique opportunities to advance the understanding and application of remote sensing techniques in the context of Martian mineralogy prediction.

Bibliography

[1] R. W. Zurek and S. E. Smrekar, "An overview of the mars reconnaissance orbiter (mro) science mission," *Journal of Geophysical Research: Planets*, vol. 112, p. E05S01, 2007.

[2] S. Murchie, R. Arvidson, P. Bedini, K. Beisser, J. Bibring, J. Bishop, J. Boldt, P. Cavender, T. Choo, R. Clancy *et al.*, "Compact reconnaissance imaging spectrometer for mars (crism) on mars reconnaissance orbiter (mro)," *Journal of Geophysical Research: Planets*, vol. 112, p. E05S03, 2007.

[3] N. Thomas, G. Cremonese, R. Ziethe, M. Gerber, M. Brändli, G. Bruno, M. Erismann, L. Gambicorti, T. Gerber, K. Ghose, M. Gruber, P. Gubler, H. Mischler, J. Jost, D. Piazza, A. Pommerol, M. Rieder, V. Roloff, A. Servonet, W. Trottmann, T. Uthaicharoenpong, C. Zimmermann, D. Vernani, M. Johnson, and J. J. Wray, "The colour and stereo surface imaging system (cassis) for the exomars trace gas orbiter," *Space Science Reviews*, vol. 212, no. 3-4, pp. 1897–1944, 2017.

[4] S. Pelkey, J. Mustard, S. Murchie, R. Clancy, M. Wolff, M. Smith, R. Milliken, J. Bibring, A. Gendrin, F. Poulet *et al.*, "Crism multispectral summary products: Parameterizing mineral diversity on mars from reflectance," *Journal of Geophysical Research: Planets*, vol. 112, p. E05S03, 2007.

[5] P. K. Sethy, C. Pandey, Y. K. Sahu, and S. K. Behera, "Hyperspectral imagery applications for precision agriculture - a systemic survey," *Multimedia Tools and Applications*, vol. 81, pp. 3005–3038, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11042-021-11729-8

[6] *Development of Hyperspectral Imaging for Mineral Exploration.* Society of Exploration Geophysicists. [Online]. Available: https://pubs.geoscienceworld.org/segweb/books/book/1381/chapter/107029963/Development-of-Hyperspectral-Imaging-for-Mineral

[7] D. Koßmann, T. Wilhelm, and G. Fink, "Towards tackling multi-label imbalances in remote sensing imagery," in *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, January 2021, pp. 5782–5789.

[8] U. Gewali, S. Monteiro, and E. Saber, "Machine learning based hyperspectral image analysis: A survey," *arXiv preprint arXiv:1802.08701*, 2018.

[9] A. R. Azari, J. B. Biersteker, R. M. Dewey, G. Doran, E. J. Forsberg, C. D. K. Harris, H. R. Kerner, K. A. Skinner, A. W. Smith *et al.*, "Integrating machine learning for planetary science: Perspectives for the next decade," *arXiv preprint arXiv:2007.15129*, 2020. [Online]. Available: https://arxiv.org/abs/2007.15129

[10] C. Nixon, Z. Yahn, E. Duncan, I. Neidel, A. Mills, B. Seignovert *et al.*, "Feature extraction and classification from planetary science datasets enabled by machine learning," *arXiv preprint arXiv:2310.17681*, 2023. [Online]. Available: https://arxiv.org/abs/2310.17681

[11] M. Dundar, B. L. Ehlmann, and E. K. Leask, "Machine-learning-driven new geologic discoveries at mars rover landing sites: Jezero and ne syrtis," *arXiv preprint arXiv:1909.02387*, 2019.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 1125–1134.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 770–778.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 4700–4708.

[16] S. Stepcenkov, T. Wilhelm, and C. Wöhler, "Learning the link between albedo and reflectance: Machine learning-based prediction of hyperspectral bands from ctx images," *Remote Sensing*, vol. 14, p. 3457, 2022.

[17] N. Zabari, A. Azulay, A. Gorkor, T. Halperin, and O. Fried, "Diffusing colors: Image colorization with text guided diffusion," *arXiv preprint arXiv:2201.00590*, 2022.

[18] (2024) Crism: Exploring the geology of mars. [Online]. Available: https://science.nasa.gov/wp-content/uploads/2024/03/31680_mro-crism-exploring-geology-of-mars.pdf?emrc=662e2f5928528

[19] "Mineralogical characterization of martian jezero crater from mro crism," 2014. [Online]. Available: https://isprs-archives.copernicus.org/articles/XL-4/117/2014/isprsarchives-XL-4-117-2014.pdf

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv preprint arXiv:1505.04597*, 2015.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[22] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arXiv:2106.06567*, 2021.

[23] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[24] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," *arXiv preprint arXiv:2106.06567*, 2021.

[25] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019. [Online]. Available: http://dx.doi.org/10.1561/2200000056

[26] Steins, "Stable diffusion clearly explained!" https://medium.com/@steinsfu/stable-diffusion-clearly-explained-ed008044e07e, 2023.

[27] OpenAI, "Clip: Connecting text and images," *OpenAI Research*, 2021. [Online]. Available: https://openai.com/research/clip/

[28] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," *arXiv preprint arXiv:2211.09800*, 2022.

[29] C. E. Viviano-Beck, F. P. Seelos, S. L. Murchie, E. G. Kahn, K. D. Seelos, H. W. Taylor, K. Taylor, B. L. Ehlmann, S. M. Wiseman, J. F. Mustard, and M. F. Morgan, "Revised crism spectral parameters and summary products based on the currently detected mineral diversity on mars," *Journal of Geophysical Research: Planets*, vol. 119, pp. 1403–1431, 2014.

[30] "What are the mro crism products? a crism product primer," Mars Orbital Data Explorer (ODE). [Online]. Available: https://ode.rsl.wustl.edu/mars/pagehelp/Content/Missions_Instruments/Mars%20Reconnaissance%20Orbiter/CRISM/CRISM%20Product%20Primer/CRISM%20Product%20Primer.htm

[31] S. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda, and M. Laishram, "Multivariate statistical data analysis—principal component analysis (pca)," *International Journal of Livestock Research*, vol. 7, pp. 60–78, 2017.

[32] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," *CoRR*, vol. abs/1503.03585, 2015.

[33] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, "Pretraining is all you need for image-to-image translation," *arXiv preprint arXiv:2205.12952*, 2022.

[34] M. Cheon, S.-J. Yoon, B. Kang, and J. Lee, "Perceptual image quality assessment with transformers," *Computer Science ¿ Computer Vision and Pattern Recognition*, 2021, accepted to NTIRE workshop at CVPR 2021. 1st Place in NTIRE 2021 perceptual IQA challenge. [Online]. Available: https://doi.org/10.48550/arXiv.2104.14730