

THESIS

DEEP LEARNING FOR DOWNSCALING GOES-18 MEASUREMENTS FOR WILDFIRE
DETECTION

Submitted By:

Luke Taulbee

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2025

Master's Committee:

Advisor: Haonan Chen

Steve Simske

Chandrasekar Venkatachalem

Copyright by Luke Taulbee 2025

All Rights Reserved

ABSTRACT

DEEP LEARNING FOR DOWNSCALING GOES-18 MEASUREMENTS FOR WILDFIRE DETECTION

This thesis aims to address the challenge of accurate wildfire detection using satellite imagery. Despite the availability of various satellite-based fire products, real-time detection of fire perimeters remain difficult due to limitations in the spatio-temporal resolution of current satellite imagery. For example, the Geostationary Operational Environmental Satellites (GOES-R) series containing the Advanced Baseline Imager (ABI) offers high temporal resolution for frequent observations but suffers from low spatial resolution. In contrast, low Earth orbit (LEO) satellites like Suomi-NPP, NOAA-20, and NOAA-21 with the Visible Infrared Imaging Radiometer Suite (VIIRS) imager provide high spatial resolution but with limited temporal coverage. To overcome these limitations, this research proposes a deep learning framework for wildfire detection that leverages GOES ABI observations, which are downscaled to a spatial resolution of 375 meters using a Generative Adversarial Network (GAN). High-resolution VIIRS images are used as ground truth labels during the training phase. Experimental results demonstrate that the proposed framework successfully enhances the spatial resolution of GOES ABI data while preserving its high temporal frequency, allowing more precise and timely wildfire detection.

ACKNOWLEDGEMENTS

I would like to thank Dr. Haonan Chen for taking me on as one of his students, and for his valuable feedback, guidance, and patience throughout the course of this project. I am also deeply grateful to Yifan Yang for taking the time to guide me through any difficulties I encountered along the way. Both Dr. Chen and Yifan devoted their time and effort to carefully evaluating my work and helping me grow as a researcher.

I would also like to thank my family and friends for their support throughout my graduate studies. Their encouragement has been instrumental in helping me complete this journey.

DEDICATION

I would like to dedicate this thesis to my family.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
DEDICATION	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
 Chapter 1	
Introduction	1
1.1 Problem Statement and Motivations	1
1.2 Literature Review	3
1.3 Contributions of the Present Work	7
1.4 Thesis Organization	7
1.5 Conclusion	8
 Chapter 2	
Dataset: GOES, VIIRS, and Ground-Based Wildfire Boundary	10
2.1 Introduction	10
2.2 History of Satellite imagery	10
2.3 Satellite Working Principles	11
2.3.1 Brightness Temperature Conversion	13
2.3.2 GOES ABI Introduction	13
2.3.3 VIIRS Introduction	16
2.3.4 Examples of Wildfire imagery	20
2.4 Data Sources	22
2.5 Ground-based Wildfire Boundary	23
2.6 Conclusion	24
 Chapter 3	
Methodology	26
3.1 Introduction	26
3.2 Data Processing and Selection	28
3.2.1 Data Pairing	28
3.2.2 Active Fire Index	29
3.2.3 Data Processing	31
3.2.4 Data Selection	31
3.3 Processed Data Examples	32
3.4 Deep Learning For Super-Resolution Fire Detection	33
3.4.1 Introduction to Generative Adversarial Networks	33
3.4.2 Generative Adversarial Network Fundamentals	37
3.4.3 Convolutional Neural Networks	38
3.5 Enhanced Super-Resolution Generative Adversarial Network Architecture	39
3.5.1 Dense Residual Blocks	41
3.5.2 Residual In Residual Dense Block	44
3.6 GeneratorRRDB	45

3.7	Discriminator	52
3.7.1	Patch Creation	53
3.7.2	Discriminator Block	54
3.8	Feature Extraction	59
3.8.1	Feature Extraction	59
3.9	Loss Calculations	61
3.9.1	GeneratorRRDB Loss Calculations	61
3.9.2	Discriminator Loss Calculations	65
3.10	Normalization	67
3.11	Model Training	69
3.12	Evaluation Metrics	71
3.13	Fire Confidence Calculation	76
3.14	Conclusion	79
Chapter 4	Results Evaluation and Discussion	81
4.1	Introduction	81
4.2	Evaluation of Fire Detection and FireGAN	82
4.2.1	Fire Detection Results	82
4.3	Quantitative FireGAN Testing Results	86
4.4	Testing Maui, HW on August 9th, 2023	91
4.4.1	Full Maui Island Results	91
4.4.2	Lahiana, Maui Zoomed in Results	97
4.4.3	Center of Maui Island Results	103
4.5	Conclusion	109
Chapter 5	Summary and Future Work	111
5.1	Summary	111
5.2	Future Work	112
Bibliography		114

LIST OF TABLES

2.3.1 ABI Spectral Bands	14
2.3.2 VIIRS Spectral Bands	18
2.6.1 Satellite Imager Information	25
3.7.1 Discriminator Architecture Details	57
4.2.1 Fire Product Compared to VNP14 Results	82
4.2.2 Fire Product Compared to Wildfire Boundary Results	85
4.3.1 Full Testing Dataset Metrics	86
4.3.2 November 2, 2023 (12:10 UTC) Testing Metrics	88
4.3.3 July 1st, 2024 (12:10 UTC) Testing Metrics	90

LIST OF FIGURES

2.3.1 Planck Function on Earth Brightness Temperatures	15
2.3.2 VIIRS Planck Function on Earth Brightness Temperatures	19
2.3.3 Brightness Temperature Measurements for Smokehouse Creek Fire on February 27th .	20
2.3.4 Brightness Temperature Measurements for Alexander Mountain Fire on August 1st, 2024 (08:58 UTC)	21
2.3.5 Brightness Temperature Measurements for Smith River Complex Fire, California on September 20th, 2023 (21:12 UTC)	22
2.5.1 Ground-Based Wildfire Boundary	24
3.2.1 Pixel Values from Fire and Non-Fire Regions on August 9th, 2023 (12:12 UTC)	30
3.3.1 Processed Data Example Showing Fire: August 9th, 2023 (12:12 UTC)	32
3.3.2 Processed Data Example no Fire: March 5th, 2023 (12:12 UTC)	33
3.5.1 FireGAN Architecture	40
3.5.2 Dense Residual Block Architecture	41
3.5.3 Residual In Residual Dense Block Architecture	44
3.6.1 Generator Architecture	45
3.7.1 Discriminator Architecture	53
4.2.1 Fire Confidence Compared to VNP14: August 9th, 2023 at 12:12 (UTC)	82
4.2.2 Aggregate 3 Day Fire Confidence Compared to Ground Based Wildfire Boundary . . .	84
4.3.1 Testing Example: November 2, 2023 (12:10 UTC)	88
4.3.2 Testing Example: July 1st, 2024 (12:10 UTC)	90
4.4.1 Full Island C07 Results on August 9th, 2023	92
4.4.2 Full Island C14 Results on August 9th, 2023	93
4.4.3 Full Island AFI Results on August 9th, 2023	94
4.4.4 Full Island AFI and Fire Confidence Results on August 9th, 2023	96
4.4.5 Zoomed in Lahiana, HW region	97
4.4.6 Lahiana Fire C07 Results on August 9th, 2023	98
4.4.7 Lahiana Fire C14 Results on August 9th, 2023	99
4.4.8 Lahiana Fire AFI Results on August 9th, 2023	100
4.4.9 Lahiana Fire AFI and Fire Confidence Results on August 9th, 2023	102
4.4.10 Zoomed in center region	103
4.4.11 Center Region Fire C07 Results on August 9th, 2023	104
4.4.12 Center Region Fire C14 Results on August 9th, 2023	105
4.4.13 Center Region Fire AFI Results on August 9th, 2023	106
4.4.14 Center Region Fire AFI and Fire Confidence Results on August 9th, 2023	108

Chapter 1

Introduction

1.1 Problem Statement and Motivations

As a result of overgrown forests, prolonged droughts, and rising global temperatures, the risk of wildfires has significantly increased. Just in 2024 there were 8.9 million acres burned in the United States alone [1]. Wildfires cause long-term damage to local ecosystems by leaving behind ash and partially burned material, which not only hinders vegetation regrowth but also increases the risk of mudslides during rainstorms. More importantly, wildfires pose a significant risk to land, homes, and lives. Specifically, for the Maui, Hawaii fires in 2023 there were 115 fatalities and \$5.52 billion (USD) in rebuilding expenses, and 3200 acres burned most of which was populated [2]. Resulting from this is a need for improved wildfire detection to aid response efforts. Existing wildfire detection systems leverage both ground-based and satellite-based approaches, each with their respective advantages and disadvantages. Ground-based technologies, such as the use of aerial vehicles, are effective for localized and real-time fire detection but are often constrained by resource demands and coverage limitations. Aerial observations, including the use of drones, helicopters, and airplanes, add another layer of precision and adaptability to wildfire detection. These systems are capable of surveying remote and inaccessible areas, capturing detailed, high-resolution imagery, and even mapping the spread of active fires in real-time. Drones, in particular, offer the advantage of autonomous operation and can be equipped with infrared sensors for night detection. However, aerial methods are not without drawbacks. Manned aircraft require frequent fueling stops, routine maintenance, and skilled operators, all of which contribute to high operational costs. Their dependence on favorable weather conditions and limited flight times further reduces their ability to provide consistent, long-term surveillance.

Satellite-based systems, on the other hand, offer broad coverage and include platforms like low Earth orbit (LEO) sensors (e.g., Landsat, VIIRS) and geostationary orbit (GEO) sensors (e.g.,

GOES ABI). Along with satellites' increased coverage and reliability of images, they also offer multiple spectral bands and features, such as fire products, which enable efficient environmental monitoring. These fire products derive radiance captured from spectral bands which are converted to brightness temperatures. With brightness temperatures, fire detection algorithms such as GOES ABI FDC and VIIRS VNP14 are applied to calculate the pixel-wise likelihood of a fire for each satellite image [3] [4] [5].

This reduces the need for ground based methods such as the previously mentioned examples. For example, LEO satellites, such as MODIS, Suomi-NPP, and Landsat, to operate at relatively low altitudes, allowing them to capture images with high spatial resolution, which is effective for detecting smaller wildfires and accurately pinpointing hotspots. This detailed imagery allows for improved insights into the intensity and location of fires. However, even the VIIRS sensor is aboard three satellites (Suomi-NPP, NOAA-20, and NOAA-21) each with a revisit time of 16 days, which is still too infrequent of observations to effectivity monitor an active wildfire. This delay between observations creates a significant gap in real-time monitoring, as rapidly evolving fire conditions may go undetected until the next satellite pass. For instance, a fast-moving wildfire could spread extensively before updated imagery becomes available, making LEO satellites less effective for immediate fire detection efforts.

GEO satellites, on the other hand, remain fixed over a specific region, offering continuous and frequent observations with high temporal resolution. Specifically, the GOES-R series offers a 10 minute gap in between images, but with a 2 km resolution. These satellites are ideal for real-time wildfire monitoring, as they can provide a high volume of images, allowing responders to track fire progression and make timely decisions. However, because GEO satellites are positioned at much higher altitudes, their imagery has poor spatial resolution, making it difficult to identify smaller fires or discern precise hotspot locations. This lack of spatial detail can be a limitation in the early detection of fires, particularly in areas with complex landscapes or dense vegetation.

To address this limitation, an enhanced super-resolution generative adversarial network (FireGAN) was developed inspired by [6] to downscale images and improve their spatial resolution. While

[6] proposed visually pleasing results, it was trained on natural image datasets such as DIV2K, Flickr2K, and OutdoorSceneTraining, which consist of RGB images taken from personal cameras. While effective in training high-resolution images in the context of photography, this does not directly translate to infrared satellite imagery as the spectral characteristics and noise profiles differ from that of RGB images from a personal camera. Due to that, the FireGAN’s architecture is altered to more accurately fit a remote sensing application by including a deeper network and more complex design to accurately super-resolve brightness temperatures from satellite imagery. However, this method proposed by [6] serves as a foundation to improve upon for enhancing the spatial resolution of GOES ABI data, allowing more accurate wildfire detection by aligning it with the higher spatial resolution of the VIIRS sensor.

By overcoming GOES ABI’ coarse 2 km spatial resolution and VIIRS sensor’s limited 16-day temporal resolution, this research enhances wildfire monitoring and response capabilities by leveraging GOES ABI’ 10-minute temporal frequency alongside VIIRS’ 375 m spatial resolution. The study uses data from 2023 to 2024 focused on Maui, Hawaii, to generate high-resolution downsampled imagery suitable for wildfire detection algorithms. The core objective is to develop a deep learning (DL)-based framework that downscales GOES ABI measurements to improve spatial resolution while preserving their high temporal resolution. This enables more accurate and timely wildfire detection. By employing a Generative Adversarial Network (GAN), the framework produces high-resolution wildfire maps at 375 m spatial resolution with a 10-minute frequency, effectively bridging the gap between GOES ABI and VIIRS sensor data. This strengthens wildfire monitoring strategies, allowing emergency services to pinpoint hotspots with greater precision and respond more efficiently.

1.2 Literature Review

Numerous studies have explored the application of deep learning for image super-resolution and this section explores deep learning based approaches for super-resolution, along with its application in remote sensing, and finally its application for wildfire detection.

An influential super-resolution paper by Wang et al. [6] introduced the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN), which builds upon the earlier Super-Resolution GAN (SRGAN). This work presented several key advancements. Notably, it integrated Residual-in-Residual Dense Blocks (RRDB), allowing the model to learn image features in a hierarchical manner while enhancing feature extraction by removing batch normalization and applying residual scaling. Additionally, ESRGAN replaced the traditional discriminator with a Relativistic Average GAN (RaGAN), allowing the model to evaluate how realistic a generated image is relative to a real one, rather than treating them as independent which helps mitigate hallucinations from the GAN. Lastly, the authors improved perceptual loss by extracting features from the VGG-19 network before the activation layers, which proved to be more effective for capturing fine-grained textures. While ESRGAN demonstrated impressive perceptual quality, but this research mainly focused was on visual realism rather than pixel-level accuracy. As a result, this method may fall short in applications requiring accurate numerical consistency with the ground truth, such as generating remote sensing imagery.

Another super-resolution method proposed by Cai et al. [7] introduced a Variational Autoencoder (VAE) that extracts features from high-dimensional datasets and assigns probabilistic representations for image reconstruction. This approach incorporates a latent variable model, where latent variables are generated at lower levels of the network to preserve spatial information from the original images as they are passed through the decoder. To further enhance reconstruction quality, the decoder is supplemented with residual block networks, effectively increasing the model's depth and capacity. While this method contributed valuable insights into super-resolution, the reconstructed images remained relatively blurry which can be attributed to the shallowness of the network and a limited set of the loss functions.

Next, an application of the ESRGAN framework proposed by Clabaut et. al. [8], proposed a model trained on a highly diverse dataset comprising of the DIVERse 2K (high resolution dataset), along with airborne and satellite imagery with spatial resolutions ranging from sub-meter (< 1 m) to 5 meters. This variety in training data encouraged the ESRGAN to achieve strength in generaliza-

tion across different images and environmental conditions. Training was conducted over several stages, specifically at 150, 300, 600, 1200, 2400, and 4800 epochs, with progressively improved results. The progressive improvements with a higher number of epochs was due to the adversarial nature of the GAN, where the model discriminates against itself to identify generated versus real high resolution imagery, effectively the model adjusts itself to achieve higher accuracy. The model yielded Peak Signal-to-Noise Ratio (PSNR) values ranging from 30.11 dB to a maximum of 39.1 dB, indicating high-quality reconstructions and minimal distortion relative to ground truth images. This study highlights ESRGANs potential in remote sensing and environmental monitoring applications, particularly in scenarios where image accuracy is important. However, it focuses on very high resolution reconstructions which include buildings, trees, etc. which is useful in object level applications, but does not capture a large enough area for environmental monitoring and more specifically wildfire monitoring.

In the research by Shao and Wang [9], the authors proposed an encoder-decoder architecture to perform super-resolution on three datasets: NWPU VHR-10 (1-meter resolution), ZY-3 (5.6-meter resolution), and MOMs-2P (18-meter resolution). This study introduced a Coupled Sparse Autoencoder (CSAE), which learns the relationships between low-resolution and high-resolution images. The CSAE extracts sparse coefficients from low-resolution patches so the images are segmented. During training, a Sparse Coefficients of Residuals (SCOR) module is used to predict the residual features, allowing the model to refine and enhance the high-frequency details that may be lost in low-resolution images. Essentially, the CSAE-SCOR framework maps low- to high-resolution representations by focusing on both the sparse structure and the residual information, resulting in improved reconstruction accuracy. The final high-resolution image is reconstructed by summing the predicted residuals with the upsampled low-resolution image. Which again is effective in its own application, but the dataset shows a high (1-18 meter) resolution, which is not ideal for large scale environmental monitoring. Additionally, the study places greater emphasis on perceptual quality rather than pixel-level accuracy, which is important in applications such as wildfire monitoring, where accurate pixel representations are important.

The last SR for remote sensing example was done by Salgueiro et. al. [10], who proposed a variant of the ESRGAN named the Segmentation Enhanced-ESRGAN (SEG-ESRGAN) which utilizes an encoder-decoder architecture to perform image segmentation. In this research they use the WorldView-2 and Sentinel-2 satellites with 2 meter and 10 meter resolutions respectively. In this framework, the encoder is composed of Residual-in-Residual Dense Blocks (RRDBs) which extracts high-level features from the low-resolution input. The decoder then applies transposed convolutions (also known as deconvolutions) to reconstruct a segmentation map, capturing structural elements such as object boundaries and regions in the images. This segmentation map is used to guide the super-resolution process, allowing the network to focus on meaningful features. By combining super-resolution and segmentation in the framework, SEG-ESRGAN can achieve strong perceptual and structural quality in the generated high-resolution images. Particularly in very high 2 meter resolution scenarios where precise object shapes and boundaries are important, which is useful in object level analysis.

One example of super-resolution for detecting wildfires was done by Badhan et. al. [11], who proposed an autoencoder specifically for wildfire detection by designing an autoencoder to match GOES-17's 2 km resolution to match the VIIRS sensor's 375 m. They proposed an autoencoder where the images are broken down through convolutional layers in the encoder and then rebuilt through the decoder architecture in an inverse fashion of the encoder. This extracts features at high to low levels for each of the images. While partially effective, there is a limitation in the model's ability to accurately capture high-resolution fire-related features. However, the autoencoder architecture lacks adversarial training or perceptual loss components, which can restrict its capability to reconstruct sharp, high-resolution fire signatures from the lower-resolution GOES ABI imagery. As a result, the model performs well in aligning the resolutions between GOES ABI and VIIRS, but struggles with precise wildfire identification, as shown with its low Intersection over Union (IOU) metric which ranges from 0.12-0.14.

1.3 Contributions of the Present Work

The contributions of this work include:

- Application of the Enhanced Super-Resolution Generative Adversarial Network (FireGAN) to generate high-quality, high-resolution synthetic brightness temperatures aligned with the 375 m spatial resolution of VIIRS sensor. This results in 375 m-resolution downsampled brightness temperature at 10-minute intervals using GOES ABI observations.
- Demonstration of improved wildfire detection capabilities through visual and quantitative comparisons between original and super-resolved GOES ABI data.
- Application of wildfire confidence values assigned to FireGAN generated images. Allowing faster and more precise identification of fire hotspots from GOES ABI imagery.
- Application of quantitative metrics such as PSNR, SSIM, R^2 , and RMSE, along with a ground-based wildfire boundary map from Maui, Hawaii, to evaluate the accuracy of the generated images and validate their corresponding fire confidence outputs.

1.4 Thesis Organization

This thesis is organized by the following

- Chapter 2: Dataset of GOES ABI, VIIRS sensor, and Ground-Based Wildfire Boundary where the fundamentals of GOES ABI and VIIRS satellite imagers are introduced, including their own respective spectral bands and their applications in this research for wildfire detection. Next is an introduction to Planck Radiance and its conversion to brightness temperature. Next is the Active Fire Index Calculation which will be used later in the research as an aid to identifying wildfire hotspots. Finally, ground-based wildfire boundary data is introduced, which is used later in the study to validate satellite-based results.
- Chapter 3: Study domain and dataset introduces the data sources, pairing, processing, and selection. Where the VIIRS satellites orbits line up with the study domain, how those images

are processed, and which are selected for training and testing datasets. Next, Deep Learning for Super-Resolution and Wildfire Detection, introduces the fundamentals of deep learning relevant to this study, including the FireGAN architecture and training methodology. This chapter also explains the fire confidence calculation and how the model is integrated with concepts from chapter 2 and the beginning of chapter 3 to generate high-resolution and relevant imagery.

- Chapter 4: Results evaluation and discussion walks through the results from the testing dataset, testing for August 9th, 2023 (which shows brightness temperatures, AFI calculations, and fire confidence results), and then specific fire detection accuracy results in comparison to the ground-based Maui, HW wildfire boundary.
- Chapter 5: Summary and Future Work provides a comprehensive overview of the key findings from Chapters 2 through 4. It also outlines potential improvements and extensions for future research that can build upon this work’s contributions to wildfire detection using enhanced satellite imagery.

1.5 Conclusion

This chapter introduced the need for improved wildfire detection due to the increasing frequency and severity of wildfires, driven by climate change, overgrown forests, and prolonged droughts. It discussed the limitations of current ground-based and satellite-based detection systems, emphasizing the trade-off between spatial and temporal resolution. While LEO satellites that contain the VIIRS sensors offer high 375 m spatial resolution but infrequent coverage of 16 days, GEO satellites like GOES ABI provide near-real-time data at 10 minute intervals, but at the expense of image clarity with its 2 km spatial resolution.

To address this challenge, this thesis proposes a deep learning-based super-resolution framework, FireGAN, designed to enhance GOES ABI imagery by increasing its spatial resolution to 375 meters. By aligning the high temporal resolution of GOES ABI (10-minute intervals) with the fine spatial resolution of the VIIRS sensor, FireGAN enables more timely and accurate wildfire

detection by bridging the strengths of geostationary (GEO) and low Earth orbit (LEO) satellite systems.

The literature review presented a range of existing methods in the field of image super-resolution, including ESRGAN, VAEs, RDN, and their adaptations for remote sensing and airborne imagery. While these approaches have shown success in various domains, few directly address the challenges posed by wildfire monitoring using infrared satellite data. This motivates the need for a GAN-based super-resolution approach that is capable of accurately enhancing brightness temperature measurements for fire detection.

Looking ahead, the following chapters will detail the dataset details, including GOES ABI and VIIRS sensor data, Planck Radiance and the brightness temperature conversion, and the ground-based wildfire boundary.

Chapter 2

Dataset: GOES, VIIRS, and Ground-Based Wildfire Boundary

2.1 Introduction

Building off of the last chapter which emphasized the need for improved wildfire monitoring with the aid of deep learning based super-resolution techniques. Along with understanding the various models such as ESRGAN, VAE, and RDNs that can be used to super-resolve satellite imagery. This chapter provides an overview of the GOES ABI (Geostationary Operational Environmental Satellite) and VIIRS (Visible Infrared Imaging Radiometer Suite) sensors aboard the Suomi-NPP, NOAA-20, and NOAA-21 satellites, with a focus on radiance, brightness temperature, and their respective spatial and temporal resolutions. Where radiance refers to the amount of electromagnetic energy emitted or reflected from the Earth's surface and is measured from satellite sensors across various spectral bands. Brightness temperature, derived from radiance measurements, represents the temperature of an object as measured by a thermal sensor. It serves as an indicator in identifying thermal anomalies, such as active wildfires, which typically emit very high infrared radiation, resulting in high brightness temperatures. Similarly, calculations such as the Active Fire Index (AFI) [12] can be calculated to further isolate wildfire hotspots. Next, the data sources are introduced, including where the data is from and their native file format. Lastly, this chapter introduces a fire boundary map for Maui, Hawaii, established through ground-based methods, which will later be used to validate the effectiveness of the models in identifying wildfire hotspots.

2.2 History of Satellite imagery

Satellite imagery has revolutionized the way we observe and understand the Earth's environment, offering global perspective on ecological and atmospheric changes. This technology allows for

continuous, large-scale monitoring of environmental events, making it a useful tool for environmental research, disaster management, and climate studies. By capturing electromagnetic radiation reflected or emitted from the Earth's surface, satellites provide multi-spectral data that enable researchers to analyze and monitor land cover, natural disasters, water bodies, air quality, and atmospheric conditions.

The use of satellite imagery for environmental observation dates back to the launch of early Earth observation satellites, such as NASA's Landsat program [13] and NOAA's GOES-A in 1975 [14] in the early 1970s. Since then, technological advancements have led to the development of high-resolution optical and thermal imaging sensors aboard satellites like MODIS (Moderate Resolution Imaging Spectroradiometer), VIIRS (Visible Infrared Imaging Radiometer Suite), and GOES (Geostationary Operational Environmental Satellites). These instruments provide continuous, near-real-time data that have significantly improved our ability to track environmental changes, predict natural disasters, and assess the impact of human activities on ecosystems.

One of the many applications of satellite imagery is in monitoring wildfires and assessing their impacts. Thermal sensors onboard satellites can detect active fires, estimate fire intensity, and map burned areas [15]. This capability is particularly important for understanding fire behavior, mitigating risks, and supporting emergency response efforts. Additionally, satellite data enables post-fire vegetation recovery monitoring and long-term ecological assessment, contributing valuable insights to wildfire management strategies.

2.3 Satellite Working Principles

A multispectral sensor captures images in specific wavelength bands, typically covering visible (RGB), near-infrared (NIR), shortwave infrared (SWIR), and thermal infrared (TIR) regions [16]. The sensor detects and records reflected or emitted radiation from the Earth's surface, allowing for environmental analysis based on spectral differences between different materials [16]. Specifically, Max Planck mathematically described how a black body, an idealized perfect emitter in thermal equilibrium, emits radiation at various wavelengths depending on its temperature. Planck's Law

explains how the intensity of emitted radiation increases with temperature and how the peak of the emission spectrum shifts toward shorter wavelengths as temperature rises. This phenomenon forms the theoretical foundation for satellite-based thermal remote sensing. Planck's Law is expressed as [17]:

$$B(\nu, T) = \frac{8\pi h\nu^3}{c^3} \cdot \frac{1}{e^{\frac{h\nu}{k_B T}} - 1} \quad (2.1)$$

Where B is the spectral radiance of a body, λ (ν) is the wavelength, T is the absolute temperature, h is the Planck constant, defined as 6.62×10^{-34} Joule $\cdot$ Hertz $^{-1}$, c is the speed of light in a vacuum, equal to 299,792,458 m/s, and k_B is known as the Boltzmann constant, which is defined as:

$$k_B = 1.380649 \times 10^{-23}, \text{J/K} \quad (2.2)$$

Planck's derivation was motivated by efforts to resolve limitations in earlier models of black body radiation, such as the Rayleigh-Jeans Law. The Rayleigh-Jeans Law describes spectral radiance at longer wavelengths but failed at shorter wavelengths, predicting an unphysical "ultraviolet catastrophe" [18]. It is defined as:

$$B_\nu(T) = \frac{2\nu^2 k_B T}{c^2} \quad (2.3)$$

Planck's Law resolved the limitations of classical physics by explaining blackbody radiation by introducing the concept of quantized energy levels. In the context of satellite remote sensing, Planck's Law plays an important role by allowing the conversion of radiance values, measured by thermal sensors, into brightness temperatures. These brightness temperatures are essential for identifying thermal anomalies on Earth's surface, such as active wildfires, making Planck's Law fundamental concept in fire detection algorithms.

2.3.1 Brightness Temperature Conversion

Generally, satellite sensors measure radiance and provide data composed of these radiance values, which represent the amount of energy emitted by the Earth's surface or atmosphere. However, radiance alone does not directly indicate the physical temperature of the observed region. Therefore, this research derives brightness temperature from Planck's Law, allowing the estimation of surface temperature at specific locations.

$$T_B = \frac{hc}{k_B \lambda} \left[\ln \left(1 + \frac{2hc^2}{\lambda^5 L_\lambda} \right) \right]^{-1} \quad (2.4)$$

Definition of the brightness temperature equation where: T_B is the brightness temperature, h is Planck's constant ($6.62607015 \times 10^{-34}$ J·s), c is the speed of light (3.00×10^8 m/s), k_B is the Boltzmann constant (1.380649×10^{-23} J/K), λ is the wavelength of the radiation, L_λ is the spectral radiance at wavelength λ .

With the brightness temperature computed, wildfire detection becomes more straightforward, as thermal anomalies relative to the background are more easily identifiable. Wildfires emit significantly more radiance compared to their surroundings especially evident in bands such as C07 and I04 as previously stated. This increased radiance leads to higher brightness temperatures, making the thermal anomalies more distinct and aiding in the accurate detection of wildfire locations [?].

2.3.2 GOES ABI Introduction

The Geostationary Operational Environmental Satellites (GOES ABI) are a series of weather and environmental monitoring satellites operated by the National Oceanic and Atmospheric Administration (NOAA). These satellites provide continuous, near-real-time observations of Earth's atmosphere, oceans, and land surfaces. Positioned in geostationary orbit, GOES satellites remain fixed over specific regions on Earth, allowing uninterrupted monitoring of rapidly evolving events such as wildfires, and other natural disasters.

The most recent generation the GOES-R series which includes GOES-16, GOES-17, and GOES-18 satellites. All of which orbit approximately 22,236 miles above Earth's surface [?]. This research

focuses on GOES-18 (west), which is positioned to observe the western United States, including the Hawaiian Islands and specifically Maui. Since GOES-18 observes a fixed region, it can provide frequent updates with a temporal resolution of approximately 10 minutes, which is especially valuable for real-time tracking and emergency response during natural disasters.

While GOES-18 offers excellent temporal coverage, its spatial resolution is limited. The Advanced Baseline Imager (ABI) onboard GOES-18 captures imagery at a spatial resolution of 2 km for its spectral bands, which is not sufficient for detecting finer details such as small or early-stage wildfires.

The ABI sensor on the GOES-R series provides 16 spectral bands, allowing a wide range of environmental observations [19], as summarized in the table below:

Table 2.3.1: ABI Spectral Bands

Bands	Wavelength (μm)	Primary Use
Visible Bands (1-2)	0.47 - 0.64	Cloud cover, smoke detection, land surface mapping
Near-Infrared Bands (3-6)	0.86 - 2.25	Vegetation monitoring, cloud properties, water vapor
Shortwave Infrared Bands (7)	3.9	Wildfire detection, nighttime cloud imaging
Water Vapor Bands (8-10)	6.2 - 7.3	Mid and upper-level atmospheric moisture
Thermal Infrared Bands (11-16)	8.4 - 13.3	Temperature monitoring, cloud top height, fire detection

Table 2.3.1 visually demonstrates the relationship between the Planck function and the spectral bands of the GOES ABI, focusing on Earth-like temperatures along with brightness temperature emitted by wildfires.

The following visualizes the connection of Planck Radiation in comparison to GOES ABI spectral bands:

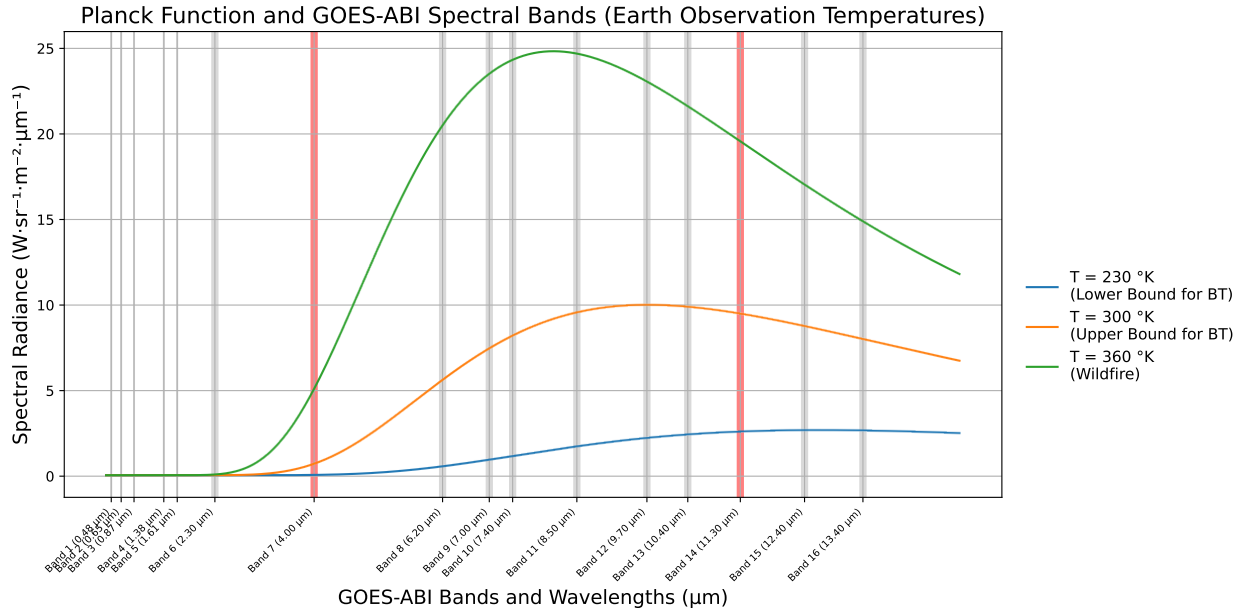


Figure 2.3.1: Planck Function on Earth Brightness Temperatures

The Planck curves represent the spectral radiance emitted by a black body at different temperatures ($230^{\circ}K$, $300^{\circ}K$) in degrees Kelvin, which represent a common threshold for brightness temperatures emitted from earth. The common temperatures correspond to different regions on Earth and more specifically Maui, HW as this contains common temperatures from the high altitude (non-erupting) volcano, to dense forest vegetation, and down to costal areas. Then, a high brightness temperature curve shown at $360^{\circ}K$ [3] represents the thermal signature of wildfires, as it significantly exceeds the normal brightness temperature temperature thresholds. This plot also overlays the GOES ABI spectral bands as shaded regions, which correspond to the specific wavelength ranges used by the satellite to observe Earth's atmosphere and surface. The red shaded regions highlight bands C07 and C14 which are used in this research.

More specifically, Band C07 ($4.0 \mu m$) is sensitive to wildfire hotspots due to its placement in the medium-wave IR (MWIR) region, where small increases in temperature result in a significant rise

in spectral radiance. This can be seen in the plot, where higher temperatures (e.g., 360 K) cause a steep increase in radiance compared to lower temperatures. Making C07 ideal for detecting active fires, as it responds strongly to the thermal emittance from flames. In contrast, Band C14 (11.3 μm) operates in the longwave infrared (LWIR) region and is primarily used for measuring surface and cloud-top temperatures. As shown in Figure 2.3.1, the change in radiance with brightness temperature in this band is more gradual. Although C14 is not as sensitive to high-temperature fire hotspots as C07, it is effective for identifying smoke plumes. Smoke particles emit thermal radiation at this wavelength, allowing C14 to distinguish smoke from background surfaces and clouds.

Together, Bands C07 and C14 complement each other in wildfire monitoring as C07 excels in detecting fire hotspots, while C14 aids in tracking smoke swaths, offering an informative assessment of wildfire activity and its atmospheric impacts.

The selection of Bands 7 and 14 also aligns with the GOES ABI Fire Detection and Characterization (FDC) product which is calculated for each GOES ABI image every 10 minutes. As specified by [4], the GOES ABI FDC algorithm utilizes Band 7 for hotspot identification, which is complemented by Band 14 for detecting sub-pixel fires. Together, these two bands allow the FDC algorithm to establish thresholds of wildfire presence based on brightness temperatures from both Bands 7 and 14.

2.3.3 VIIRS Introduction

The Visible Infrared Imaging Radiometer Suite (VIIRS) is an instrument aboard the Suomi National Polar-orbiting Partnership (Suomi-NPP), NOAA-20, and NOAA-21 satellites [20]. The VIIRS sensor provides high-resolution, multi-spectral imaging that complements GOES ABI observations by offering global coverage with twice-daily observations with ascending and descending observations. Unlike GOES ABI, which operates in a geostationary orbit, VIIRS is onboard polar-orbiting satellites. There are significant drawbacks resulting from VIIRS' polar orbit and that is the 16 day temporal resolution after the ascending and descending images are captured over a specific

sample area such as Maui, HW. Since the temporal resolution is 16 days there is too significant of a time delay in between images to be useful for emergency services by just using VIIRS.

In terms of spatial resolution, the VIIRS sensor can achieve resolutions ranging from 375 m to 750 m, depending on the spectral band, offering a balance between high spatial resolution and global coverage [20]. Specifically, this research uses spectral bands with 375 m resolution (I bands), offering very high spatial resolution over sample regions. In addition, the VIIRS sensor can provide better insight into the intensity, location, and spread of the fire in comparison to GOES ABI measurements. When available, this information is very useful for emergency services to specifically locate fire hotspots.

The VIIRS imager is equipped with 22 spectral bands, ranging from visible to thermal infrared wavelengths. These bands support a wide range of environmental and atmospheric monitoring applications, including ocean color monitoring, vegetation health assessment, cloud and weather studies, and fire detection all fundamentally similar to the GOES ABI Bands. VIIRS images are particularly useful for identifying wildfire boundaries due to its high spatial resolution. The VIIRS sensor spectral bands are as follows [20]:

Table 2.3.2: VIIRS Spectral Bands

Band ID	Wavelength Range (μm)	Primary Use
M1	0.402 - 0.422	Ocean Color, Aerosols
M2	0.436 - 0.454	Ocean Color, Aerosols
M3	0.478 - 0.498	Ocean Color, Aerosols
M4	0.545 - 0.565	Ocean Color, Aerosols
M5	0.662 - 0.682	Ocean Color, Vegetation
M6	0.740 - 0.758	Atmospheric Correction
M7	0.846 - 0.885	Vegetation Index
M8	1.23 - 1.25	Cloud Optical Properties
M9	1.36 - 1.38	Cirrus Detection
M10	1.58 - 1.64	Snow and Cloud Detection
M11	2.225 - 2.275	Cloud Properties
M12	3.60 - 3.78	Sea Surface Temperature
M13	3.97 - 4.13	Active Fires, SST
M14	8.40 - 8.70	Cloud Top Properties
M15	10.26 - 11.26	Sea Surface Temperature
M16	11.54 - 12.49	SST, Cloud Properties
I1	0.58 - 0.68	High-Resolution Imagery (VIS)
I2	0.845 - 0.885	Vegetation Index (NIR)
I3	1.58 - 1.64	Snow and Cloud Detection
I4	3.55 - 3.93	Active Fires, Clouds
I5	10.3 - 12.5	SST, Imagery Band
DNB	0.5 - 0.9	Day/Night Low-Light Imaging

Similar to Table 2.3.1, Table 2.3.2 highlights I04 and I05 VIIRS spectral bands which are very similar to GOES ABI' C07 and C14 in identifying wildfire hotspots and smoke swaths.

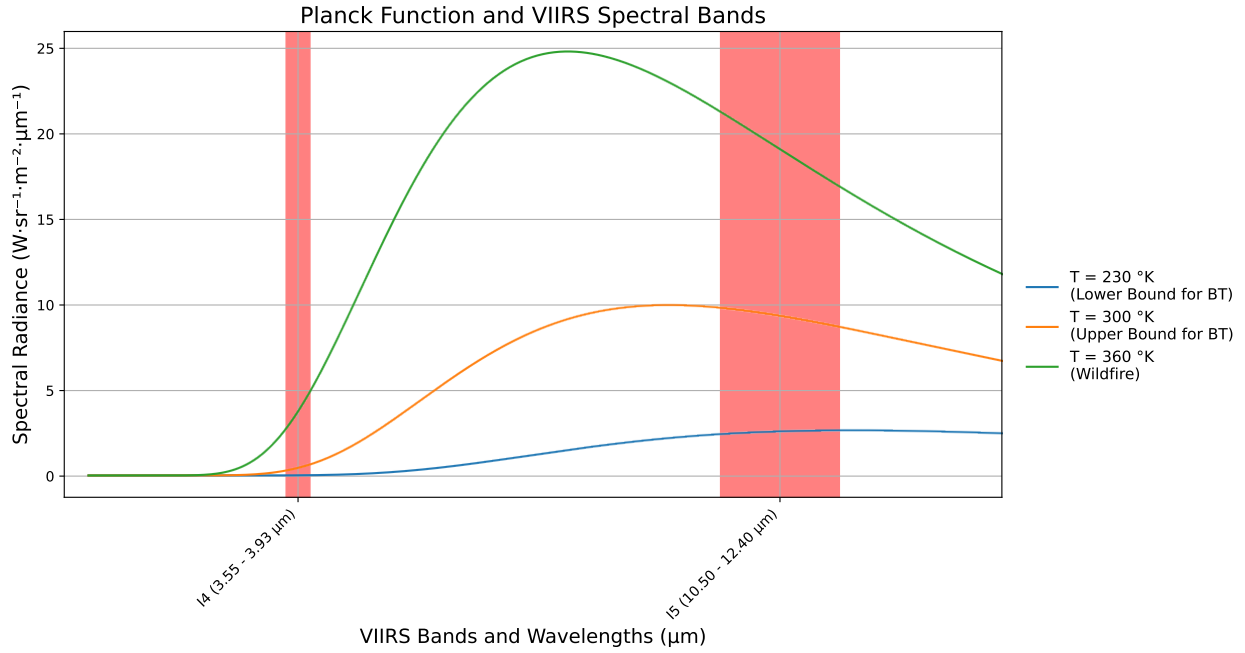


Figure 2.3.2: VIIRS Planck Function on Earth Brightness Temperatures

By using I04 and I05 bands along with VIIRS sensor's 375 m resolution measurements there is a high spatial resolution for the model to use as label images when downscaling GOES ABI measurements to an improved resolution. The increased spatial detail from VIIRS sensor's 375 m resolution enables the model to better capture precise fire locations that GOES ABI alone would not be able to achieve. This increased precision directly enhances the super-resolution process, leading to sharper and more accurate GOES ABI imagery. As a result, VIIRS data plays an important role as a labels in generating high-quality super-resolved outputs.

Similar to the GOES ABI FDC product, the VIIRS sensors also provide a fire product (VNP14), which is calculated using Bands I04 and I05, as described in [5]. The VNP14 product estimates fire likelihoods by categorizing pixels into three confidence levels: low, nominal, and high. These confidence levels represent the probability of a pixel being associated with a fire, aiding in the identification of wildfire hotspots for every VIIRS image that is captured.

2.3.4 Examples of Wildfire imagery

Examples of wildfires producing high brightness temperatures are illustrated below, beginning with the Smokehouse Creek Fire in the panhandle region of Texas:

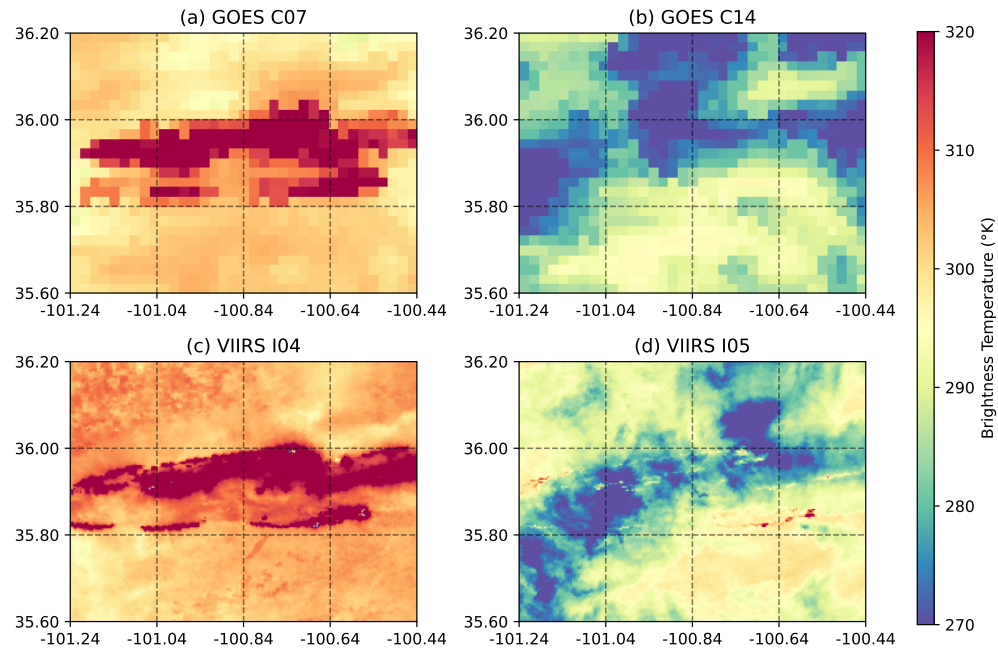


Figure 2.3.3: Brightness Temperature Measurements for Smokehouse Creek Fire on February 27th, 2024 (19:20 UTC): (a) GOES ABI C07, (b) GOES ABI C14, (c) VIIRS I04, (d) VIIRS I05

In total the Smokehouse Creek wildfires had burned approximately 1.05 million acres, the total burntime spanned between February 26th to March 16th [21]. This example displays clearly defined wildfire hotspots in Figure 2.3.3, with a clear fire boundary shown from both GOES ABI and VIIRS images, more specifically the VIIRS sensor aboard the Suomi-NPP satellite. Notably, channels C07 (GOES ABI) and I04 (VIIRS) demonstrate strong capability in detecting these hotspots. However, this figure also highlights a limitation of the lower spatial resolution in GOES ABI imagery: while the C07 band detects what appears to be one large fire, the VIIRS sensor is able to identify two distinct smaller fires. Addressing this discrepancy is one of the goals of this research by improving the spatial precision of GOES ABI-derived wildfire boundaries.

Next, is the Alexander Mountain Fire, which burned between Estes batch and Loveland, Colorado, on the north side of US Highway 34:

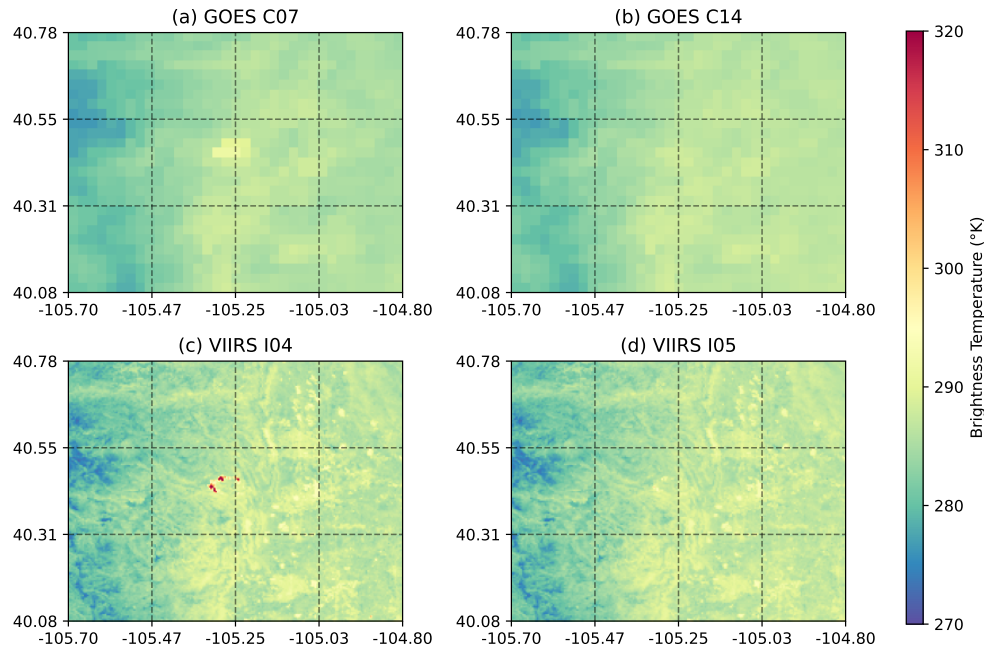


Figure 2.3.4: Brightness Temperature Measurements for Alexander Mountain Fire on August 1st, 2024 (08:58 UTC): (a) GOES ABI C07, (b) GOES ABI C14, (c) VIIRS I04, (d) VIIRS I05

This image captures the Alexander Mountain Fire, which burned a total of 9,669 acres between July 29th and August 18th, 2024 [22]. For this case, the VIIRS sensor was sourced from the NOAA-21 satellite. Similar to the Baja California example, distinct wildfire hotspots are visible, particularly in the VIIRS I04 channel. However, because this fire was significantly smaller in size, the GOES ABI imagery failed to fully capture the fire. This highlights a key constraint of GOES ABI: wildfires must be large enough to occupy at least one 2 km pixel to be clearly detected. It is not necessary for the entire kkm pixel area to be burning, but the fire must be large enough in comparison to its surroundings to heighten the brightness temperature enough that would indicate a fire. This constraint further emphasizes the need for high resolution enhancement techniques to improve wildfire detection using GOES ABI data.

Last, is the Smith River Complex Fire in Northern California:

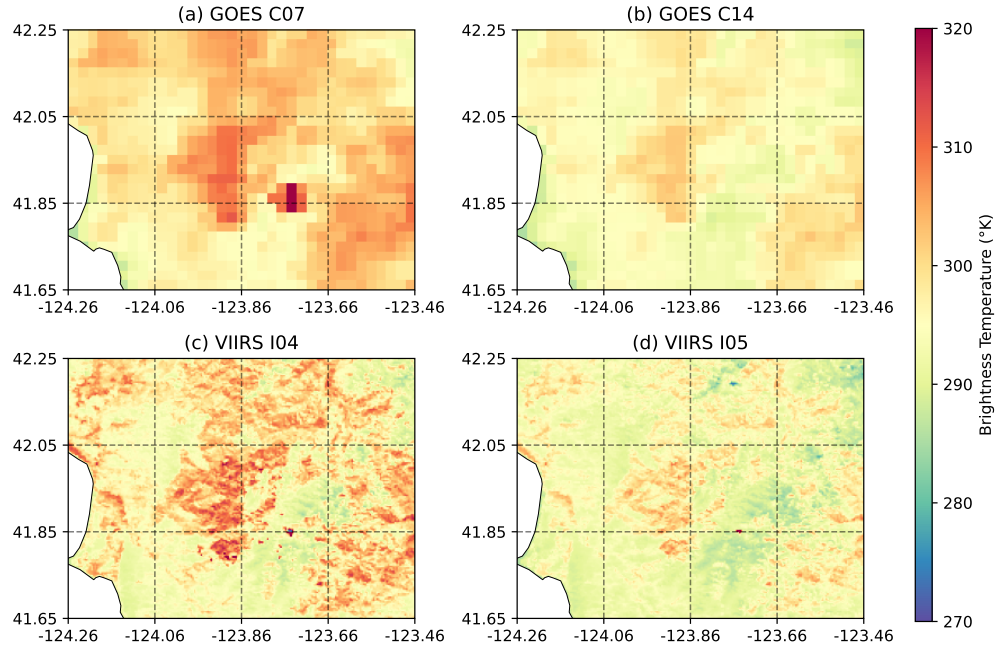


Figure 2.3.5: Brightness Temperature Measurements for Smith River Complex Fire, California on September 20th, 2023 (21:12 UTC): (a) GOES ABI C07, (b) GOES ABI C14, (c) VIIRS I04, (d) VIIRS I05

The Smith River Complex Fire in California spanned over a wide region totalling in 95,107 acres burnt over the course of August 15th, 2023 to November 17th, 2023, meaning it was active for 94 days [23]. While this fire burned a significant acreage, it was not all at once as indicated by the relatively small GOES ABI C07 pixels which indicate the primary hotspot. Additionally, scattered VIIRS I04 hotspots, captured by the Suomi-NPP satellite, are visible on the plot and may reflect smaller, spreading fires branching out from the main fire area. This is another example that was highlighted in 2.3.4 where individual smaller hotspots were not as distinguishable on GOES ABI imagery in comparison to VIIRS measurements.

2.4 Data Sources

Data from both GOES ABI and VIIRS sensors were utilized in this study, as both are commercial satellites accessible through government supported and private platforms. The data sources are summarized as follows:

- GOES ABI data is available through Amazon Web Services (AWS) [?] and contains radiance data along with geolocations within each downloadable file.
- VIIRS sensor data is available through the "Level-1 and Atmosphere Archive and Distribution System Distributed Active Archive Center" (LAADS DAAC) for all Suomi-NPP, NOAA-20, and NOAA-21 [24]. VIIRS data is available in two primary formats: one containing brightness temperatures and another containing geolocation data. The "Level 1" data title indicates that the images were calibrated for brightness temperature and geolocation.

2.5 Ground-based Wildfire Boundary

Lastly, to validate the satellite-based findings, a boundary representing the total burn area from the Maui wildfires was provided by the University of Hawaii team. This dataset was derived from ground-based methods, which were effective in capturing a detailed representation of the wildfire perimeter. By comparing the satellite-based fire confidence maps to this detailed ground-truth boundary, we are able to evaluate the accuracy and reliability of the super-resolution fire detection approach. The fire confidence maps, generated from super-resolved GOES ABI imagery, represent areas with varying levels of fire probability. Thus, by overlapping the satellite-generated fire confidence map with the ground-based boundary, we can assess how well the satellite system detects fire regions.

The following plot is an example of the ground based fire boundary:

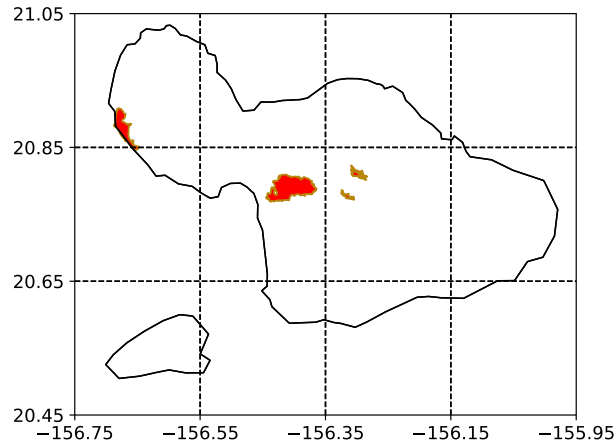


Figure 2.5.1: Ground-Based Wildfire Boundary

It is important to note that the ground-based boundary depicted in the above plot was assembled based on the wildfires in Maui, which were primarily active between August 8th and August 10th of 2023. The boundary was drawn using data collected within this specific timeframe, covering all four major fire locations on the island. Since wildfires rapidly evolving, the boundary corresponds to the total burn area as it appeared during those three days.

With the boundary shown in the above plot, there is validating data to assess the super-resolved outputs and fire confidence maps to ensure the results are within the fire boundary.

2.6 Conclusion

To conclude, this project will use the brightness temperatures which are calculated from Planck Radiation. Using data from both the GOES-18 and VIIRS sensors with the following spectral bands and satellite information:

Table 2.6.1: Satellite Imager Information

Satellite Sensor	Central Wavelength (μm)	Spatial Resolution (m)	Temporal Resolution
VIIRS I04	3.74	375	16 Days
VIIRS I05	11.45	375	16 Days
GOES-18 C07	3.9	2000	10 Minutes
GOES-18 C14	11.2	2000	10 Minutes

This combination of GOES ABI and VIIRS sensor data allows the model to leverage the strengths of both sensors, capturing frequent observations from GOES-18 while maintaining the finer spatial details from VIIRS images. With these datasets, the model can downscale GOES ABI images to achieve a VIIRS resolution. The combination of high temporal and spatial resolution data enables more precise wildfire monitoring, ultimately aiding in early detection and response efforts as well. In conclusion, this project will utilize brightness temperatures derived from Planck Radiation, using data from both GOES-18 and VIIRS satellites. The combination of these two satellite systems, with their strengths and weaknesses in spatial and temporal resolution. VIIRS sensor provides fine spatial resolution (375 m) but with less frequent coverage (16 days), while GOES-18 offers high temporal resolution (10 minutes) at a coarser spatial resolution (2 km). This connection enables the enhancement of GOES ABI imagery through a super-resolution approach, ensuring more accurate and timely wildfire monitoring.

The following chapter presents the methodology for data preprocessing, the architecture of the FireGAN model, training procedures, and the design of the fire confidence metric. This chapter directly addresses the spatial and temporal limitations discussed above using a deep learning-based approach.

Chapter 3

Methodology

3.1 Introduction

Building upon the previous chapter’s discussion of satellite data, this chapter introduces the study domain and outlines the steps in dataset processing, including converting radiance to brightness temperature for both VIIRS sensor and GOES ABI images, calculating AFI, and preparing the ground-based wildfire boundary, which is required for training and evaluating the FireGAN model. The success of the model is linked to the quality of the input data, as clear, well-processed satellite imagery is essential for generating accurate super-resolved outputs. Data preprocessing is an important phase in this process, as it helps to reduce noise, mitigate artifacts, enhance model performance, and ensure that the generated super-resolved images are as accurate and reliable as possible.

The study area is focused on Maui, Hawaii, chosen due to the high-profile and well-documented wildfires that occurred in 2023. The recent availability of high-resolution satellite data corresponding to these events makes Maui an ideal location for training the model. This selection ensures that the model is trained on relevant data, improving its ability to detect wildfire-related features and improve the accuracy of fire detection.

A key challenge addressed in this chapter involves ensuring the clarity and usability of the satellite imagery, particularly in pairing GOES ABI and VIIRS datasets. For an image to be suitable for model training, one of the three VIIRS satellites must have captured imagery over Maui, Hawaii, and the corresponding GOES ABI image must align temporally (on the same year, day, and time) and spatially (with the closest time match). This temporal and spatial alignment is important for ensuring consistency between the two datasets, allowing the model to learn meaningful features across both satellite systems.

This chapter provides an overview of the rationale behind the selection of datasets and details the preprocessing techniques used to enhance the model's performance. Key preprocessing steps include spatial and temporal alignment of the datasets, converting radiance to brightness temperatures, and calculating the AFI. These techniques are designed to improve the overall quality and reliability of the input data, ultimately leading to more accurate model outputs.

Following the preprocessing stage, the FireGAN architecture is introduced. FireGAN comprises two primary components, being a generator and a discriminator. The generator is designed to enhance the spatial resolution of GOES ABI input imagery, improving it from 2250 m to 375 m. While the discriminator is trained to differentiate between real (high-resolution) and generated (super-resolved) images, thereby refining the generator's outputs through adversarial learning. After that, the feature extraction and loss functions are presented in relation to the adversarial training objective. These include pixel-wise loss, content loss, and adversarial loss, all of which guide the model toward generating high-quality, realistic outputs.

The training process focuses on optimizing both the generator and discriminator. The Adam optimizer is used to ensure stable convergence. Loss functions and evaluation metrics are integrated into the training loop to assess the training progress for both the generator and discriminator, so the Adam optimizer can adjust for improved performance. To evaluate the performance of the model, both image-based and classification-based metrics are employed. Image quality is assessed using metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Root Mean Square Error (RMSE), and r^2 , all of which provide insights into the visual accuracy of the generated images. Additionally, fire detection performance is evaluated using metrics such as Heidke Skill Score (HSS), Critical Success Index (CSI), Probability of Detection (POD), and False Alarm Ratio (FAR), which are used to assess the fire confidence's ability to correctly identify hotspots.

Finally, the chapter discusses the calculation of fire confidence scores, which quantifies the likelihood that each pixel in the output image represents a fire based on the model's predictions. By providing a confidence score for each pixel, fire locations are clearly visually highlighted.

3.2 Data Processing and Selection

3.2.1 Data Pairing

Data pairing is important in this research as there are images every 16 days from one of the three VIIRS satellites that covers Maui, HW. So, there needs to be a corresponding GOES ABI image within the same day and time period as the VIIRS images. Fortunately, since the GOES ABI temporal resolution is 10 minutes, there is only a small time delay (< 5 minutes) in between the paired images. This temporal alignment between the GOES ABI and VIIRS datasets ensures that both fire and non-fire images observed by the VIIRS imager are accurately matched with imagery from GOES ABI, providing consistent data for training the model.

To accurately determine the orbit paths and timing of the satellite overpasses within the study domain, the Fire Information for Resource Management System (FIRMS) [25] was utilized. FIRMS provides visualizations for satellite orbit tracks, including capture times in Coordinated Universal Time (UTC), which are essential for identifying when each VIIRS satellite passes over Maui. This tool determined the initial overpass dates and times for 2023 and 2024, from which additional dates were determined using the 16-day revisit cycle for each satellite.

By aligning GOES ABI imagery with these predetermined VIIRS imager timestamps, GOES ABI data was acquired at matching 16-day intervals and at the closest possible capture times. Due to GOES ABI's temporal resolution, this process allowed for flexible selection of corresponding images. As a result, for every VIIRS observation over Maui, there exists a paired GOES ABI image captured at nearly the same moment, forming a consistent dataset for training the FireGAN model.

Another important consideration for data downloading is the initial dates that Suomi-NPP, NOAA-20, and NOAA-21 data was made available. Data availability via LAADS DAC coverage dates are as follows:

- Suomi-NPP: January 20th, 2012 - Present
- NOAA-20: January 1st, 2020 - Present

- NOAA 21: January 17th, 2024 - Present

These bullet points represent the each satellite with their respective coverage dates. Most importantly of the three, NOAA-21 provides a limited number of images that are relevant to this work since the sample time is 2023 and 2024. So the data downloading was adjusted to accommodate for NOAA-21's more recent launch time.

3.2.2 Active Fire Index

A key component of this research is the calculation of the Active Fire Index (AFI) [12], which is given by the following formula:

$$\text{Active Fire Index (AFI)} = 100 \times \frac{\text{MWIR} - \text{LWIR}}{\text{MWIR} + \text{LWIR}} \quad (3.1)$$

AFI normalizes temperature differences, making active fire pixels more distinguishable from the background [12]. MWIR bands, such as VIIRS I04 and GOES ABI C07, that capture fire hotspots specifically, along with LWIR bands, like VIIRS I05 and GOES ABI C14, measure cooler background temperatures. This ratio effectively highlights fire intensity, where higher AFI values indicate stronger fire activity. By reducing background noise and normalizing temperature variations, the AFI enhances fire classification accuracy, making it a valuable feature for the model in identifying wildfire hotspots.

An example of the effectiveness of the AFI calculation can be seen in the following plot:

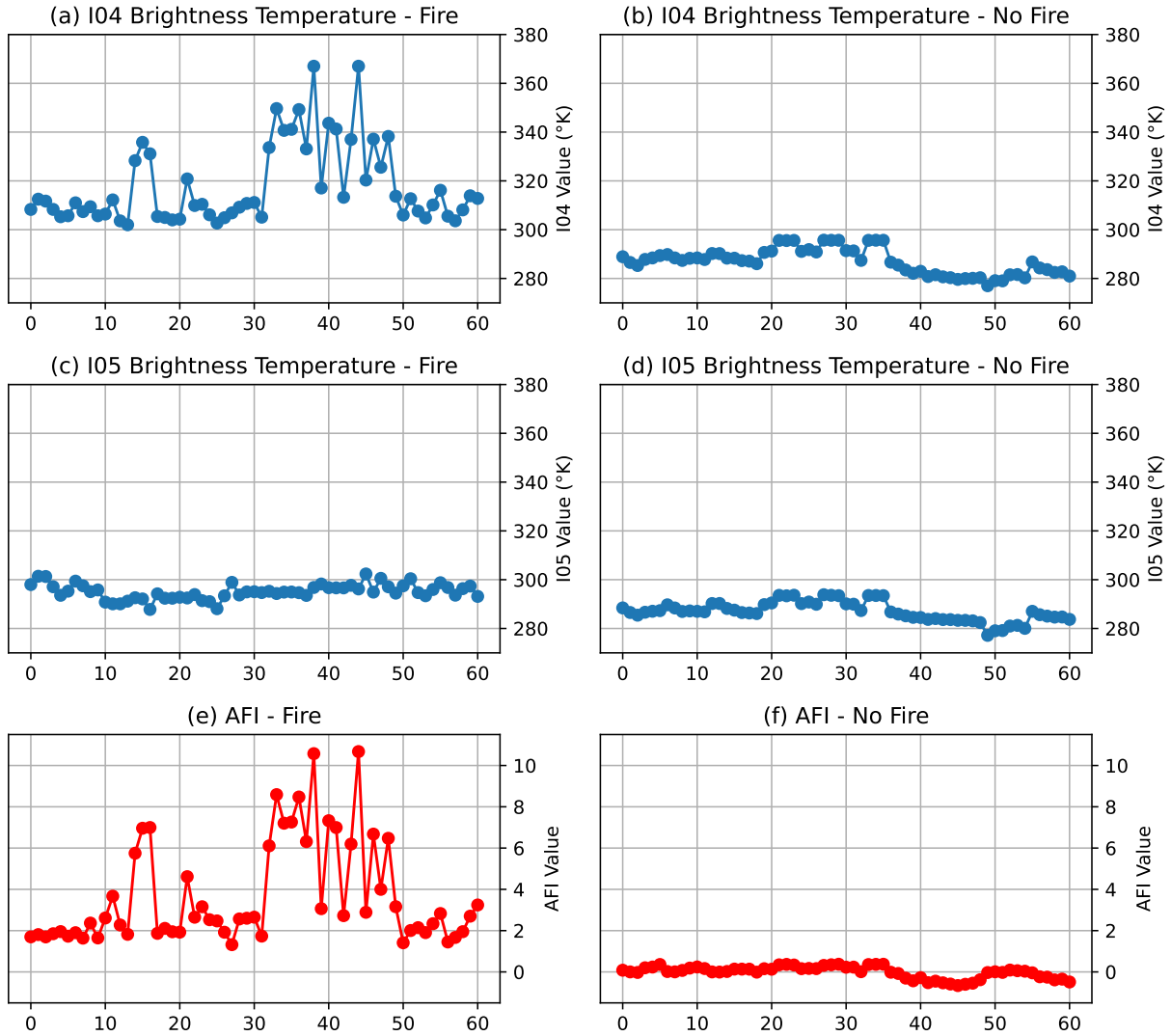


Figure 3.2.1: Pixel Values from Fire and Non-Fire Regions on August 9th, 2023 (12:12 UTC): (a) VIIRS I04 Brightness Temperature in Fire Region, (b) VIIRS I04 Brightness Temperature in Non-Fire Region, (c) VIIRS I05 Brightness Temperature in Fire Region, (d) VIIRS I05 Brightness Temperature in Non-Fire Region, (e) VIIRS AFI Calculation in Fire Region, (f) VIIRS AFI Calculation in Non-Fire Region

This plot illustrates the contrast in brightness temperature and AFI values between fire and non-fire pixel values. As shown in the first column, pixels from active fire regions exhibit significantly higher brightness temperatures in the I04 band in comparison to non-fire pixels in the second column. Which again shows the value of using the I04 and C07 spectral bands for their high sensitivity to wildfire hotspots as previously stated. Additionally, the AFI values in fire-affected areas are notably elevated, creating a clear distinction between burning and non-burning regions.

These results highlight the effectiveness of the AFI calculation in distinguishing active fire pixels from satellite imagery.

3.2.3 Data Processing

When processing the raw GOES ABI data, the .nc files are sampled for a 2.25 km resolution instead of the original 2 km resolution. This is because the final output from the model will achieve a 375 m spatial resolution and by sampling GOES ABI at a lower resolution will allow for a 6x increase in resolution. This simple integer scale of 6 is ideal for the models efficiency in training. This is also where the brightness temperature conversion shown in 2.4 is applied. Similarly the AFI, specified in 3.1 is calculated for each image as well. After sampling, there is an output array following the dimensions (27, 35) pixels for C07, C14, and AFI.

Next, the VIIRS images aboard each of the three satellites are processed by matching the files containing brightness temperatures to the geolocation files. From there, the brightness temperature data is mapped to the correct latitudes and longitudes using the geolocation files. Once the brightness temperatures are properly geolocated, the AFI values for VIIRS images are calculated using the same methodology as applied to GOES ABI. Last, the resulting VIIRS array dimensions are (162, 210) for I04, I05, and AFI.

After both GOES ABI and VIIRS imager datasets are prepared, they are paired based on their corresponding timestamps and spatial locations then saved as an .h5 file format for both GOES ABI and VIIRS imager along with their corresponding spectral bands and AFI calculations.

3.2.4 Data Selection

To filter and remove poor-quality images from the dataset, the VIIRS images were resized to match the resolution of the GOES ABI images. The following correlation coefficient function was then calculated for each image:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (3.2)$$

Where: The correlation coefficient calculation is calculated for every pair of GOES ABI and reduced-resolution VIIRS images across the MWIR, LWIR, and AFI bands. The resulting correlation coefficient r will be a value between 0 and 1, where $r = 1$ indicates a perfect correlation, and $r = 0$ indicates no correlation.

Images with a correlation coefficient ≥ 0.6 were retained in the dataset. Resulting in 33 training and 26 testing files containing MWIR, LWIR, and AFI for both GOES ABI 2.25 km and VIIRS 375 m categories, both containing fire images.

3.3 Processed Data Examples

The following is an example of the processed GOES ABI and VIIRS image during the Maui wildfires:

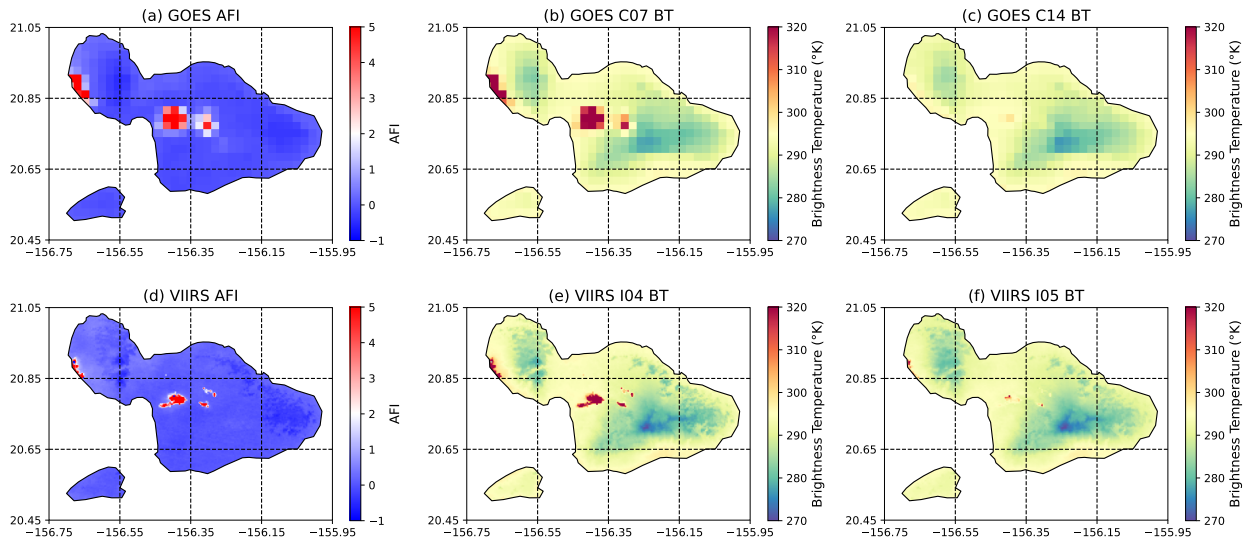


Figure 3.3.1: August 9th, 2023 (12:12 UTC): (a) GOES ABI AFI, (b) GOES ABI C07, (c) GOES ABI C14, (d) VIIRS AFI, (e) VIIRS I04, (f) VIIRS I05

Figure 3.3.2 illustrates paired VIIRS sensor and GOES ABI images. The VIIRS data was captured by the NOAA-20 satellite at 12:12 UTC and paired with a GOES ABI image captured at 12:10 UTC. There is a slight temporal offset but the difference of less than two minutes so the difference

is negligible and does not significantly impact the observed fire activity. This demonstrates the effectiveness of temporal alignment between the two datasets for capturing accurate images.

Similarly, the following is an example of the processed GOES ABI and VIIRS sensor images with their respective AFI calculations over Maui, HW, but on a day without wildfire:

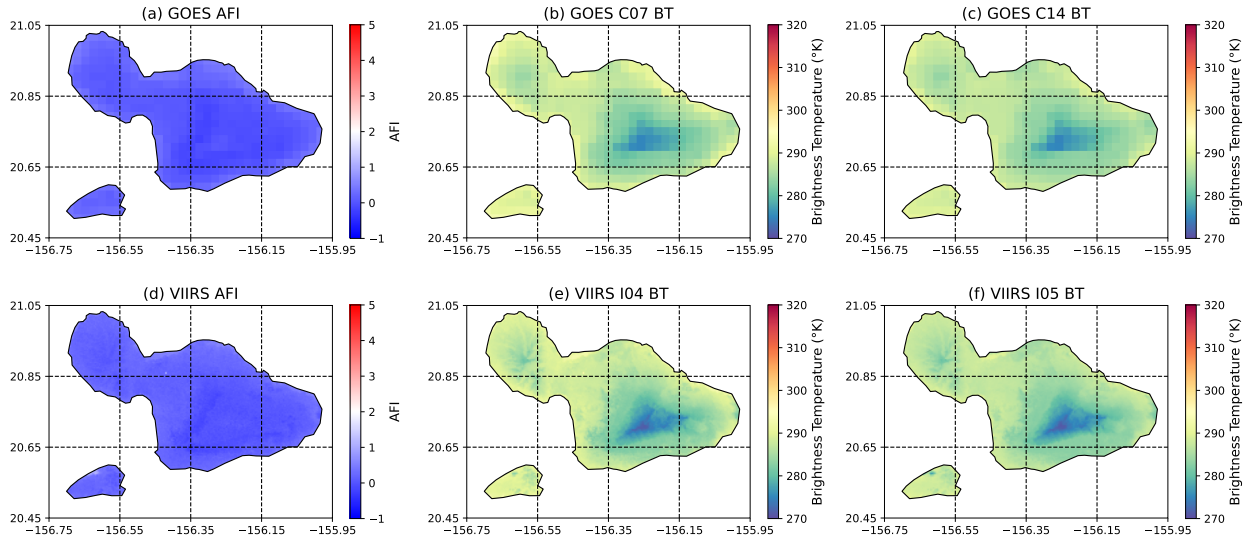


Figure 3.3.2: March 5th, 2023 (12:12 UTC): (a) GOES ABI AFI, (b) GOES ABI C07, (c) GOES ABI C14, (d) VIIRS AFI, (e) VIIRS I04, (f) VIIRS I05

This is an example of image pairing as the VIIRS sensor images were captured from the Suomi-NPP satellite at 12:12 UTC and paired with the GOES ABI images captured at 12:10 UTC, similar to the previous image. This shows the dataset represents both fire and non-fire conditions for training the model.

3.4 Deep Learning For Super-Resolution Fire Detection

3.4.1 Introduction to Generative Adversarial Networks

This research utilizes a Generative Adversarial Network (GAN), a type of generative model that has become increasingly popular in the field of deep learning. GANs are widely recognized for their ability to generate realistic outputs, with popular services such as ChatGPT and Google Gemini often cited as prime examples. While these models have recently gained mainstream attention

due to their ability to generate coherent and contextually appropriate responses, the foundations of GANs were laid through a long history of earlier models and techniques that contributed to their development. Some of the most influential predecessor models that laid the groundwork for this research include:

- 1950 Turing test was proposed by Alan Turing. The Turing Test is a measure of a machine's ability to exhibit intelligent behavior indistinguishable from that of a human. If a human interrogator cannot distinguish between human and machine responses in a conversation, the machine is considered to have "passed" the test due to its ability to "parrot" ie. replicate a human response to a conversation. This laid the foundational principals for artificial intelligence development [26].
- 2012 Neocognitron (Precursor to CNN) created by Kunihiro Fukushima resulted in important concepts like using convolution in a neural network, along with pooling and feature extraction. This is because with convolution the neural network can extract visual features such as edges and borders, along with that it was able to learn at various levels of an image. This was a fundamental in later breakthroughs such as AlexNet (2012) which leveraged the fundamentals placed by the Neocognitron and created a deep convolutional network to automatically learn hierarchical features from raw image pixels, resulting in improved performance for image classification [27].
- 1983 Boltzmann Machine was developed by Geoffrey Hinton and Terry Sejnowski. the Boltzmann Machine is a type of stochastic recurrent neural network used for learning probability distributions over input data. The input data was initially a set of binary vectors which the Boltzmann Machine would generate probabilities for by adding weights which has been a fundamental of artificial intelligence. In turn, this was one of the earliest neural networks [28].
- 1990 Adversarial Artificial Curiosity (AC) from Jürgen Schmidhuber. This started as an early application of an unsupervised neural network to seek out novel points of data. Which

resulted in artificial curiosity as the system itself would determine patterns within a dataset. It was also rewarded when it had generated an error-minimizing result which is a cornerstone in GANs moving forward as the generators must fool the discriminators [29].

- 2014 Generative Adversarial Network (GAN) introduced by Ian Goodfellow et. al. which consist of two neural networks being a generator and a discriminator. The generator creates synthetic data (such as images, text, etc.), while the discriminator tries to distinguish between real and fake data. These networks are trained adversarially i.e., the generator tries to fool the discriminator, while the discriminator attempts to accurately identify fake data. This adversarial process leads to the creation of very realistic synthetic data, generally based on several loss calculations to determine their individual accuracies. The GAN is one of the most used forms of artificial intelligence by the general public today [30].

For this application, the use of AI for super-resolution is particularly effective due to GANs' ability to reconstruct fine details from low-resolution imagery, making them ideal for enhancing satellite data in environmental monitoring and wildfire detection tasks. Unlike traditional interpolation techniques such as bicubic interpolation, which tends to produce overly smooth images lacking sharp features unlike GAN-based super-resolution methods can generate fine-grained details, leading to more accurate reconstructions of images [31]. This is especially important for highlighting features such as fire boundaries, atmospheric conditions, and geographic characteristics.

The GAN framework enables the model to learn both pixel-level and perceptual-level representations by comparing generated outputs to high-resolution ground truth data. This style of learning allows the generator to create visually coherent and highly detailed images from coarse GOES ABI image inputs, significantly enhancing the quality of extracted features [32].

GANs operate through an adversarial training process, where a generator network produces high-resolution synthetic data, while a discriminator network attempts to distinguish between the generated and real images. This competition compels the generator to continuously refine its outputs, resulting in increasingly realistic imagery. The adversarial nature of GANs acts as an effective quality control system, reducing discrepancies and ensuring high-quality reconstructions.

Throughout the training process, both networks improve simultaneously as the generator becomes more proficient at producing realistic images, while the discriminator becomes better at identifying subtle differences between real and synthetic data. This feedback loop is a key strength of GAN-based super-resolution, as it facilitates the generation of accurate and reliable imagery [32]. Such precision is important for downstream tasks like wildfire detection, where preserving fine details can significantly impact detection performance.

Without the adversarial component, other deep learning based methods such as the Super-Resolution Convolutional Neural Network (SRCNN) can produce smoother images may not properly extract features from low resolution images in comparison to their high resolution counterparts [33]. This is why the GANs adversarial quality control method is important in generating realistic synthetic data.

To apply these fundamentals to this research, the spatial and temporal resolutions of VIIRS sensor and GOES ABI imagers pose a significant problem for frequent wildfire tracking. As previously stated, the GOES ABI imager does not offer high enough spatial resolution for wildfire tracking, but does offer an ideal temporal resolution. While VIIRS sensor offers an ideal spatial resolution, but the VIIRS satellites do not revisit the study domain frequently enough for reliable wildfire tracking.

To overcome this trade-off, the Enhanced Super-Resolution Generative Adversarial Network (which will now be referred to as FireGAN) partially based on the architecture proposed by [34], a variant of a traditional GAN is used for super-resolving images. In FireGAN, the generator network enhances the spatial resolution of GOES ABI imagery to match that of VIIRS imager, while the discriminator learns to distinguish between ground-truth VIIRS images and the generated high-resolution GOES ABI images. The objective is to produce GOES ABI images that are visually and statistically comparable to VIIRS, thereby preserving GOES ABI's high temporal resolution while gaining the spatial detail of VIIRS images.

This section will elaborate on the FireGAN's architecture and how it effectively enhances GOES ABI imagery to achieve a spatial resolution comparable to VIIRS imager while maintaining GOES

ABI's high temporal resolution. The methodology behind this enhancement will be explained, including the roles of the generator and discriminator in refining image quality. Additionally, a fire pixel confidence metric will be introduced, leveraging the enhanced resolution to improve the accuracy of active wildfire detection.

3.4.2 Generative Adversarial Network Fundamentals

As previously mentioned, the Generative Adversarial Networks (GANs) are composed of two networks, one being the generator and the other being a discriminator that work in opposition to each other, creating an adversarial environment that ultimately leads to the generation of high-quality synthetic data. This adversarial training process drives the generator to improve over time by learning to produce data samples that resemble real data as closely as possible.

The generator's performance is evaluated using a loss function that reflects how well it can "fool" the discriminator. This loss function plays an important role in training, as it provides a signal for the generator to adjust its parameters in order to generate more convincing data. Over time, as training progresses, the generator minimizes this loss, ideally reaching a point where the discriminator can no longer reliably distinguish the real and generated samples [35].

Mathematically, a GAN can be expressed as the following equation proposed by [36]:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (3.3)$$

Which represents the objective function of a GAN. The variables involved in this equation are explained as follows:

- D : The discriminator network.
- G : The generator network.
- $V(D, G)$: The value function of the GAN, representing the total loss.

- $\min_G$: The generator G aims to minimize the value function $V(D, G)$. This means the generator tries to generate data that reduces the discriminator's ability to distinguish between real and fake data.
- $\max_D$: The discriminator D aims to maximize the value function $V(D, G)$. This means the discriminator improves its ability to distinguish real data from generated data.
- $\mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)]$: The expected log-likelihood over real data. Here, x is a real data sample, $p_{\text{data}}(x)$ is the real data distribution, and $D(x)$ is the probability that D classifies x as real. This term represents the discriminator's success in classifying real data.
- $\mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$: The expected log-likelihood for fake data. Here, z is a random noise variable drawn from a prior distribution $p_z(z)$, and $G(z)$ is the generated data. $D(G(z))$ is the probability that the discriminator classifies the generated data as real. The term $\log(1 - D(G(z)))$ represents the success of the discriminator in classifying fake data as fake.

Essentially, the generator G aims to minimize the function $V(D, G)$ by producing realistic data. While the discriminator D aims to maximize the function $V(D, G)$ by accurately distinguishing between real and fake data. Through this adversarial training both networks improve, ultimately resulting in a generator capable of producing realistic synthetic data.

3.4.3 Convolutional Neural Networks

Another important concept in image downscaling, along with the GAN calculations, is the use of Convolutional Neural Networks (CNNs). Since each image needs to be processed at different scales and resolutions, CNNs are ideal for capturing hierarchical spatial patterns and features. The convolution operation, which is a fundamental component of CNNs, can be described mathematically as follows [37]:

$$f[i, j] = \sum_{x=1}^{m_1} \sum_{y=1}^{m_2} K_{[x, y]} I_{[i+x-1, j+y-1]} \quad (3.4)$$

This equation represents the 2D convolution operation used in CNNs where:

- $f[i, j]$ is the output value at position (i, j) in the resulting image.
- I is the 2D input image, with its spatial dimensions denoted by i and j .
- K is the 2D kernel or filter, with its dimensions defined by m_1 and m_2 , which represent the size of the kernel in the x and y directions, respectively.
- x and y are the indices used to sum over the kernel's dimensions.
- The operation is a sliding window, where $K[x, y]$ is the kernel value at position (x, y) , and $I[i + x - 1, j + y - 1]$ is the corresponding image value at position $(i + x - 1, j + y - 1)$.

This convolution operation is repeated across the entire image by shifting the kernel based on the specified stride value. As the kernel slides over the image, it generates a feature map, which is a new representation of the image emphasizing specific learned features such as edges, textures, and shapes. Each filter in the CNN produces a different feature map, allowing the network to learn and detect various patterns from the input image. By stacking multiple layers of filters, CNNs enable the model to capture both simple local patterns (e.g., edges and textures) in the initial layers and increasingly complex structures in the deeper layers. This hierarchical learning of spatial features at multiple scales makes CNNs effective for image processing tasks. In the case of the FireGAN, this ability to learn and preserve fine details is important for generating high-quality images that accurately depict fires and fire boundaries.

3.5 Enhanced Super-Resolution Generative Adversarial Network Architecture

The image below illustrates the FireGAN architecture, which incorporates foundational principles from both Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs), as previously discussed. This architecture will be further explained in the following sections.

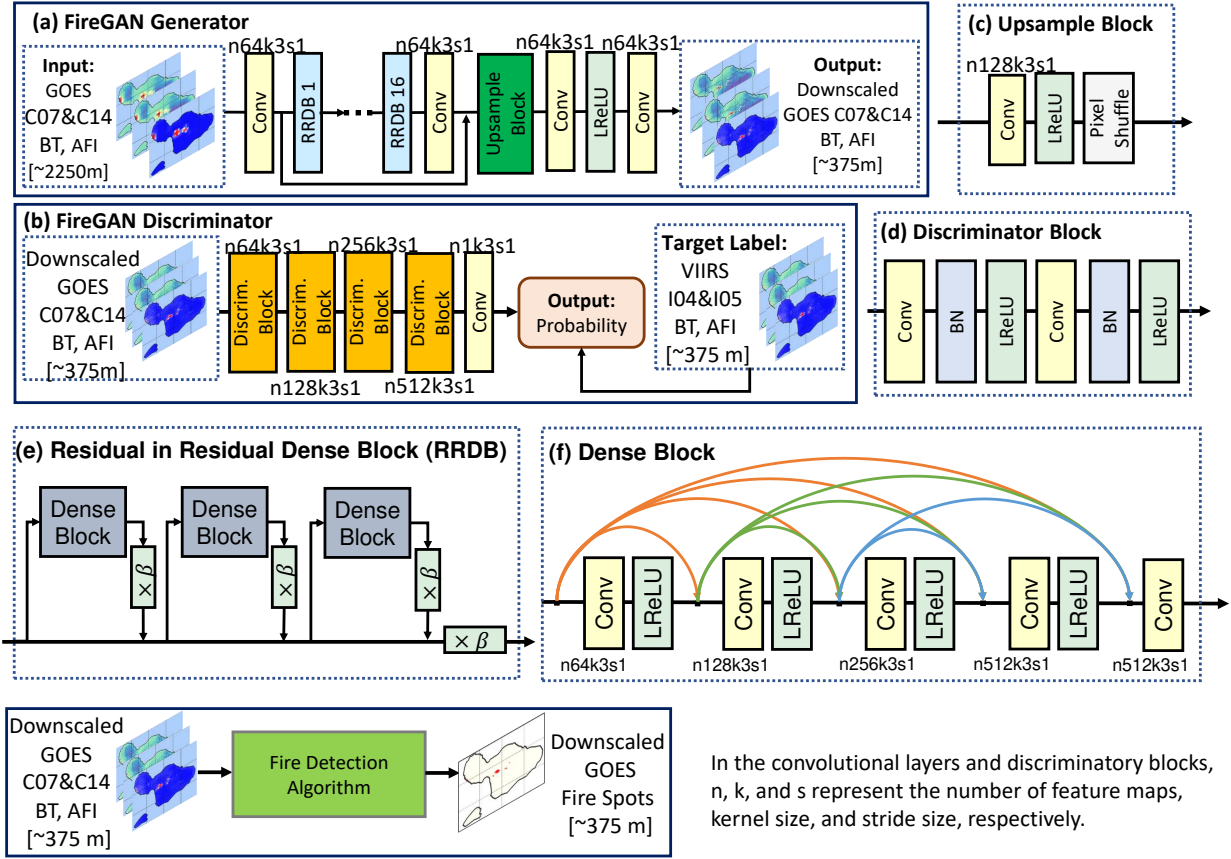


Figure 3.5.1: FireGAN Architecture

This full architecture will be explained in the following sections.

3.5.1 Dense Residual Blocks

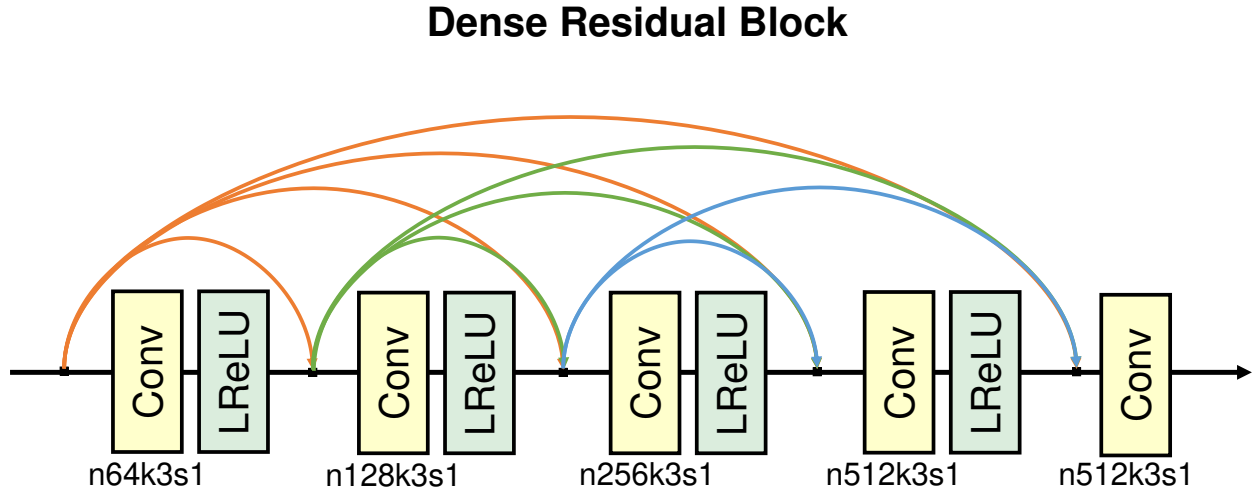


Figure 3.5.2: Dense Residual Block Architecture: n, k, and s represent the number of feature maps, kernel size, and stride size, respectively

Inspired from [38] the DenseResidualBlock shown in 3.5.2, enhances feature extraction by leveraging densely connected convolutional layers, which is helpful in information reuse, gradient flow, and residual learning. It consists of five convolutional layers, where each layer receives as input not only the output of the previous layer but also the concatenated outputs of all previous layers. This dense connectivity preserves fine details which are highly important for image reconstruction. To expand further on the benefits of concatenation:

- Features learned in earlier layers remain accessible throughout the block. For example, a feature detected in layer 2 may be less pronounced in layer 3, but the dense connections ensure it remains available in later layers.
- Gradient flow in concatenation is improved during backpropagation, as it allows the network to propagate gradients across a deeper architecture more effectively. This helps mitigate issues like the vanishing gradient problem [38].

The convolutional layers use a 3×3 kernel, stride of 1, and padding of 1, maintaining spatial dimensions. The final output is scaled by a residual scaling factor $\text{res_scale}=0.2$, stabilizing training and preventing gradient explosion.

Also, each convolutional layer except the last, is followed by a Leaky Rectified Linear Unit (LeakyReLU) activation to introduce non-linearity. Along with that, LeakyReLU allows negative slopes for negative inputs which prevents a "dying ReLU" problem where neurons stop or "die" due to negative input values [39], which is important for retaining important features as data. The increasing number of feature maps, also known as local feature fusion, in each block (e.g., 64 feature maps per block) introduces progressively higher levels of complexity to the convolutional layers [38]. This growing complexity allows the model to learn increasingly detailed features within the images, which is important for refining low-resolution inputs into high-resolution outputs. By adding seemingly minute details to the low-resolution input, the network can generate more accurate, visually realistic high-resolution VIIRS imager outputs. This process enhances the model's ability to make small adjustments to improve the accuracy of the generated images.

For the block definition, let the input to the block be x and the number of feature maps in each convolutional layer be f_i , where $i \in \{1, 2, 3, 4, 5\}$.

For each block b_i , the equation is:

$$\text{Block}_i(x) = \text{Conv}(x, f_i) + \alpha_i \quad (3.5)$$

- $\text{Conv}(x, f_i)$ represents the convolution operation applied to the input x with along with the number of feature maps (64) multiplied by each block (b_x).
- α_i represents the activation function:
 - $\alpha_i = \text{LeakyReLU}(x)$ for blocks b_1, b_2, b_3, b_4
 - For block b_5 , the activation function is omitted ($\alpha_5 = 0$).

Let the input to the network be x , and let the output of each block be denoted as Out_i . The input to each block includes a concatenation of the original input x and the outputs of all previous blocks:

$$\text{Out}_i = \text{Block}_i \left(\sum_{j=1}^{i-1} \text{Out}_j + x \right) \quad \text{for } i \in \{1, 2, 3, 4, 5\} \quad (3.6)$$

Where the $\sum_{j=1}^{i-1} \text{Out}_j$ represents the concatenation of all previous outputs. This concatenation allows each block to access and reuse features learned in earlier blocks, promoting better gradient flow and richer feature learning as mentioned previously.

The final output is obtained by adding a scaled residual connection:

$$y = \text{Out}_5 \times \text{res_scale} + x \quad (3.7)$$

- Out_5 is the output from the last block.
- Scaling Factor $\text{res_scale} = 0.2$
- x is the original input image, which is added back to the output after scaling.

These equations represent the flow of data through the DenseResidualBlock and describe the functionality of the residual dense network module. This is a fundamental step to achieve a super-resolution output as the DenseResidualBlock learns and reuses features from various block levels ($b_1 \rightarrow b_5$), also known as global residual learning [38]. By aggregating features across multiple levels, the network can capture a wide range of hierarchies from each of the feature maps. The concatenation of outputs from previous layers ensures that the network retains important fine details, textures, and subtle features that might otherwise be lost in a traditional deep learning model. This capability is particularly important for tasks like super-resolution, where the model needs to upscale images while preserving intricate structures and high-frequency details. As a result, the super-resolved outputs generated by the DenseResidualBlock is important for allowing the network to learn and recover the fine details, for tasks such as fire detection in the FireGAN framework.

3.5.2 Residual In Residual Dense Block

Residual in Residual Dense Block (RRDB)

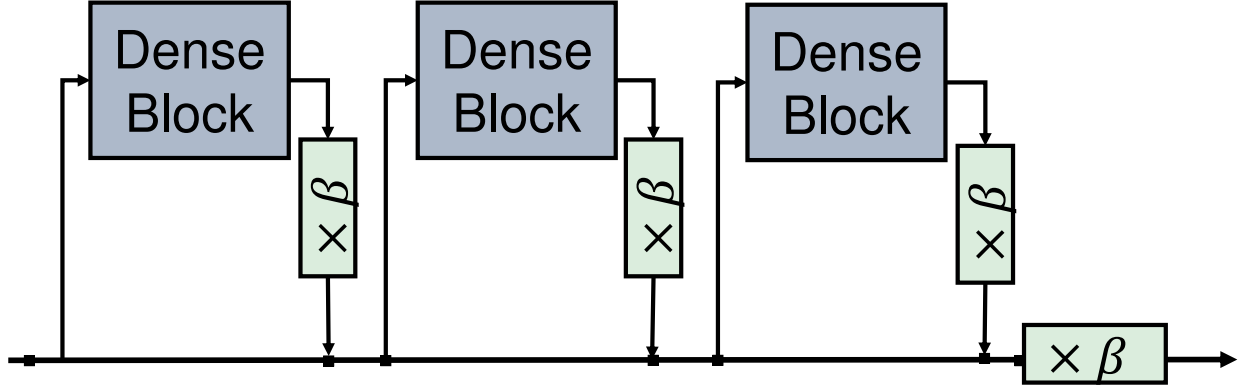


Figure 3.5.3: Residual In Residual Dense Block Architecture: Where n is the number of feature maps, k is the kernel size, and s is stride

Next, the residual in residual dense block, builds upon the dense residual block architecture by stacking three operations of three dense residual block, inspired from [34]. Each of the three dense blocks are processed from the input x and added sequentially, and their outputs are summed together. This combined output is then passed through a residual connection, scaled by residual scale = 0.2, similar to the first module. The operation can be mathematically expressed as:

$$y = \beta_n \cdot (f_1(x) + f_2(x) + f_3(x)) + x \quad (3.8)$$

Where β_n is the residual scaling parameter and $(f_1(x) + f_2(x) + f_3(x))$ are the outputs of the first, second, and third dense residual block, respectively, with x denoting the input feature map. Each of these three blocks retains the same internal structure (e.g., convolutional layers, LeakyReLU activation), but they operate independently and produce distinct outputs.

This cascading design allows the network to model more complex and abstract features by capturing hierarchical representations across residual dense blocks as it adds three multi-level residual

networks [34]. The residual connections within each dense residual block, combined with the overarching residual connection across the entire RRDB, facilitate gradient flow and improve training stability. This architecture accelerates convergence, meaning the model learns to the best of its ability. Especially when super-resolving GOES ABI imagery where preserving both fine details and global image structure is essential.

The nested residual structure of RRDB significantly enhances the model's ability to learn high-frequency as many details possible from the inputted GOES ABI images. As a core component of FireGAN, RRDB enables three levels making a deep network in total [34].

3.6 GeneratorRRDB

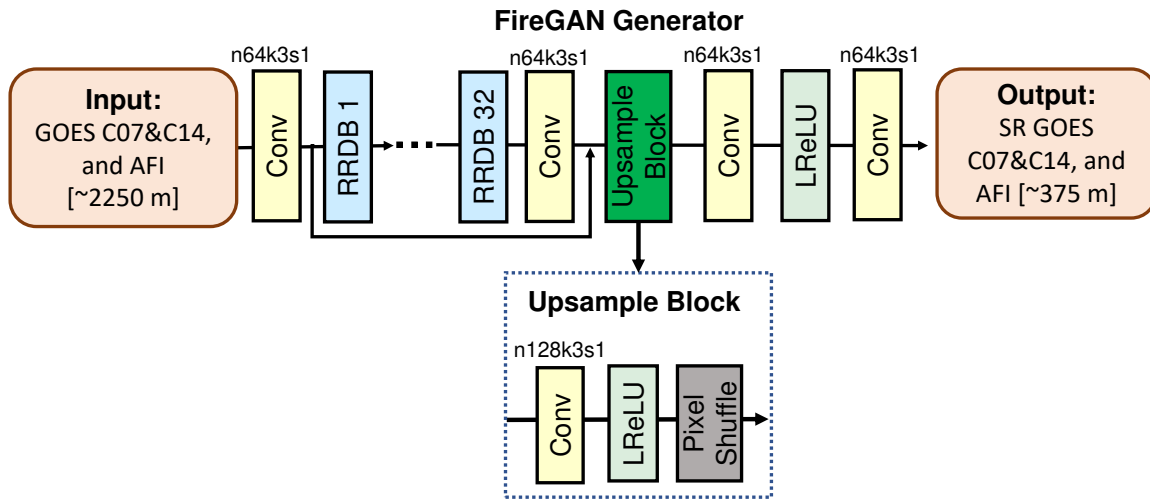


Figure 3.6.1: Generator Architecture: Where n is the number of feature maps, k is the kernel size, and s is stride

The generator is where this research comes together, because this is where the low resolution GOES ABI images are enhanced to meet the resolution of VIIRS images. The purpose of this generator is to use the Dense Residual Blocks along with the Residual In Residual Dense Blocks to learn the GOES ABI images to increase their resolution by 6x.

The generator takes an input tensor x of shape (B, C_{in}, H, W) , where $B = 1$ represents a single training image (out of 33) treated as a batch of size one, $C_{\text{in}} = 3$ is the number input channels (C07, C14, and AFI). Last, the spatial dimensions being $H = 27$ and $W = 35$.

1. Initial Convolution Layer: The first operation applied to the input image x is a convolutional layer with kernel size 3×3 , stride 1, and padding 1. This layer outputs a feature map of shape $(B, 64, 27, 35)$, where 64 is the number of filters applied during this operation:

$$\text{out1} = \text{Conv2D}(x, 64, 3, 1, 1) \quad (3.9)$$

This initial layer extracts low-level features such as edges and textures. These features form the foundation that later layers use to identify more complex patterns, such as wildfire signatures, cloud structures, and land cover boundaries.

As with the convolutional layers used throughout the Dense Residual Blocks and Residual-in-Residual Dense Blocks, this layer employs a kernel size of 33, stride 1, and a LeakyReLU activation function.

Formally, the output tensor can be described as:

$$\text{out}_1 \in \mathbb{R}^{B \times 64 \times H_{\text{in}} \times W_{\text{in}}} \quad (3.10)$$

Again, where B is the batch size per calculation, 64 is the number of filters which generate 64 feature maps, and H_{in} and W_{in} are the height and width of the input image (27, 35). The feature maps in out_1 will capture low-level features (edges, textures, etc.) from the original images.

2. Residual in Residual Dense Block: The output from the initial convolutional layer is passed through a series of 32 Residual-in-Residual Dense Blocks (RRDBs). Each RRDB consists of a sequence of convolutional layers with LeakyReLU activations, allowing the model to learn hierarchical features from low-level edges to high-level patterns as specified in previously. Additionally,

skip connections within and across blocks facilitate residual learning, allowing the network to focus on learning the difference between the input and target.

The forward propagation through the RRDB stack can be mathematically described as follows:

$$x_0 = x \quad (3.11)$$

Where, x is the input feature map to the generator, and x_0 is the input to the first RRDB block.

$$x_i = x_{i-1} + F_{\text{RRDB}}(x_{i-1}), \quad \text{for } i = 1, 2, \dots, 32 \quad (3.12)$$

Each RRDB block takes the output of the previous block x_{i-1} , processes it through the RRDB pipeline as previously described F_{RRDB} , and adds it back to the input in a residual manner.

$$\hat{y} = x_{32} = x_0 + \sum_{i=1}^{32} F_{\text{RRDB}}(x_{i-1}) \quad (3.13)$$

The final output $\hat{y}$ is the result of the input x_0 combined with the accumulated residuals from all 32 RRDB blocks. This structure enables the model to capture both low-level and high-level features in a hierarchical manner. As such, this stage is important for the super-resolution process, allowing the model to progressively learn fine details through the cascading forward propagation of each RRDB layer.

3. Second Convolution Layer and Skip Connection: The output from the residual blocks is passed through a second convolutional layer, which again encapsulates the RRDB functions within a convolutional neural network function as following:

$$x_{\text{res}} = \text{Conv2D}_2 \left(x + \sum_{i=1}^{32} F_{\text{RRDB}}(x_{i-1}) \right) \quad (3.14)$$

$$x_{\text{out}} = x_{\text{res}} + x \quad (3.15)$$

This second convolutional layer acts as a transitional stage which consolidates the features generated by the RRDB blocks. Here, it distills and preserves the most informative aspects from the large quantity of features extracted during the residual learning process.

Following the convolution, the result is added back to the shallow features extracted by the first convolutional layer ($Conv_1$) via a skip connection. This residual addition ensures that both low-level features and high-level hierarchical representations are retained in the output, promoting more diverse feature representations during training.

By applying the same 3×3 convolution, this layer filters and compresses the enriched RRDB features into a coherent representation and aligns them with the earlier feature map, thereby enhancing the model's ability to learn mappings from low-resolution to high-resolution space.

4. Upsampling (PixelShuffle): The output is then passed through an upsampling block using the Pixel Shuffle operation, which is responsible for increasing the spatial resolution by 6x. The upsampling consists of the following three stages:

First, a convolutional layer is applied to the feature map. To enable a PixelShuffle operation with an upscale factor of 6, the number of output feature maps must be multiplied by the desired up-sampling scale ($6^2 = 36$). So the convolution function for upsampled Features has a shape of $(B, 36 * 64, H, W) = (B, 2304, 27, 35)$:

$$\text{Upsample Features} = \text{Conv2D}(B, 2304, 3, 1, 1) \quad (3.16)$$

From there, a LeakyReLU activation function is added:

$$x_{\text{relu}} = \text{LeakyReLU}(\text{Upsample Features}) \quad (3.17)$$

The final step in the upsampling block is the PixelShuffle operation, which rearranges the channel dimensions into higher spatial dimensions, effectively increasing the resolution of the feature map. The upscaling factor is 6, meaning that both the height and width of the feature map are increased by a factor of 6. The PixelShuffle operation works by first applying a convolutional layer that

increases the channel depth by a factor of r^2 , where r is the upscaling factor. In this case, $r = 6$, so the number of channels is increased by $6^2 = 36$. Then, PixelShuffle redistributes these extra channels into higher spatial dimensions, taking the 2304 feature maps (i.e., $64 \text{ filters} \times 36$) and reshaping them into 64 channels, each spatially upsampled by a factor of 6 in both height and width. The final equation for this operation is:

$$x_{\text{upsample}} = \text{PixelShuffle}(\text{LeakyReLU}(\text{Upsample Features})) \quad (3.18)$$

The resulting shape is now $(B, 64, 6H, 6W)$ after the Pixel Shuffle operation. It is important to note that the resulting shape returns from 2304 to 64 feature maps as the pixel shuffle operation then takes the 2304 feature maps, then redistributes them back into 64 channels, but each channel is spatially upsampled by a factor of 6 in both height and width. So, Each feature map is now 6×6 times the size of the original spatial resolution, both in height and width.

To summarize, the output feature maps from the RRDBs are passed through an upsampling block, which begins with a convolutional layer designed to increase the channel depth. This is followed by a LeakyReLU activation that introduces non-linearity, allowing the model to learn more complex features. The final step in the upsampling process is the PixelShuffle operation. This operation rearranges the feature maps into higher spatial dimensions, effectively increasing the resolution of the feature map. The upscaling factor is 6, meaning that both the height and width of the feature map are increased by a factor of 6.

Thus, the number of filters in the final output is 64, as the channels are redistributed into 64 output feature maps, with each map now representing a higher-resolution version of the original image. The pixel shuffle operation is designed to redistribute the extra channels spatially rather than increasing the number of output channels, making it an efficient method for performing super-resolution while preserving important features learned by the RRDB.

5. Final Convolution Layer: After the upsampling process, which increases the spatial resolution of the feature map by 6x, a final convolutional layer is applied to refine the upsampled features and produce the final output. This involves the following steps:

First, the refinement of upsampled features: after the feature map has been upsampled by the PixelShuffle operation, it may still contain noise or require further refinement to ensure smooth and coherent high-resolution details. The PixelShuffle operation increases the spatial resolution by redistributing the channels spatially, but this may introduce some artifacts or imperfections. The final convolution layer helps to adjust these features by applying a convolution with the same kernel size and stride as previously used, ensuring consistency and clarity in the output. This convolution operation acts as a filter, smoothing out any unwanted noise, sharpening fine details, and ensuring that the upsampled image appears natural and visually coherent. By learning from local spatial patterns, it helps eliminate any inconsistencies or roughness introduced by the PixelShuffle, producing a refined output that matches the intended high-resolution details. Similarly, the LeakyReLU activation function is applied again at the end of the convolutional function. This is shown by:

$$x_{\text{relu}} = \text{LeakyReLU}(\text{Conv2D}(x, \text{filters}, 3, 1, 1)) \quad (3.19)$$

Last, the final convolutional layer reduces the number of channels back to the original input of 3 channels, representing the GOES ABI C07 and C14 bands, along with the AFI calculation. This step brings the processed features back to the same format as the input data, allowing proper reconstruction of the output image. The equation for this final convolution operation is:

$$x_{\text{final}} = \text{Conv2D}(x_{\text{relu}}, \text{channels}, 3, 1, 1) \quad (3.20)$$

Summary of GeneratorRRDB: To summarize, the GeneratorRRDB section describes the architecture of the generator used for enhancing low-resolution GOES ABI images to match the resolution of VIIRS images which involves some key components:

- **Initial Convolution Layer:** The input tensor x of shape (B, C_{in}, H, W) with a batch size of 1 for each of the training images being generated, 3 input channels being GOES ABI C07, C14, and AFI, along with a height and width of $(27, 35)$, respectively. This is first processed through a convolutional layer with a kernel size of 3×3 , which extracts low-level features like edges and textures. The output tensor shape becomes $(B, 64, H_{in}, W_{in})$, where 64 feature maps are generated.
- **Residual in Residual Dense Block (RRDB):** The output of the initial convolution passes through 32 RRDBs, which refine the features through a series of convolutions and LeakyReLU activations. This enables hierarchical learning of the image features. The final output of this block is a combination of all features learned from low to high levels.
- **Second Convolution Layer and Skip Connection:** The output from the RRDBs is passed through a second convolutional layer and combined with the shallow features from the initial convolution using a skip connection. This step refines and consolidates the learned features.
- **Upsampling (PixelShuffle):** The output is upsampled by a factor of 6 using a PixelShuffle operation, which redistributes the feature maps into higher spatial dimensions, increasing the height and width of the feature maps by a factor of 6. This is a key operation for the generator as it increases the GOES ABI spatial resolution to meet that of VIIRS sensor.
- **Final Convolution Layer:** The upsampled feature maps are refined through a final convolutional layer, which reduces the number of channels back to 3 (representing GOES ABI C07, C14, and AFI), for the final high-resolution output.

Overall, this generator architecture combines convolutional layers, residual connections, and up-sampling to progressively enhance image resolution while preserving its original low-level features.

3.7 Discriminator

The Discriminator in a GAN is responsible for distinguishing real VIIRS measurements from GeneratorRRDB images. It acts as a binary classifier, predicting whether an input image is real (from the dataset) or fake (generated by GeneratorRRDB).

The discriminator works by progressively reducing the spatial resolution of the input image to extract high-level features that help distinguish between real and generated images. This process is done through a series of convolutional layers.

Similar to how the Generator increases the resolution of images by upsampling through layers, the discriminator does the opposite by downsampling the input image. Through each convolutional layer, the resolution (height and width) of the input image gets smaller, and the number of feature maps increases. This allows the network to focus on a large variety of features.

As both the real (VIIRS sensor) and fake (super-resolved GOES ABI) images are passed through the discriminator, it calculates how real or fake each image is during the forward pass. This information is used to compute the loss function, which indicates how well the Discriminator is performing its task. The Discriminator's loss is then used to update its weights, helping it become more proficient at distinguishing between real and generated images.

This adversarial process in training the FireGAN directly influences the generator's ability to learn and improve over time. Through continuous feedback from the discriminator, the generator refines its output, progressively generating more realistic super-resolved images. Over time, this iterative process helps the generator produce images that closely resemble real VIIRS measurements, enhancing the overall performance and realism of the model.

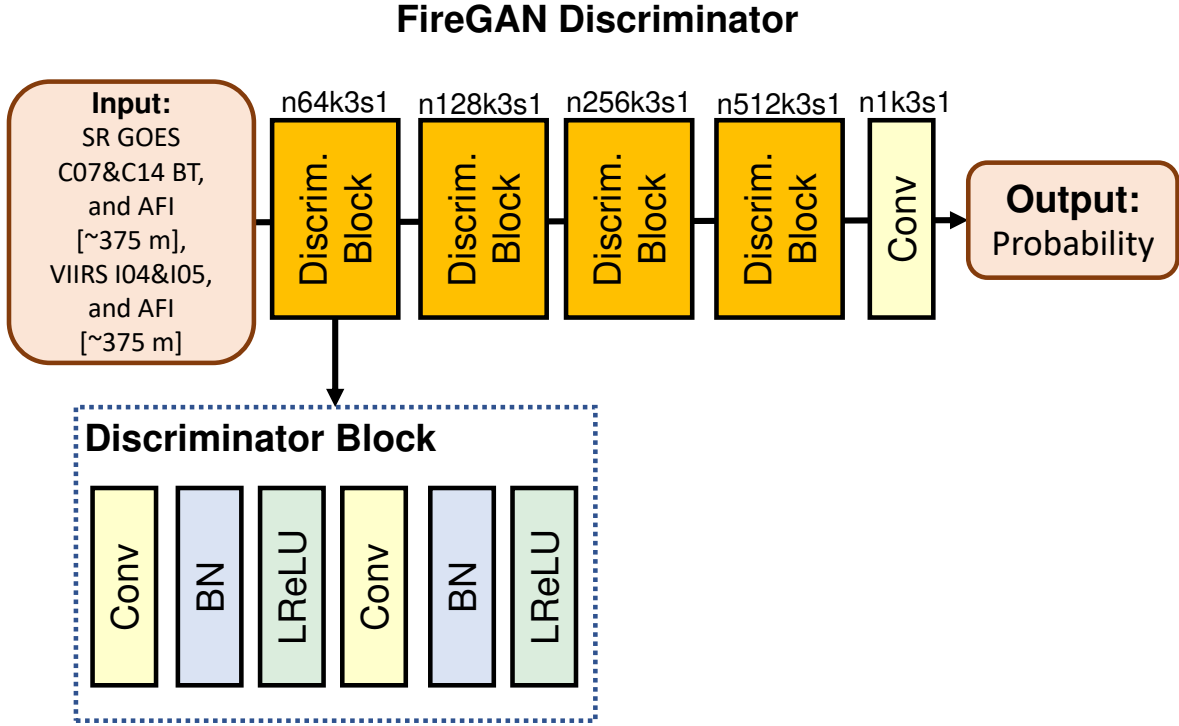


Figure 3.7.1: Discriminator Architecture: Where n is the number of feature maps, k is the kernel size, and s is stride.

3.7.1 Patch Creation

The Discriminator receives high-resolution VIIRS images as input, with the shape defined as (Channels, Height, Width):

$$\text{Input Shape} = (3, 162, 210) \quad (3.21)$$

As the image passes through the network, it undergoes a series of downsampling operations via convolutional layers with stride 2. Each downsampling step reduces both the height and width by half. After four such layers, the output shape is:

$$\text{Output Shape} = \left(1, \frac{162}{2^4}, \frac{210}{2^4}\right) = (1, 10, 13) \quad (3.22)$$

This progressive downsampling again closely resembles the generators progressive upsampling but the discriminator performs the reverse. The purpose of this aggressive reduction in spatial dimensions is to enable the discriminator to capture increasingly abstract, high-level features such as textures, edges, and object structures by expanding its receptive field at each layer.

The final output, a feature map of shape $(1, 10, 13)$, represents a grid of patch-level predictions. Instead of classifying the entire image as real or fake using a single scalar output, the discriminator evaluates overlapping local regions (or patches) within the image. This approach, often referred to as a PatchGAN [40], has been shown to be effective in encouraging the Generator to produce locally realistic details.

The use of small patches as output is intentional as it allows the discriminator to apply pressure at a more granular level, penalizing local inconsistencies rather than global mismatches. Each value in the output map can be interpreted as a probability of "realness" for a specific region of the input image. This level of supervision is particularly beneficial in high-resolution image tasks, where pixel-level realism is important.

By the end of the network, this compression achieved through n downsampling layers which results in an output that is both compact and information-dense, allowing the discriminator to operate at its fullest capacity. The small output patches serve as rich, localized representations that retain enough semantic detail for the Discriminator to reliably evaluate the authenticity of structures within each region of the image. This patch-based approach allows for a more focused and effective assessment of realism across the entire input.

3.7.2 Discriminator Block

The output consists of a feature map rather than a single scalar value, following the patch-based approach, which classifies gradually smaller regions (patches) of the image as real or fake instead of making a single decision for the entire image [41]. By gradually reducing the VIIRS sensor image resolution, the discriminator can test low to high level features to further challenge the generator for accuracy.

The Discriminator blocks consist of the following steps:

1. First Convolutional Layer:

$$X_1 = \text{Conv2d}(X_0, W_1, b_1, \text{stride} = 1, \text{padding} = 1) \quad (3.23)$$

- X_0 is the input tensor with shape (C, H, W) , where C is the number of channels, H is the height, and W is the width.
- W_1 is the convolutional kernel of shape (C_1, C_0, k_h, k_w) , where C_1 is the number of output channels, C_0 is the number of input channels, and $k_h = k_w = 3$ represent the kernel height and width.
- b_1 is the bias term, which is learned during training.

This initial convolutional layer generates a feature map that preserves the spatial dimensions of the input and highlights local patterns in the image, such as edges, textures, and small structures. These features serve as the foundation for deeper layers to extract increasingly abstract and complex representations.

2. Batch Normalization (skipped for first block):

$$X_2 = \frac{X_1 - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta \quad (3.24)$$

Where: μ and σ^2 are the mean and variance of the feature map X_1 across the mini-batch. ϵ is a small automatically generated constant to avoid division by zero. γ is the scale parameter, and β is the shift parameter used for normalization, this is a non-traditional alternative method of biasing [42]. When starting γ is initialized at 1, while β is initialized at 0. However, through the learning process they are automatically learns optimal values for these variables via backpropagation, essentially they are trainable parameters. First, Batch Normalization computes the mean and variance for each

feature map across the mini-batch. Then, it normalizes the activations by subtracting the mean and dividing by the standard deviation. Finally, it scales and shifts the normalized activations using gamma and beta, allowing the network to adjust the normalized activations for better learning.

3. LeakyReLU Activation: After the batch normalization (if not the first convolutional layer), a LeakyReLU activation function is added.

$$X_3 = \text{LeakyReLU}(X_2) = \begin{cases} X_2, & \text{if } X_2 > 0 \\ \alpha X_2, & \text{otherwise} \end{cases} \quad (3.25)$$

where $\alpha = 0.2$ is the negative slope coefficient. This gives a slope of 0.2 for negative inputs, in contrast to the automatic setting of $\alpha = 0.1$ [43].

This stronger "leak" is beneficial for improved feature training as it amplifies the gradient flow even through negative activations. By allowing a stronger gradient in the negative domain, this setting helps mitigate issues like slow learning and partial inactivity, which can occur with the smaller slope.

4. Downsampling Convolutional Layer:

$$X_4 = \text{Conv2d}(X_3, W_2, b_2, \text{stride} = 2, \text{padding} = 1) \quad (3.26)$$

This second convolution in the block reduces the spatial dimensions by a factor of 2 while further transforming feature representations. It uses the same kernel size 3×3 , but with a stride of 2, enabling downsampling. This operation is repeated across four blocks, cumulatively resulting in a downscaling of $\frac{H}{2^4} \times \frac{W}{2^4}$, allowing the discriminator to progressively analyze the image from fine to coarse resolutions.

5. Batch Normalization and Activation (Post-downsampling): The output X_4 is again normalized and passed through another LeakyReLU activation:

$$X_5 = \text{LeakyReLU}(\text{BatchNorm}(X_4)) \quad (3.27)$$

This further stabilizes training and introduces non-linearity after downsampling.

6. Final Convolutional Layer (Patch Output): At the end of all convolutional blocks, a final convolutional layer with a single output channel is applied:

$$X_{\text{patch}} = \text{Conv2d}(X_5, W_3, b_3, \text{stride} = 1, \text{padding} = 1) \quad (3.28)$$

This generates a patch-based output rather than a single scalar value, following the PatchGAN architecture [41]. Each value in this output map represents the discriminator's assessment of a local image patch being real or fake.

Finally, each block halves the spatial dimensions while doubling the number of channels, following this pattern:

Table 3.7.1: Discriminator Architecture Details

Block	Input Channels	Output Channels	Spatial Dimensions
1	3	64	$(\frac{H}{2^0}, \frac{W}{2^0})$
2	64	128	$(\frac{H}{2^1}, \frac{W}{2^1})$
3	128	256	$(\frac{H}{2^2}, \frac{W}{2^2})$
4	256	512	$(\frac{H}{2^3}, \frac{W}{2^3})$
5	512	1	$(\frac{H}{2^4}, \frac{W}{2^4})$

Where the 512 to 1 transition doesn't reduce the spatial resolution to 1; it only reduces the depth (number of channels) to 1. The spatial resolution (height and width) of the final output corresponds to smaller image patches that the PatchGAN architecture classifies individually. This returns to the output dimensions specified in 3.22. This pattern effectively reduces the image resolution while progressively learning higher-level features from the input image, ultimately leading to a feature

map with a reduced spatial size but much more depth (features). The final output is used by the discriminator to classify the image as real or fake.

Summary of Discriminator In summary, each input image is passed sequentially through the discriminator block layers. This means that the image undergoes the series of operations being convolutions, batch normalizations, and activations as previously described. As the image progresses through the network, it learns low to high level features in a cascading method from the inputted VIIRS sensor image, allowing the discriminator to more effectively distinguish between real and generated images. The Discriminator's goal is to challenge the Generator by learning to classify real images as real and fake images as fake.

The final convolution reduces the feature map to a single-channel output, and the Discriminator's forward function processes an input image through the entire model sequentially:

$$y = \text{Discriminator}(x) \quad (3.29)$$

Here, x represents the input image, and y is the final output, where each pixel in y represents the Discriminator's decision about the corresponding patch in the input image. This output serves as a patch-based prediction of whether each image patch is real or fake. In summary, the key features of this Discriminator include its patch-style classification of image patches, four downsampling blocks that progressively reduce spatial dimensions while increasing depth, LeakyReLU activations to ensure even negative gradients are alive in the network with a steeper slope, and a final single-channel output that produces a feature map of real/fake scores. The discriminator, through its structured downsampling and patch-based output, plays an important role in training the generator by providing feedback on the realism of generated synthetic images.

3.8 Feature Extraction

3.8.1 Feature Extraction

Feature extraction in this model is an important step that involves transforming the input image into a set of features that capture patterns, textures, and structures. In the context of wildfire detection, this becomes particularly important because the high brightness temperature values associated with wildfires are essential features that need to be preserved in the generated images. These features, such as sharp temperature gradients and specific thermal patterns, are necessary for accurately detecting and monitoring wildfire events. To extract these relevant features, the model uses a pre-trained VGG-19 network, which has been trained on large, diverse datasets like ImageNet [44]. VGG-19 is well known for its ability to capture hierarchical features across many levels of complexity due to its deep architecture of convolutional layers. The pretrained VGG-19 model was imported through torchvision to be used. As the data progresses through the layers, the network detects increasingly complex patterns, from simple edges to intricate shapes and textures.

In this architecture, the first 35 layers of VGG-19 are used for feature extraction. The first layers capture low-level to high-level features, which help in differentiating between areas affected by wildfires and those that are not [45]. This feature extraction approach enables the model to produce more realistic and visually pleasing outputs by aligning the generated image with the high-level features of the target image.

$$I \in \mathbb{R}^{H \times W \times C} \quad (3.30)$$

The input image I has dimensions of height H , width W , and C channels. The image is represented as a tensor with shape $\mathbb{R}^{H \times W \times C}$, where H , W , and C denote the height, width, and the number of channels of the image.

$$F = \mathcal{F}_{35}(I) \quad (3.31)$$

The function $\mathcal{F}_{35}$ represents the application of the first 35 layers of the VGG-19 model to the input image I . This process extracts feature maps from the image, resulting in a feature tensor F that contains multi-dimensional representations of the image's features. The output F captures low-level to high-level features depending on which layer of the model it corresponds to (higher being more complex).

$$F = \mathcal{F}_{35}(I) = [f_1(I), f_2(I), \dots, f_n(I)] \quad (3.32)$$

F represents the final feature map output after passing the image through the first 35 layers of VGG-19. This output consists of individual feature maps $f_1, f_2, \dots, f_n$, each corresponding to the features extracted at different layers. $f_1(I)$ represents low-level features which were extracted from the earlier layers of the network. $f_2(I), \dots, f_n(I)$ represent progressively more complex features as the image passes through deeper layers.

To summarize, feature extraction plays an important role in the FireGAN by allowing the model to focus on the most relevant and informative aspects of the input images, which is especially important in the context of wildfire detection. In this case, feature extraction helps the model to identify specific patterns related to wildfires, such as rises in brightness temperatures in the MWIR bands and rises in AFI values.

With VGG-19, which is known for its ability to extract features at various levels of abstraction, the FireGAN can capture both low-level features (such as edges, textures, and color contrasts) and high-level features (such as geographical structures, objects, and fire-related thermal patterns). When an input image passes through the first 35 layers of VGG-19, it generates a feature map that contains a mix of these various features, with each successive layer learning more complex representations of the image.

These extracted features are then incorporated into the loss function during the training process. By using these features, the model can better align the generated images with the target images, effectively minimizing the difference between them. This strategy helps the model to focus on the most important characteristics of the images—such as the presence of fire, smoke, or specific

environmental changes associated with wildfires—thereby reducing the overall loss and improving the accuracy of the FireGAN’s predictions.

In essence, feature extraction serves to simplify complex input data by transforming it into a more informative and structured representation. This not only enhances the model’s predictive capabilities but also improves its ability to make more accurate, realistic predictions with lower error rates, ultimately leading to more effective wildfire detection.

3.9 Loss Calculations

3.9.1 GeneratorRRDB Loss Calculations

In the FireGAN model, three main loss functions: The GAN loss (based on binary cross-entropy), the content loss (based on mean squared error), and the pixel loss (also based on mean squared error). Below are the implementations of these loss functions.

Starting with Binary Cross Entropy with Logits Loss function used to calculate the GANs Adversarial Loss:

$$\text{loss}_{\text{GAN}} = - [\text{valid} \cdot \log (\sigma (\text{pred}_{\text{fake}} - E[\text{pred}_{\text{real}}])) + (1 - \text{valid}) \cdot \log (1 - \sigma (\text{pred}_{\text{fake}} - E[\text{pred}_{\text{real}}]))] \quad (3.33)$$

- loss_{GAN} : The adversarial loss for the generator, representing how well the generated image matches the real ground truth image.
- valid : The ground truth image (VIIRS sensor) used as the target for the binary cross-entropy (BCE) loss set to 1.
- $\text{pred}_{\text{fake}}$: The raw output (logits) from the discriminator for the generated (fake) image.
- $E[\text{pred}_{\text{real}}]$: The mean of the discriminator’s outputs for the real images in the batch, used to normalize the discriminator’s prediction for the real images.

- $\sigma(x)$: The sigmoid function, defined as $\sigma(x) = \frac{1}{1+e^{-x}}$, which maps logits to probabilities in the range $[0, 1]$.

The Binary Cross-Entropy with Logits loss function in the GAN setup is designed to train the generator to create realistic images that can successfully deceive the discriminator [46]. The loss function compares the discriminator's raw logits for generated (fake) images with the expected logits for real images. It uses the sigmoid function to convert these logits into probabilities. The generator aims to minimize the loss by making the discriminator classify the fake images as real. This is achieved by minimizing the binary cross-entropy loss for both real and fake images. For real images, the discriminator's mean output is used to normalize the logits, helping the generator learn how to produce realistic synthetic images. The "valid" variable is set to 1 because it represents the ground truth label for real images. By assigning a value of 1, this guides the generator to learn how to produce realistic images that can be distinguished from fake ones. The discriminator is then trained to predict a value close to 1 for real images. This means that when the discriminator processes real images, it should output values near 1, indicating the images are "real". In contrast, the generator's goal is to create fake images that the discriminator will mistakenly classify as real, producing outputs that should ideally lead to discriminator values close to 1 as well. The "valid" label helps the discriminator distinguish between real and fake images by training it to assign values near 1 for real images and near 0 for fake images. This is an example of the adversarial ground truth values which are assigned to the discriminator and the generator. Through its training the generator will improve over time to produce more convincing fake images thereby generating values close to 1. Additionally, the "valid" variable is reshaped to match the dimensions of the discriminator's output (e.g., 1, 10, 13).

Next is Mean Square Error Loss which is used to calculate loss content. More specifically, Loss Content is an MSE calculation between ground truth VIIRS sensor images along with FireGAN generated images, both having been run through the Feature Extractor which was previously described. The MSE Loss can be shown as the following:

$$\text{loss}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (\text{feature}_{\text{ground truth},i} - \text{feature}_{\text{generated},i})^2 \quad (3.34)$$

- loss_{MSE} : The Mean Squared Error loss, which measures the difference between the features of the ground truth (VIIRS) image and the generated image (FireGAN output).
- N : The total number of features used in the comparison.
- $\text{feature}_{\text{ground truth},i}$: The i -th feature value extracted from the ground truth (VIIRS) image after passing through the Feature Extractor.
- $\text{feature}_{\text{generated},i}$: The i -th feature value extracted from the generated (FireGAN) image after passing through the Feature Extractor.

With MSE being calculated as the loss content, the objective is for the generator in FireGAN to produce images that are as close as possible to the ground truth (VIIRS) images, but in terms of extracted features. By minimizing the MSE loss, the generator improves its ability to generate images that match the real images more accurately, feature representations that were shown from the Feature Extractor. The lower the MSE loss, the closer the generated image is to the real image, reflecting improved performance of the generator in producing realistic results. This helps the model capture the fine details of the ground truth images, which is important for tasks like wildfire detection that are generating high-resolution wildfire pixels.

The last loss calculation that contributes to the total generator loss is the application of MSE, as previously described, but for the pixel loss as shown:

$$\text{loss}_{\text{pixel}} = \frac{1}{N} \sum_{i=1}^N (\text{pixel}_{\text{ground truth},i} - \text{pixel}_{\text{generated},i})^2 \quad (3.35)$$

- $\text{loss}_{\text{pixel}}$: The Mean Squared Error (MSE) pixel loss, measuring the difference between the pixel values of the ground truth and generated images.
- N : The total number of pixels in the image.
- $\text{pixel}_{\text{ground truth},i}$: The pixel value at the i -th location in the ground truth (VIIRS) image.

- $\text{pixel}_{\text{generated},i}$: The pixel value at the i -th location in the generated (FireGAN) image.

Like the previous MSE application, by minimizing the pixel loss, the generator is encouraged to match the actual pixel values of the ground truth image, ensuring that the generated image is not only close in terms of features but also visually similar at the pixel level. This combined loss function (feature loss and pixel loss) helps the FireGAN to produce high-quality images that are perceptually accurate, which is essential for tasks like wildfire detection and high-resolution image generation.

The overall generator Loss function can be shown as the following:

$$\text{loss_G} = \text{loss_content} + 0.3 \times \text{loss_GAN} + 150 \times \text{loss_pixel} \quad (3.36)$$

- loss_G : The final generator loss, which is a weighted sum of the content loss, adversarial loss, and pixel loss.
- loss_content : The content loss, typically calculated using Mean Squared Error (MSE) between the features of the ground truth (VIIRS) and generated images, based on the Feature Extractor.
- loss_GAN : The adversarial loss, calculated using Binary Cross Entropy with logits, guiding the generator to produce images that can fool the discriminator.
- loss_pixel : The pixel loss, calculated using MSE directly on the pixel values of the generated and ground truth images, ensuring pixel-level accuracy.

This function shows the total loss function which is represented for backpropagation. It is important to note that the Pixel Loss plays a significant role in ensuring that the generator not only produces realistic features but also creates visually accurate images at the pixel level. The high weight ($150 \times \text{loss_pixel}$) applied to the pixel loss emphasizes its importance, ensuring that pixel-wise accuracy is prioritized during training. The inclusion of the adversarial loss ($0.3 \times \text{loss_GAN}$) encourages the generator to produce images that can deceive the discriminator, while the content

loss focuses on preserving the features of the ground truth in the generated images. This combination of losses ensures that the generator learns to produce both feature-accurate and pixel-accurate images, which is important for generating realistic outputs in tasks like wildfire detection or image super-resolution.

After the full loss calculation, backpropagation is applied to minimize the loss function by updating the model's weights. During backpropagation, the gradients of the loss function $loss_G$ with respect to the generator's parameters are computed. This is done by applying the chain rule, which propagates the error backward through the network and calculates the contribution of each parameter to the overall loss [47]. Once the gradients are computed, the Adam optimizer adjusts the weights based on these gradients, helping the generator improve its output. This process is repeated in each training iteration, progressively reducing the loss and allowing the model to generate more accurate results over time [48].

3.9.2 Discriminator Loss Calculations

To train the Discriminator, loss functions are also important so the model can differentiate between real and fake images. The discriminator receives two types of input: real images from the dataset and fake images generated by the generator. The goal is to train the discriminator to classify real images as real and fake images as fake, using Binary Cross-Entropy (BCE) loss. The BCE loss measures the difference between the discriminator's predicted probability and the true label (real or fake) for each image.

Starting with the real image loss:

$$loss_{real} = - [\text{valid} \cdot \log(\sigma(\text{pred}_{real} - E[\text{pred}_{fake}])) + (1 - \text{valid}) \cdot \log(1 - \sigma(\text{pred}_{real} - E[\text{pred}_{fake}]))] \quad (3.37)$$

- $loss_{real}$: The Binary Cross-Entropy loss for the real images.
- valid : The ground truth images valid set to equal 1.
- pred_{real} : The raw discriminator output logits for the real images.

- $E[\text{pred}_{\text{fake}}]$: The mean of the discriminator's predictions for the fake images across the batch.
- $\sigma(x)$: The sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$, which maps the logits to probabilities in the range $[0, 1]$.

The discriminator is penalized when it incorrectly classifies real images as fake. This loss is calculated by comparing the discriminator's output for real images against the true label (real). This function is complimented by the fake image loss calculation shown as:

$$\text{loss}_{\text{fake}} = - [(1 - \text{fake}) \cdot \log (1 - \sigma (\text{pred}_{\text{fake}} - E[\text{pred}_{\text{real}}]))] \quad (3.38)$$

- $\text{loss}_{\text{fake}}$: The Binary Cross-Entropy loss for the fake images.
- fake : The ground truth label for fake images set to 0.
- $\text{pred}_{\text{fake}}$: The raw discriminator output logits for the fake images.
- $E[\text{pred}_{\text{real}}]$: The mean of the discriminator's predictions for the real images across the batch.
- $\sigma(x)$: The sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$, which maps the logits to probabilities in the range $[0, 1]$.

Similarly, the discriminator is penalized when it classifies fake images as real. This loss is calculated by comparing the discriminator's output for fake images against the true label (fake). However, there is an important distinction in this calculation because the "fake" variable is set to zero. This is because the discriminator is trained to predict a value close to 0 when differentiating fake images from real ones. Just like the generator, the fake generated image is compared to the corresponding low-resolution GOES ABI image, which is reshaped to match the dimensions expected by the discriminator (e.g., 1, 10, 13).

Both of these loss calculations are added to the following total loss function:

$$\text{loss_D} = \text{loss}_{\text{real}} + \text{loss}_{\text{fake}} \quad (3.39)$$

- loss_D : The total discriminator loss, which is the sum of the loss for real and fake images.
- $\text{loss}_{\text{real}}$: The loss for the real images, as calculated in the first equation.
- $\text{loss}_{\text{fake}}$: The loss for the fake images, as calculated in the second equation.

As the sum of the losses is added it will show the discriminators ability to discriminate against both real and generated images, with the ultimate goal of minimizing the total value, indicating that the discriminator is well-trained.

This process follows a similar optimization and backpropagation strategy that is used in the generator's training. Once the total loss for the discriminator is calculated by adding up its individual loss components, backpropagation is used to propagate the error backward through the network. This involves calculating how much each parameter (or weight) of the discriminator contributed to the error. After the gradients for the discriminator's loss are computed, the Adam optimizer adjusts the model's weights based on these gradients. This iterative process helps the discriminator progressively improve its performance by reducing the loss, making it more accurate in distinguishing between real and generated images. Over time, this combination between the generator and the discriminator drives continuous refinement in both models, resulting in higher-quality image generation. Each training cycle enables the discriminator to become better at its task, ultimately leading to more realistic results from the generator.

3.10 Normalization

This research uses Max-Min Normalization and Denormalization with the FireGAN model to ensure the training and testing images are consistent. Normalization is applied after the image processing stage, so all of the inputs are scaled before being fed into the model. Likewise, the generated outputs are immediately denormalized to restore them to their original physical units for evaluation. Since this research uses three separate channels (MWIR, LWIR, and AFI), there are three separate datasets that need to be calculated. This issue is solved when normalization scales each input channel to the range $[0, 1]$, helping the model to converge more efficiently and pre-

vents bias due to different value ranges across channels. This issue is highlighted when comparing brightness temperatures to the AFI calculation, as they have significantly different values.

The Max-Min Normalization formula used is:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.40)$$

The minimum and maximum values for each variable are:

- C07: $x_{\min} = 220^{\circ}\text{K}$, $x_{\max} = 350^{\circ}\text{K}$
- C14: $x_{\min} = 220^{\circ}\text{K}$, $x_{\max} = 320^{\circ}\text{K}$
- AFI: $x_{\min} = 0$, $x_{\max} = 8$

A wide range for brightness temperature is selected to capture the majority of pixel values across all images. The ranges are also different between MWIR and LWIR because, generally the LWIR results will lower even when there is an active fire. However, since MWIR is far more sensitive to wildfire hotspots, it needs to have a larger max value to accomodate for that. The Min-Max values are fixed for the entire dataset to ensure consistency.

This ensures consistency in the normalized data fed into FireGAN. With this consistency, the model can learn more stable and generalizable patterns across varying input conditions, such as changes in altitudes, time of day, or atmospheric phenomena. Next, the normalized values are converted back to their original scale when Max-Min Denormalization is applied:

$$x_{\text{denorm}} = x_{\text{norm}} \cdot (x_{\max} - x_{\min}) + x_{\min} \quad (3.41)$$

In summary, normalization is a necessary step in this application due to the range of satellite imagery values and the added complexity introduced by calculations such as AFI. It ensures that the model focuses on learning meaningful patterns rather than being influenced by arbitrary differences in Brightness Temperature value scales.

3.11 Model Training

This section provides an overview of the key hyperparameters, training strategies, and optimization techniques used to train the model. Including choices made regarding the learning rate, the number of epochs, batch sizes, and the specific optimizer used to update the model's parameters. Along with the rationale behind the selection of various hyperparameters, including their impact on model performance and convergence. This section also covers the optimization strategy, with a focus on how the Adam optimizer, with its adaptive learning capabilities, was used to stabilize training, particularly in the adversarial setting of the GAN.

In total, the FireGAN model was trained over the course of 24,000 epochs. This large number of training iterations was essential due to the complexity of the model architecture as it trains a generator and discriminator which both need to be accurate to fully reap the benefits of the adversarial framework. Due of this complexity, the model requires a large number of passes through the dataset to effectively learn the mapping from low-resolution to high-resolution satellite imagery. A prolonged training duration allows the generator to iteratively refine its outputs, while simultaneously allowing the discriminator to become increasingly effective at distinguishing between real and generated images. The extended training ensures stability in the adversarial process and allows the model to converge towards generating visually realistic and pixel-accurate high-resolution images.

During training, the learning rate was set to 0.0001. The learning rate is an important hyperparameter that determines the size of the steps the optimizer takes when updating model weights in response to the calculated gradients. These steps are essential for converging towards the minimum of the loss function. If the learning rate is set too high, the optimizer may overshoot optimal values, leading to unstable training and significant fluctuations in the loss, ultimately resulting in poor image generation. On the other hand, if the learning rate is too low, the model may converge very slowly, consuming excessive computational resources and time. In this research, a learning rate of 0.0001 was selected as it provided a stable training process while producing accurate and visually realistic high-resolution images.

The optimization strategy used in this training process is based on the "Adam Optimizer", combined with the previously stated learning rate (α) of 0.0001, along with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The Adam optimizer is used for updating the weights of both the generator and the discriminator. Adam combines the advantages of two other extensions of stochastic gradient descent: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp) [48].

For each parameter θ , the Adam optimizer maintains two moving averages:

- The first moment estimate (mean of gradients): m_t
- The second moment estimate (uncentered variance of gradients): v_t

The update rules for Adam are as follows [48]:

$$g_t = \nabla_{\theta} \mathcal{L}_t(\theta_{t-1}) \quad (3.42)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (3.43)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3.44)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3.45)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (3.46)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (3.47)$$

- θ_t : model parameters at iteration t
- α : learning rate ($\alpha = 0.0001$)
- β_1 : exponential decay rate for the first moment estimate ($\beta_1 = 0.9$)
- β_2 : exponential decay rate for the second moment estimate ($\beta_2 = 0.999$)
- ϵ : small constant for numerical stability (typically $\epsilon = 10^{-8}$)
- $\mathcal{L}_t$: the loss function at step t

For each parameter θ , Adam maintains two moving averages, being the first moment estimate m_t (the mean of gradients) and the second moment estimate v_t (the uncentered variance of gradients). The parameters are updated according to θ_t where, θ_t is the updated parameter at iteration t , α is the learning rate (set to 0.0001), $\hat{m}_t$ is the bias-corrected first moment estimate, $\hat{v}_t$ is bias-corrected second moment estimate, and ϵ is small constant automatically added for numerical stability.

Adam is employed for both the generator and the discriminator networks. During training, the loss is first backpropagated to determine how the model is performing, and to compute the gradients of the loss with respect to the model parameters. Following this, the optimizer function is called, which updates the parameters using the Adam optimization based on θ_t .

This optimization strategy is effective for training GANs as it adaptively scales the parameters based on the estimates from the first and second moments, helping to stabilize the adversarial training process.

3.12 Evaluation Metrics

This study uses a total of four performance metrics to evaluate the models' accuracy and ability to successfully generate realistic synthetic data based from GOES ABI images. These metrics include, Root Mean Square Error (RMSE), Coefficient of Determination (R^2), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM).

Starting with RMSE calculation, where the average distance between the models prediction in comparison to the ground truth (VIIRS) values are determined.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.48)$$

- n : The total number of data points in the dataset.
- y_i : Ground truth (VIIRS data) for the i -th data point.
- $\hat{y}_i$: FireGAN predicted value for the i -th data point.
- $(y_i - \hat{y}_i)^2$: The squared error for the i -th data point.

- $\sum_{i=1}^n$: The summation over all data points, from $i = 1$ to n .

From the RMSE calculation, a value closer to 0 is ideal and indicates the model is producing highly accurate results when compared to the ground truth image [49]. In this case a low RMSE would indicate high accuracy in pixel value predictions from the model compared to its corresponding VIIRS image.

Next, the Coefficient of Determination (R^2) function

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.49)$$

- n : The total number of data points in the dataset.
- y_i : The ground truth (VIIRS) value for the i -th data point.
- $\hat{y}_i$: FireGAN predicted value for the i -th data point.
- $\bar{y}$: The *mean* of the actual ground truth values, $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$, which is the average of all y_i values.

The value of R^2 measures the proportion of the variance in the ground truth data y that is predictable from the model's predictions $\hat{y}$ by comparing two primary terms. One being Residual sum of squares (RSS), where $\sum_{i=1}^n (y_i - \hat{y}_i)^2$, that represents the squared differences between the ground truth and predicted values. Next, Total sum of squares (TSS), where $\sum_{i=1}^n (y_i - \bar{y})^2$, that represents the squared differences between the actual values and the mean of the actual values. Importantly, this indicates the total variance of the data [50].

An ideal result would yield a result around $R^2 = 1$. This would indicate that the model perfectly fits the data, the predictions match the ground truth, and all variance in the actual data is explained by the model.

If R^2 equals 0 or a nearby decimal value, it would indicate that the model does no better than just predicting the mean of the actual values, so the model fails to explain any of the variance in the data.

Next being the Peak Signal-to-Noise Ratio (PSNR), is a metric used to measure the quality of an image, comparing the difference between the ground truth image and the predicted image. PSNR is often used to evaluate image reconstruction and compression quality.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{\text{MSE}} \right) \quad (3.50)$$

Where:

$$\text{MSE} = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N (I(i, j) - K(i, j))^2 \quad (3.51)$$

- MAX : Is 1 in this application as the image is normalized.
- M and N are the dimensions of the image (height and width).
- $I(i, j)$: The pixel value of the ground truth image at position (i, j) .
- $K(i, j)$: The pixel value of the predicted image at position (i, j) .
- $(I(i, j) - K(i, j))^2$: The squared error for each pixel.
- $\sum_{i=1}^M \sum_{j=1}^N$: The summation over all pixels in the image.
- $M \cdot N$: The total number of pixels in the image, used to calculate the average squared error.

For the results, a higher PSNR value, usually greater than 30dB, indicates good image quality with less noise or distortion. Moderate PSNR of 20-30dB will show some distortion, but is still acceptable. While a lower PSNR, such as < 20dB indicates significant distortion or error between the predicted and ground truth images [51].

Last, the Structural Similarity Index Measure (SSIM) is a perceptual metric for evaluating image quality. SSIM is designed to model the human visual system's perception by capturing structural information within images. It measures the similarity between two images by comparing their luminance, contrast, and structural components [52].

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.52)$$

- μ_x and μ_y : The mean pixel intensity values of the ground truth image x and the predicted image y , respectively.
- σ_x^2 and σ_y^2 : The variance values of the ground truth image x and the predicted image y , respectively.
- σ_{xy} : The covariance between the ground truth image x and the predicted image y .
- C_1 and C_2 : Small constants used to avoid division by zero. They are defined as:

$$C_1 = (K_1 L)^2, \quad C_2 = (K_2 L)^2$$

where K_1 and K_2 are small constants ($K_1 = 0.01$ and $K_2 = 0.03$ by default for SSIM in the Python package scikit-image), and L is the range of the image (e.g., $L = 1$ in this case).

The SSIM results will range from 1 to -1. 1 being the ideal result which indicates structural and perceptual similarity between the ground truth VIIRS image and the predicted FireGAN image. Collectively, these four metrics provide a comprehensive evaluation of the model's performance. Where RMSE and R^2 assess numerical accuracy, while PSNR and SSIM focus on perceptual image quality, helping to validate both the pixel-wise accuracy and the visual realism of the generated FireGAN images in comparison to their corresponding VIIRS images for the testing portion of this research.

Next, the fire detection performance is evaluated using several metrics including: Heidke Skill Score (HSS), Critical Success Index (CSI), Probability of Detection (POD), and False Alarm Rate (FAR). Below are the equations and explanations for each metric.

The Heidke Skill Score (HSS) compares the model's predictions to a random forecast. It is useful for evaluating binary classification systems, particularly when dealing with imbalanced datasets (such as fire vs non-fire detection). The equation for HSS is given by:

$$\text{HSS} = \frac{2 \cdot (TP \cdot TN - FP \cdot FN)}{(TP + FP)(FP + TN) + (TN + FN)(FN + TP)} \quad (3.53)$$

- TP : True Positives (fire pixels correctly identified)
- TN : True Negatives (non-fire pixels correctly identified)
- FP : False Positives (non-fire pixels incorrectly identified as fire)
- FN : False Negatives (fire pixels incorrectly identified as non-fire)

The value of HSS ranges from -1 (worse than random) to 1 (perfect prediction). A higher HSS indicates better prediction skill.

The Critical Success Index (CSI) measures the accuracy of binary classification by focusing on the proportion of correctly predicted fire pixels. The equation for CSI is:

$$CSI = \frac{TP}{TP + FP + FN} \quad (3.54)$$

- TP : True Positives
- FP : False Positives
- FN : False Negatives

The CSI value ranges from 0 to 1, where 0 indicates no correct detections and 1 indicates perfect detection performance. A higher CSI indicates better model performance in terms of fire detection.

The Probability of Detection (POD) is also known as the True Positive Rate (TPR). It measures the fraction of actual fire pixels that are correctly identified by the model. The equation for POD is:

$$POD = \frac{TP}{TP + FN} \quad (3.55)$$

- TP : True Positives
- FN : False Negatives

POD measures the model's ability to detect true fire pixels. The value ranges from 0 to 1, where 0 indicates no detection, and 1 indicates perfect detection of all fire pixels.

The False Alarm Rate (FAR) quantifies the proportion of non-fire pixels that are incorrectly classified as fire. This metric is evaluated how often the model produces false positives. The equation for FAR is:

$$\text{FAR} = \frac{FP}{FP + TN} \quad (3.56)$$

- FP : False Positives
- TN : True Negatives

A lower FAR is desirable, as it indicates fewer non-fire pixels are incorrectly identified as fire.

To summarize:

- POD and FAR are complementary. A high POD is desirable for capturing as many true fire pixels as possible, while a low FAR ensures that the model doesn't misclassify too many non-fire pixels.
- HSS and CSI are overall performance indicators that combine information from all categories (TP, TN, FP, FN). These are useful for understanding how well the model distinguishes between fire and non-fire pixels.

The metrics POD, FAR, HSS, and CSI provide complementary insights into model performance. These metrics are important for evaluating the effectiveness of the fire confidence metric in accurately detecting fires and minimizing false alarms.

3.13 Fire Confidence Calculation

To complete the outputs of the FireGAN and link them directly to wildfire detection, a Fire Confidence calculation is applied. This step provides an indicator of fire presence in the high-resolution images. Specifically, this calculation uses the downscaled GOES ABI C07, C14, and AFI images.

The Fire Confidence value is determined by integrating these data sources. By doing so, the model's output can be used to predict fire locations, improving its utility for early wildfire detection and mitigation efforts.

In practice, the Fire Confidence score ranges from 0 to 1, where higher values indicate greater confidence that a fire is present. This score helps in validating the model's predictions, ensuring that fire detection is not only accurate but also aligned with ground truth observations.

The Fire Confidence calculations are inspired from fundamentals of the GOES ABI FDC fire product [4], VIIRS fire product [5], and MODIS fire product [3]. While the previous algorithms are helpful, they do not include AFI which is highly effective in identifying wildfire hotspots. The Fire Confidence calculations are as follows:

To start with the confidence map itself, the function $S(x, a, b)$ is used to normalize the values between 0 and 1:

$$S(x, a, b) = \text{clip} \left(\frac{x - a}{b - a}, 0, 1 \right) \quad (3.57)$$

Where, x is the C07 pixel value being normalized, while a and b are the upper and lower bounds for normalization.

Next, the difference between $C07$ and $C14$, denoted as ΔT , is computed as:

$$\Delta T = C07 - C14 \quad (3.58)$$

The mean of ΔT is:

$$\Delta T_{\text{mean}} = \frac{1}{N} \sum_{i=1}^N \Delta T_i \quad (3.59)$$

Where N is the total number of pixels in the image.

The median absolute deviation (MAD) of ΔT , T_4 , and T_{11} are calculated as:

$$\text{MAD}(\Delta T) = \text{median}(|\Delta T - \Delta T_{\text{mean}}|) \quad (3.60)$$

$$\text{MAD}(C07) = \text{median}(C07 - \frac{1}{N} \sum_{i=1}^N C07) \quad (3.61)$$

Where the deviation for each pixel in $C07$ and ΔT , is calculated from each of their respective means.

The standardization of $C07$ and ΔT is computed:

$$z_4 = \frac{C07 - \frac{1}{N} \sum_{i=1}^N C07}{\text{MAD}(C07) + \epsilon} \quad (3.62)$$

$$z_{\Delta T} = \frac{\Delta T - \Delta T_{\text{mean}}}{\text{MAD}(\Delta T) + \epsilon} \quad (3.63)$$

The equations for standardizing T_4 and ΔT are designed to scale these variables based on the mean and median absolute deviation (MAD) of both $C07$ and ΔT . In the first equation, $z_4 = \frac{T_4 - \frac{1}{N} \sum_{i=1}^N C07}{\text{MAD}(C07) + \epsilon}$, the value of $C07$ is standardized by subtracting the mean of $C07$ and dividing by the MAD of $C07$, with a small constant $\epsilon = 1e^{-6}$ added to prevent division by zero. Similarly, the second equation for ΔT , $z_{\Delta T} = \frac{\Delta T - \Delta T_{\text{mean}}}{\text{MAD}(\Delta T) + \epsilon}$, standardizes the difference between $C07$ and $C14$ (denoted as ΔT) by subtracting the mean of ΔT and dividing by the MAD of ΔT , again with $\epsilon = 1e^{-6}$ for numerical stability. These standardizations help make the variables comparable and stable for further analysis, such as confidence map calculation, by removing biases and ensuring consistency across different scales.

Then, the confidence is computed based on these normalized values:

$$c_1 = S(C07, 305, 320 \text{ if daytime else } 315) \quad (3.64)$$

$$c_2 = S(z_4, 3, 6) \quad (3.65)$$

$$c_3 = S(z_{\Delta T}, 3.5, 6) \quad (3.66)$$

$$c_4 = \begin{cases} 1.0 & \text{if AFI} \geq 5 \\ \frac{\text{AFI}}{5} & \text{if AFI} < 5 \end{cases} \quad (3.67)$$

The final confidence value for each pixel is calculated as:

$$\text{confidence} = c_1 \times c_2 \times c_3 \times c_4 \quad (3.68)$$

Finally, the confidence map is clipped to ensure that values are between 0 and 1:

$$\text{Confidence Map} = \text{clip}(\text{confidence}, 0, 1) \quad (3.69)$$

The result of this will show a Fire Confidence calculation for every pixel on any given super-resolved image to determine if or how many fire pixels are present by ranking them on a scale of 0 (no fire pixel) to 1 (high confidence fire pixel). By adding this calculation, fire pixels that have values > 0.75 are easily identified along with their bordering moderate confidence values of ≥ 0.25 to ≤ 0.75 which may indicate a fire boundary which surrounds a high confidence fire area. Then last being < 0.25 which indicates little to no fire confidence for those pixels.

3.14 Conclusion

In summary, this chapter outlined the methodology used to prepare and process satellite data for training and evaluating the FireGAN model. By selecting Maui, Hawaii as the study area, VIIRS sensor and GOES ABI satellite imagery was processed and used for the model.

The preprocessing included spatial and temporal alignment, brightness temperature conversion, and AFI computation, with a correlation coefficient calculation to ensure consistency, reduced noise, and enhanced the quality of the inputs. These steps formed the foundation for a strong model performance and high quality super-resolution outputs.

This also provided a comprehensive overview of the FireGANs architecture, detailing its major components and the methodology used throughout the model development. This included an ex-

planation of the generator and discriminator networks, the use of feature extraction, and the various loss functions used to guide training namely content loss, adversarial loss, and pixel loss.

The chapter also outlined the normalization process applied to the satellite data using fixed Min-Max normalization values, ensuring consistency across the dataset. This was followed by a detailed explanation of the model training procedure, including key hyperparameters such as learning rate, batch size, number of epochs, and the use of the Adam optimizer to stabilize the adversarial training process.

Additionally, evaluation metrics were described to test the model's performance in both visual and quantitative terms. Last, the architecture of the fire confidence metric was introduced as a means to assess the likelihood of fire presence at the pixel level from the generated outputs.

Together, these components demonstrate how FireGAN bridges the gap between low-resolution 2250 m GOES ABI inputs and 375 m high-resolution, fire-relevant outputs, contributing to improved wildfire detection and monitoring from satellite-based observations.

The next chapter continues the information presented above, including fire confidence results and accuracy, FireGAN testing dataset accuracy, and further elaboration on August 9th, 2023 in Maui, HW.

Chapter 4

Results Evaluation and Discussion

4.1 Introduction

Based on the methodologies outlined in the previous chapter, the results demonstrate FireGAN’s ability to successfully enhance the spatial resolution of GOES ABI satellite imagery from 2250 m to match the 375 m resolution of VIIRS imager. This effectively eliminates the spatial limitations inherent to GOES ABI imagery while preserving its high 10-minute temporal resolution. In addition to resolution enhancement, this chapter presents the fire confidence maps calculated from the model outputs, which provide insight into the likelihood of fire presence at the pixel level.

This chapter evaluates and discusses the outcomes of the FireGAN model using both a designated testing dataset and a focused case study of the Maui wildfires that occurred on August 9th, 2023. The evaluation covers island-wide visualizations, detailed zoom-ins on the Lahaina region, and imagery from central Maui to assess the model’s ability to accurately reconstruct fine spatial features and detect wildfire activity.

The results are analyzed both qualitatively and quantitatively, illustrating how well the model performs in generating high-resolution imagery and identifying active fire regions. This chapter aims to highlight the strengths of the FireGAN framework such as its resolution enhancement and fire detection capabilities, while also identifying areas for future improvement.

4.2 Evaluation of Fire Detection and FireGAN

4.2.1 Fire Detection Results

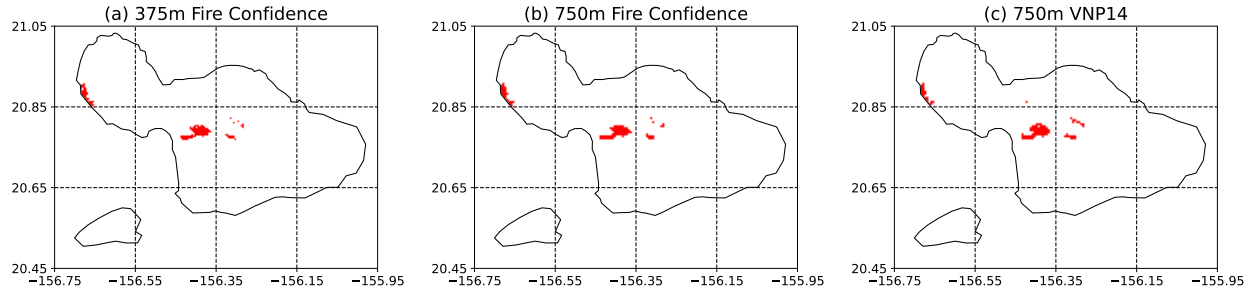


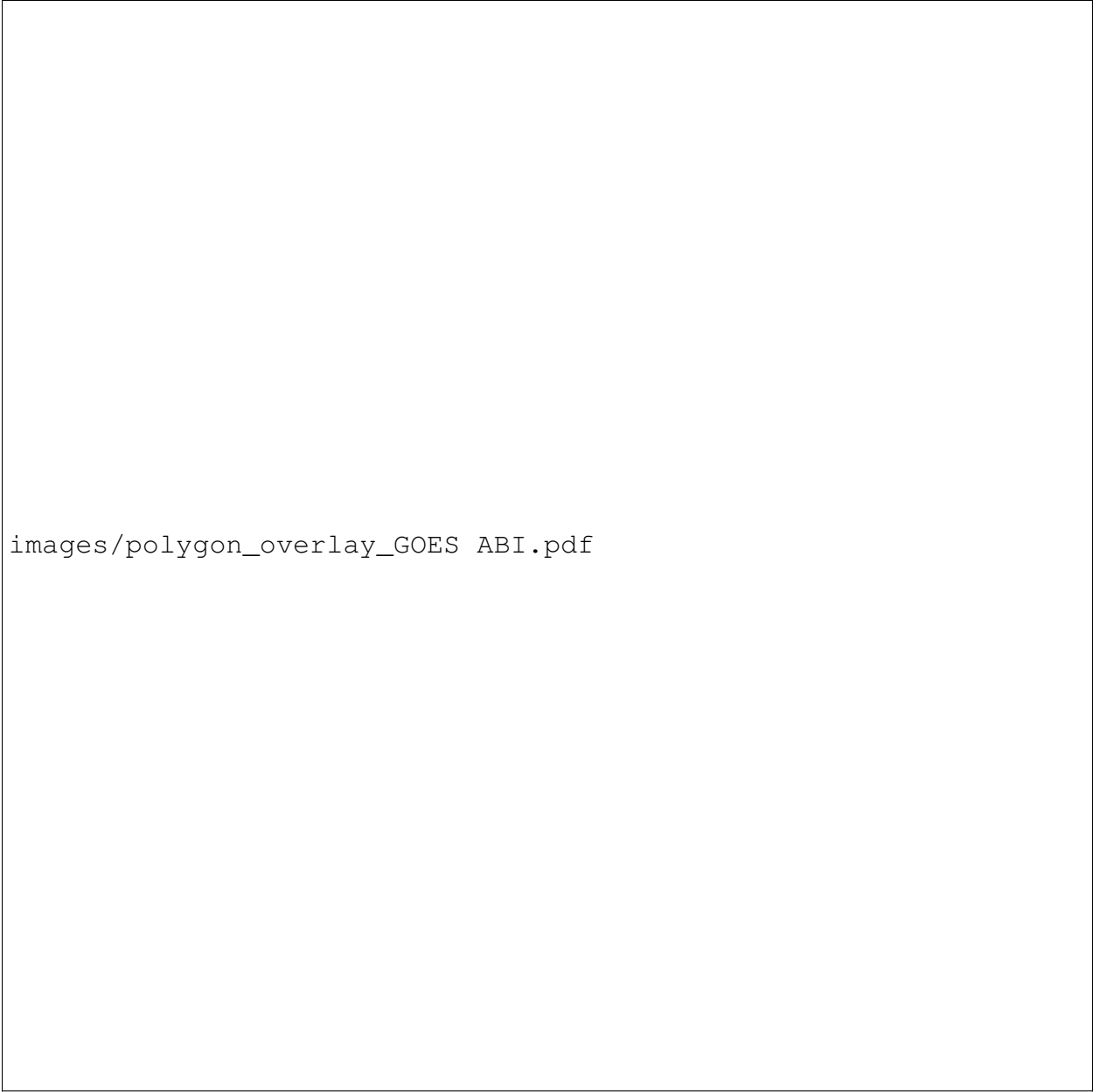
Figure 4.2.1: From August 9th, 2023 at 12:12 (UTC): (a) Fire Confidence from downscaled results, (b) Resampled Fire Confidence to meet 750 m resolution, (c) VIIRS Fire Product (750 m resolution)

It is important to note these fire pixels are ≥ 1 fire confidence which indicates that there is some form of fire, even if it is a growing fire boundary. Figure 4.2.1 shows the fire confidence findings at their original 375 m resolution, but also a reduced resolution version to match the existing VIIRS Fire Product with a resolution of 750 m. From those images being compared to each other, they achieved the following results:

Table 4.2.1: Fire Product Compared to VNP14 Results

Metric	Result Value
True Positives	92
True Negatives	19073
False Positives	16
False Negatives	19
HSS	0.839
CSI	0.724
POD	0.828
FAR	0.148

From Table 4.2.1, the FireGAN model correctly identified 92 fire pixels (True Positives), demonstrating strong alignment with the VIIRS Fire Product. Additionally, 19,073 pixels were accurately classified as non-fire (True Negatives), while only 16 pixels were mistakenly identified as fire (False Positives). The algorithm did, however, miss 19 actual fire pixels (False Negatives). These metrics indicate that the model performs well in detecting wildfire locations, achieving a high Heide Skill Score (HSS) of 0.839 and a Critical Success Index (CSI) of 0.724. The Probability of Detection (POD) was 0.828, reflecting the model's effectiveness in identifying actual fires, while the False Alarm Ratio (FAR) remained low at 0.148, suggesting minimal overprediction. Collectively, these results highlight FireGAN's capability in supporting wildfire detection from enhanced satellite imagery.



images/polygon_overlay_GOES ABI.pdf

Figure 4.2.2: (a) Ground Based Wildfire Boundary, (b) Aggregate Downscaled Fire Product, (c) Aggregate GOES ABI Fire Confidence

Figure 4.2.2 illustrates aggregate results from August 8th to August 10th, 2023, a period selected due to its alignment with the peak burn time when the ground-based fire product was created. So these days were calculated in the fire confidence metric to determine the alignment between the FireGAN results and the ground based results. During this period, every GOES-18 image was

processed through the FireGAN architecture, producing high-resolution outputs. These outputs were then evaluated using the fire confidence metric, which predicted the likelihood of fire presence at the pixel level.

The results were aggregated by averaging the fire confidence scores across all time steps to generate one fire confidence map, all of which are ≥ 1 confidence. Panel (a) shows the ground-based wildfire boundary for reference, panel (b) displays the aggregated FireGAN-derived fire confidence output, and panel (c) depicts the GOES ABI images that were also run through the fire confidence map and aggregated. This comparison demonstrates how FireGAN enhances spatial detail and more closely aligns with observed fire perimeters, offering improved spatial resolution. This is especially as evident in panel (c) where GOES ABI images alone do not accurately identify wildfire locations in comparison to the FireGAN generated images which are far more spatially accurate due to its 375 m resolution.

Table 4.2.2: Fire Product Compared to Wildfire Boundary Results

Metric	Result Value
True Positives	171
True Negatives	33705
False Positives	104
False Negatives	40
HSS	0.701
CSI	0.542
POD	0.811
FAR	0.378

Table 4.2.2 presents the performance metrics of the aggregate fire detection output from August 8th to August 10th, 2023 at a 375 m resolution instead of the 750 m resolution that was calculated in Table 4.2.1. The results indicate a total of 171 true positives, which reflect the number of

correctly identified fire pixels. Next, the model accurately classified 33,642 true negatives, or non-fire pixels, demonstrating the model’s ability to distinguish between fire and non-fire areas. The 104 false positives and 40 false negatives highlight some misclassifications, but the values are relatively low, indicating good accuracy. The Heidke Skill Score (HSS) of 0.701 and the Critical Success Index (CSI) of 0.542 suggest a moderate to strong performance in terms of correctly identifying fire regions. The Probability of Detection (POD) of 0.811 shows that the model is able to detect 81.99 percent of actual fire regions, while the False Alarm Rate (FAR) of 0.378 indicates some false alarms but still within acceptable limits for this kind of application. These metrics show the FireGAN model’s effectiveness in fire detection and its potential for operational wildfire monitoring applications.

4.3 Quantitative FireGAN Testing Results

Table 4.3.1: Full Testing Dataset Metrics

Testing Metric	Average Result	1 st Standard Deviation
RMSE	0.24	± 0.163
R ²	0.999	± 0.0005
PSNR	29.251	± 9.916
SSIM	0.966	± 0.027

Table 4.3.1 summarizes the quantitative performance of the FireGAN model on the testing dataset. The low RMSE (Root Mean Square Error) indicates a small average difference between the predicted and actual pixel values, suggesting a high level of pixel-wise accuracy. The R² value, approaching 1, implies that the model explains nearly all the variance in the data, which is a strong indication of strong predictive performance. The PSNR (Peak Signal-to-Noise Ratio) value of 29.251 dB indicates that the generated high-resolution images have a good signal quality, although the relatively high standard deviation reflects variability across different test samples. Lastly, the

SSIM (Structural Similarity Index) score of 0.966 signifies a strong structural resemblance between the generated images and the ground truth, further validating the model's capability in preserving image details during upscaling.

Collectively, these metrics demonstrate that FireGAN performs well across the testing dataset.

FireGAN Testing Dataset Results

The following images are examples from the testing dataset, accompanied by performance metrics calculated for the AFI, MWIR, and LWIR bands. Both examples highlight FireGAN's ability to accurately reconstruct GOES ABI imagery along with AFI calculations to match the spatial resolution of their corresponding VIIRS imager observations.

Figure 4.3.1 shows original GOES ABI and VIIRS sensor in the first and second rows respectively, along with the super-resolved GOES ABI results produced by the model (third row). These images are from November 2, 2023 (12:10 UTC).

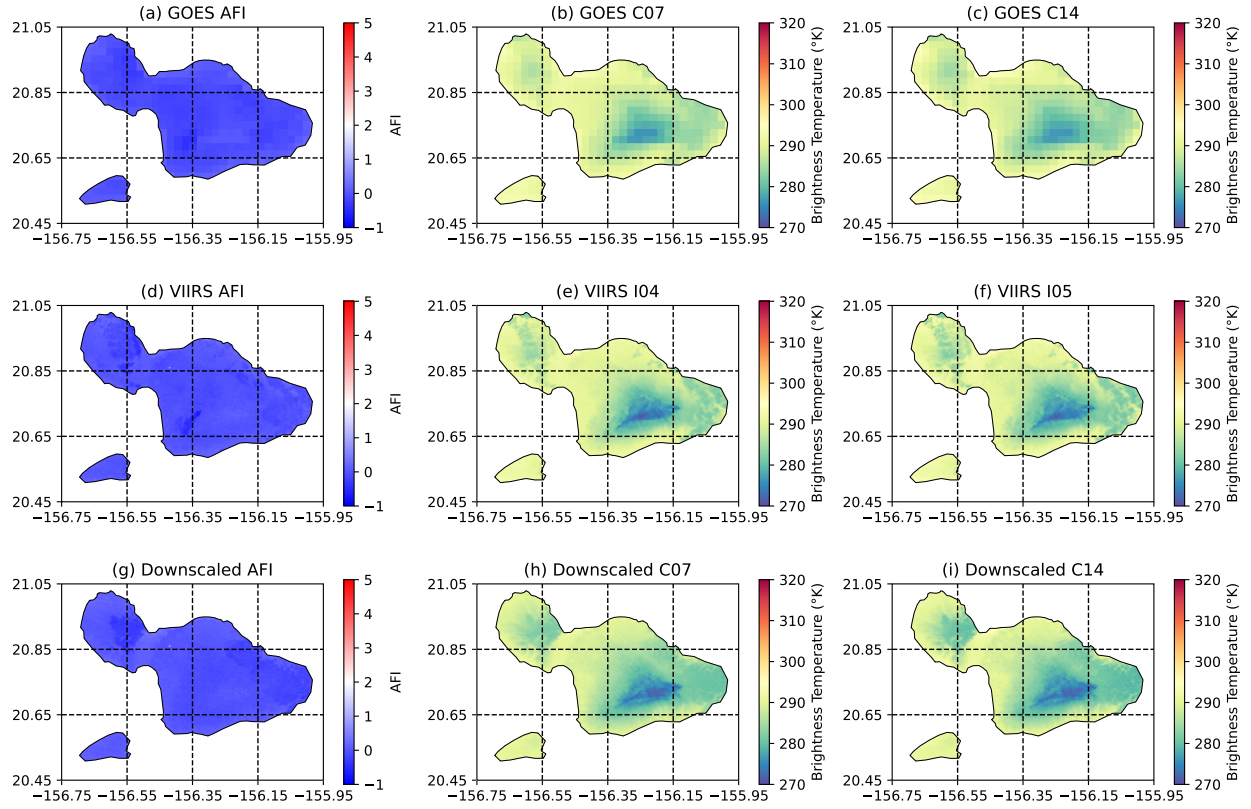


Figure 4.3.1: Testing Example From November 2, 2023 (12:10 UTC): (a) GOES ABI AFI, (b) GOES ABI C07, (c) GOES ABI C14, (d) VIIRS AFI, (e) VIIRS I04, (f) VIIRS I05, (g) Downscaled AFI, (h) Downscaled C07, (i) Downscaled C14

Table 4.3.2: November 2, 2023 (12:10 UTC) Testing Metrics

Band	RMSE	R^2	PSNR	SSIM
MWIR	0.324	0.999	36.056	0.986
LWIR	0.268	0.999	38.045	0.991
AFI	0.013	0.999	17.385	0.969

Based on the performance metrics presented in Table 4.3.3, the super-resolution model demonstrates its ability to downscale images across both spectral bands and AFI.

The MWIR band achieves strong performance with an RMSE of 0.324, a PSNR of 36.056, and an SSIM of 0.986, suggesting good noise suppression and perceptual quality. The LWIR yields the

best results, with a lower RMSE of 0.268 and a high PSNR of 38.045, reflecting minimal error between the predicted and ground truth images. The AFI band shows a notably low RMSE of 0.0139 and a high SSIM of 0.969, suggesting strong preservation of structural details. Although the AFI PSNR is lower at 17.385, its significantly reduced RMSE relative to MWIR and LWIR shows the model's ability to produce precise pixel-level predictions, which is beneficial for fire hotspot detection. The R^2 values are consistently high across all bands 0.999 for MWIR, 0.999 for LWIR, and 0.999 for AFI indicating that the model's predictions are nearly perfectly correlated with the target VIIRS sensor observations. Overall, these results confirm the effectiveness of the model in capturing both pixel-wise accuracy and structural similarity across all three bands.

It is important to note that the divergence in AFI's RMSE and PSNR metrics compared to MWIR and LWIR stems from the nature of AFI as a normalizing index, which is designed to suppress values in non-burning regions while enhancing contrast in burning areas. The lower PSNR observed in non-fire AFI images is primarily due to the inherently low variability of AFI values in these regions. Since AFI suppresses responses in non-burning areas, the resulting images tend to be flat and lack significant contrast or detail, with values clustered within a narrow range (typically between -1 and 2). PSNR can be sensitive to the range and variance of pixel values, so when variability is minimal, even small errors can disproportionately impact the PSNR calculation. As a result, the PSNR metric may appear lower despite the predictions being very close to the ground truth, simply because the underlying signal lacks complexity.

Next, Figure 4.3.2 follows the same format as Figure 4.3.1 by exhibiting the original GOES ABI and VIIRS sensor in the first and second rows respectively, along with the super-resolved GOES ABI results produced by the model (third row). However, these images are from July 1st, 2024 (12:10 UTC).

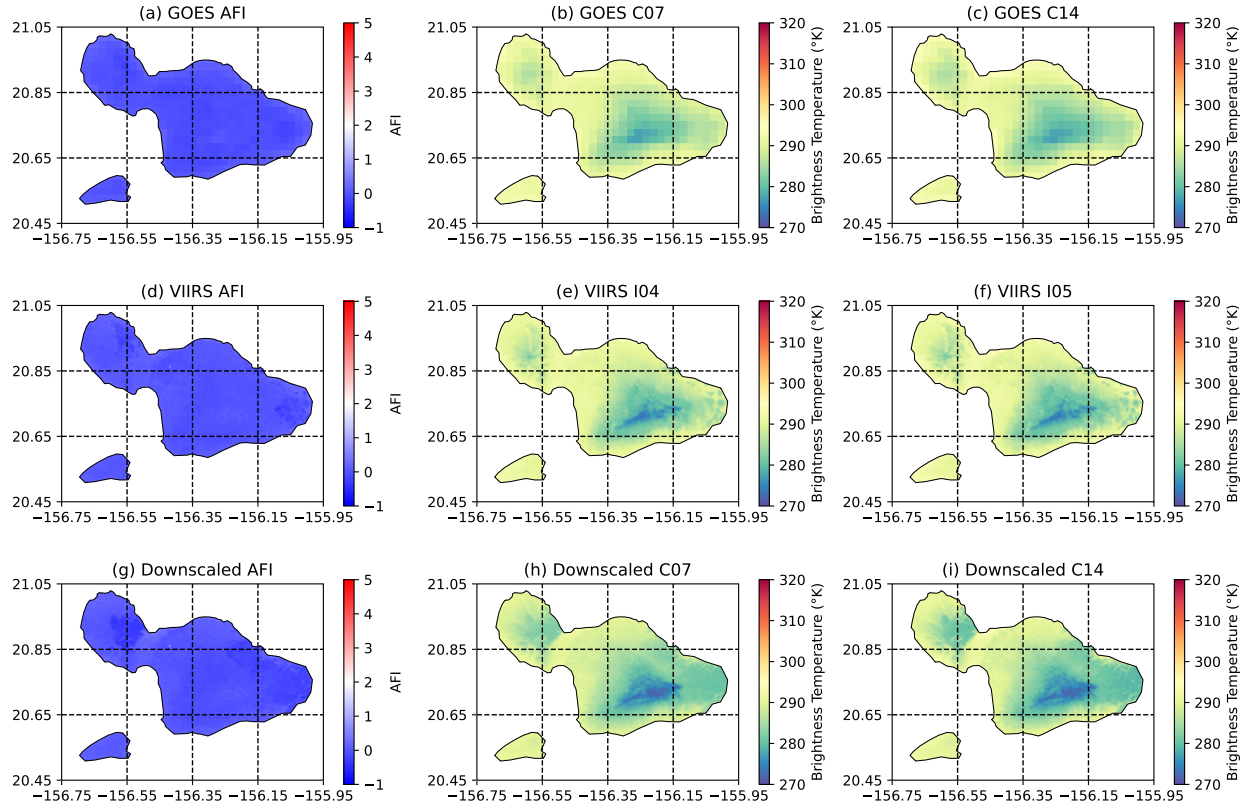


Figure 4.3.2: Testing Example From July 1st, 2024 (12:10 UTC): (a) GOES ABI AFI, (b) GOES ABI C07, (c) GOES ABI C14, (d) VIIRS AFI, (e) VIIRS I04, (f) VIIRS I05, (g) Downscaled AFI, (h) Downscaled C07, (i) Downscaled C14

Table 4.3.3: July 1st, 2024 (12:10 UTC) Testing Metrics

Band	RMSE	R ²	PSNR	SSIM
MWIR	0.311	0.999	36.811	0.992
LWIR	0.271	1	37.965	0.994
AFI	0.012	0.999	14.585	0.981

Based on the performance metrics presented in Table 4.3.3, the FireGAN continues to demonstrate strong performance across multiple output types, including spectral bands and the AFI. The MWIR band shows excellent reconstruction performance with an RMSE of 0.3111, a high PSNR of 36.811, and an SSIM of 0.9921. The LWIR band again yields the best overall results, with a

lower RMSE of 0.271, a near-perfect R^2 of 1, and the highest SSIM of 0.994, highlighting the model's ability to minimize reconstruction error and preserve fine spatial details. The AFI again exhibits a very low RMSE of 0.012 and a strong SSIM of 0.981, emphasizing its utility in retaining important fire-related features. Although the AFI PSNR is lower at 14.585 due to the nature of the AFI data, the extremely low RMSE indicates that the model effectively captures the pixel-level variations essential for accurate fire detection. The R^2 values across all bands 0.999 for MWIR and AFI, and 1 for LWIR reflect near-perfect correlation with the target VIIRS imager observations. These results reinforce the model's effectiveness in achieving both high quantitative accuracy and visual consistency across all output types.

4.4 Testing Maui, HW on August 9th, 2023

4.4.1 Full Maui Island Results

This section presents the super-resolved outputs generated by the FireGAN model for the full island of Maui, Hawaii, during the wildfire event on August 9th, 2023. The results span an 11-hour period, from 03:00 UTC to 14:00 UTC, with imagery captured at hourly intervals. These super-resolved images allow for a broad temporal analysis of the wildfire's progression across the island throughout the day.

The full-island results show the active fires that are otherwise challenging to fully detect in the original GOES ABI resolution. These outputs also serve as the foundation for more detailed, zoomed-in analyses of specific fire-affected zones, such as the Lahaina and central Maui regions, which are discussed in the following sections.

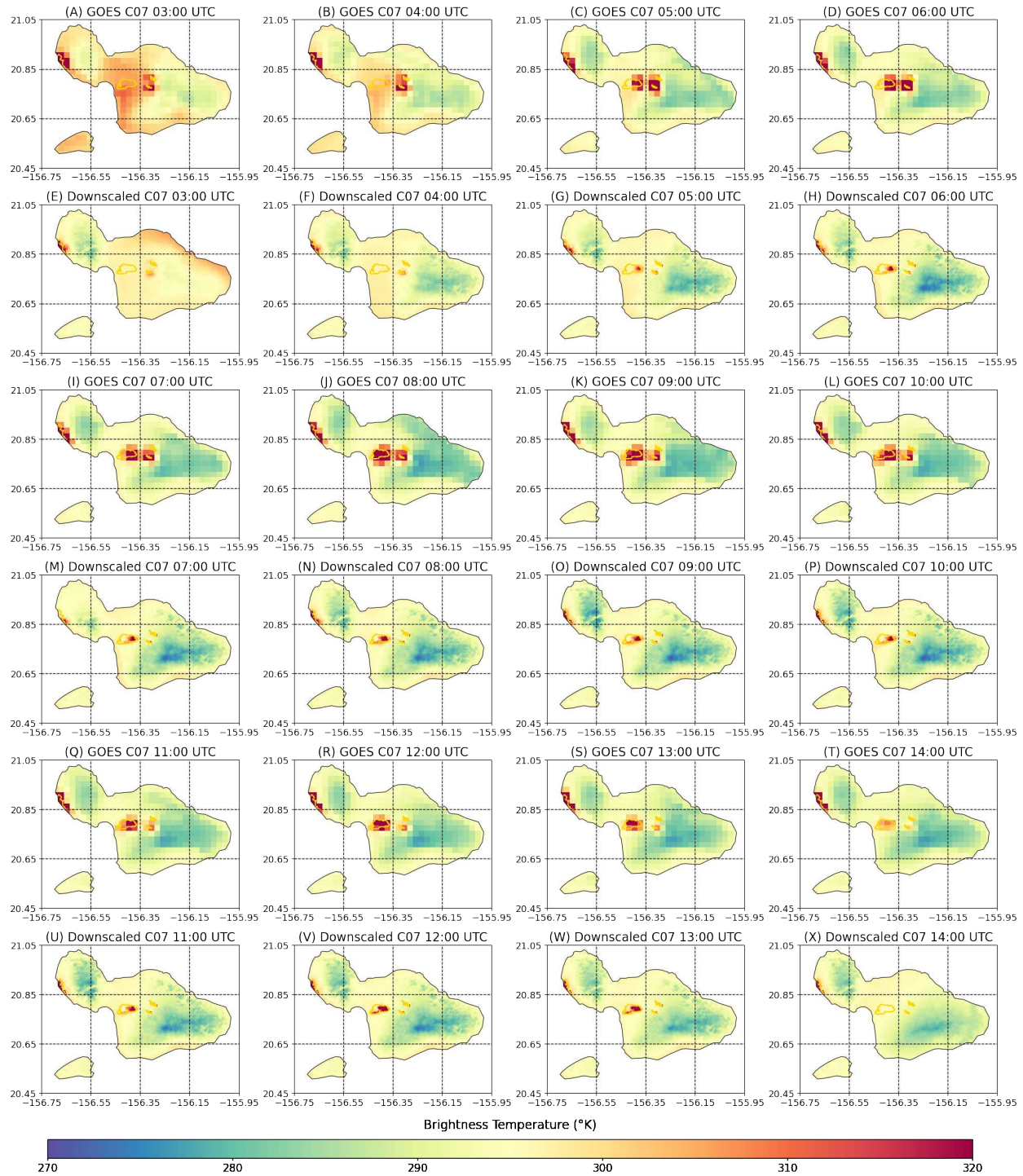


Figure 4.4.1: (a)-(d) GOES ABI C07 from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI C07 from 03:00 UTC to 06:00 UTC), (i)-(l) GOES ABI C07 from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI C07 from 07:00 UTC to 10:00 UTC), (q)-(t) GOES ABI C07 from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI C07 from 11:00 UTC to 11:00 UTC)

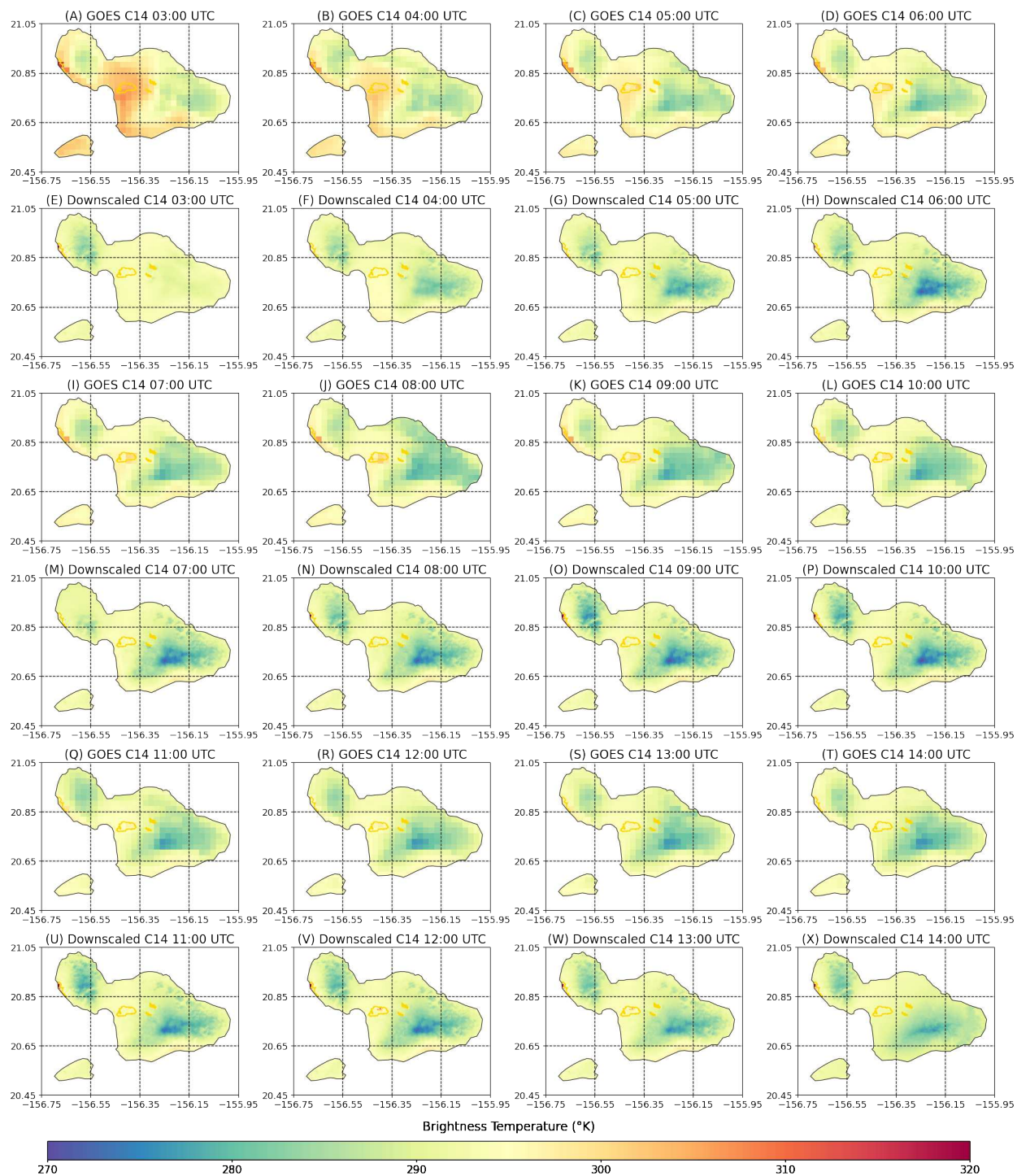


Figure 4.4.2: (a)-(d) GOES ABI C14 from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI C14 from 03:00 UTC to 06:00 UTC, (i)-(l) GOES ABI C14 from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI C14 from 07:00 UTC to 10:00 UTC, (q)-(t) GOES ABI C14 from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI C14 from 11:00 UTC to 11:00 UTC)

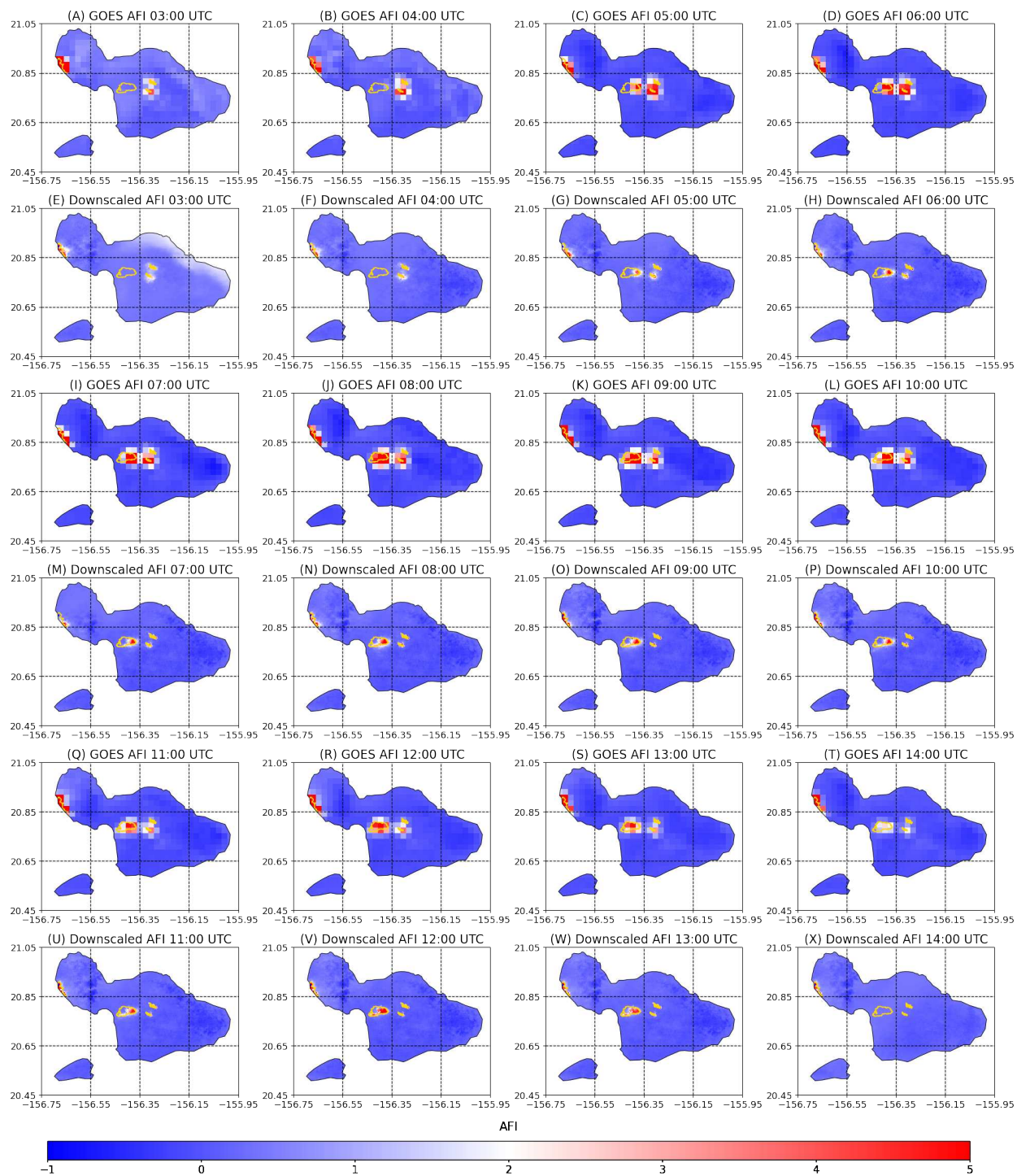


Figure 4.4.3: (a)-(d) GOES ABI AFI from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI AFI from 03:00 UTC to 06:00 UTC), (i)-(l) GOES ABI AFI from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI AFI from 07:00 UTC to 10:00 UTC), (q)-(t) GOES ABI AFI from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI AFI from 11:00 UTC to 11:00 UTC)

As shown in Figures 4.4.1, 4.4.2, and 4.4.3, the super-resolved images for C07, C14, and AFI illustrate the temporal evolution of the Maui wildfire event, providing high spatial-resolution reconstructions of GOES ABI imagery from 03:00 UTC to 14:00 UTC. Specifically, Figure 4.4.1 successfully identifies the wildfire hotspots across all times, along with the fires growth and shrinking throughout the day. The enhanced resolution allows for a clearer view of fire intensities and their movement, which are often difficult to distinguish in the original lower-resolution GOES ABI data. This is also clear with Figure 4.4.2 which identifies some high brightness temperature areas especially in the daytime like at 03:00 UTC where smoke is collecting. Last, Figure 4.4.3 isolates specific wildfire hotspots across the island. However there was some interference with the heightened brightness temperatures around the time 03:00 UTC, but not high enough to show a false positive.

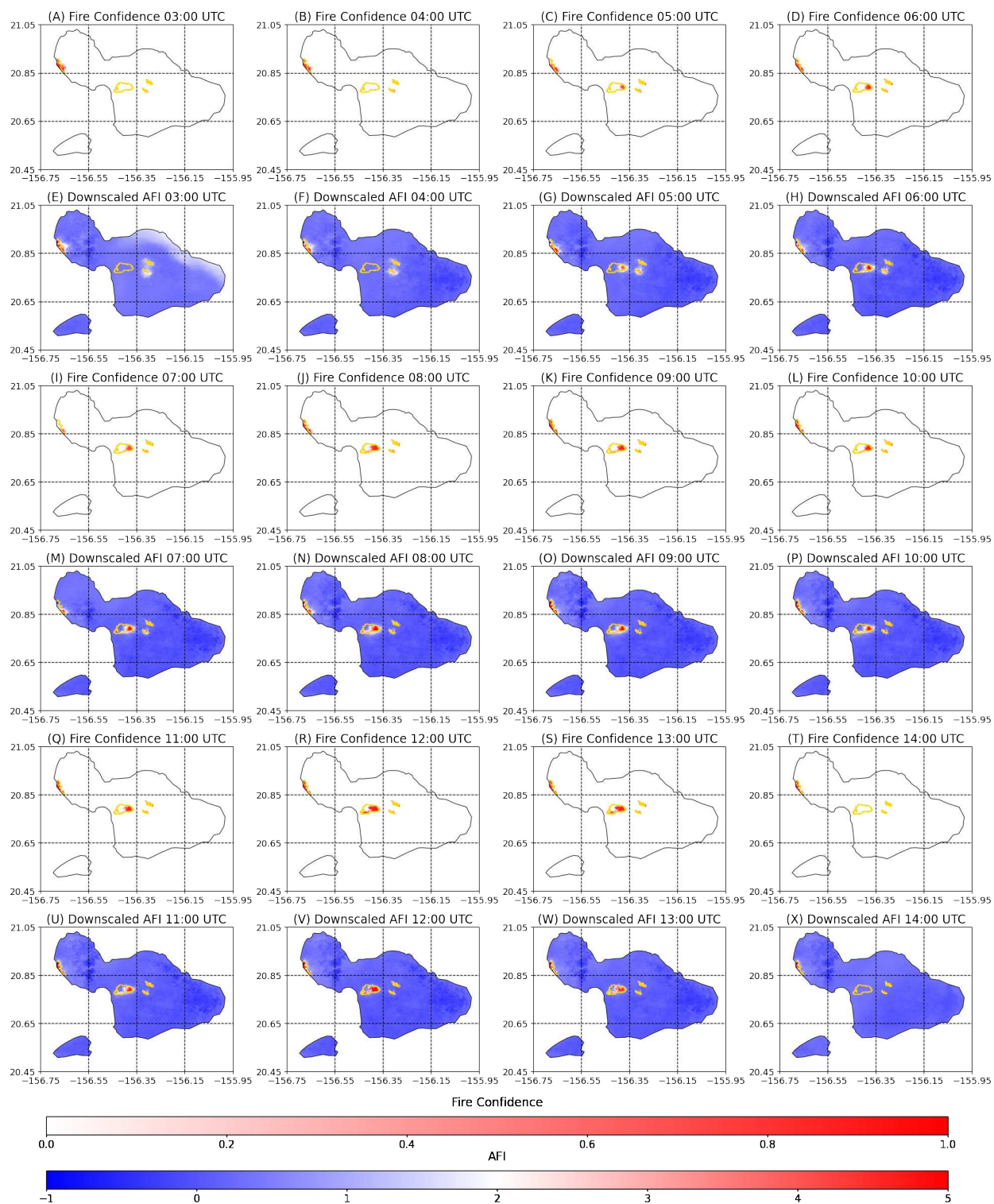


Figure 4.4.4: (a)-(d) GOES ABI AFI from 03:00 UTC to 06:00 UTC, (e)-(h) Fire Confidence from 03:00 UTC to 06:00 UTC, (i)-(l) GOES ABI AFI from 07:00 UTC to 10:00 UTC, (m)-(p) Fire Confidence from 07:00 UTC to 10:00 UTC, (q)-(t) GOES ABI AFI from 11:00 UTC to 14:00 UTC, (u)-(x) Fire Confidence from 11:00 UTC to 14:00 UTC

As shown in Figure 4.4.4, the super-resolved AFI images with Fire Confidence calculations applied demonstrate the accuracy of FireGAN’s fire detection, as the high-confidence regions generally align with the ground-based wildfire perimeters. This clearly shows the fires development through the 12 hours that were captured and how it was growing. Notably, the heightened AFI and C14 values around time 03:00 UTC did not impact the fire confidence calculation as is still only captured the hotspot within the ground based fire boundary.

4.4.2 Lahiana, Maui Zoomed in Results

The following results focus on a zoomed-in region of Maui Island, specifically highlighting the Lahaina wildfires. This localized view provides a more detailed assessment of FireGAN’s performance in detecting and super-resolving fire activity in one of the most severely impacted areas. The coordinates and boundaries of the zoomed-in region are defined as follows:

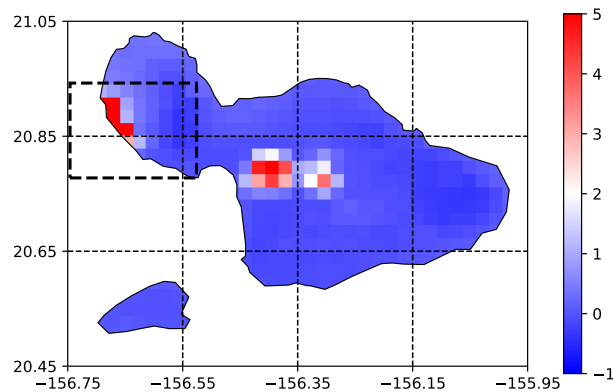


Figure 4.4.5: Zoomed in Lahiana, HW region

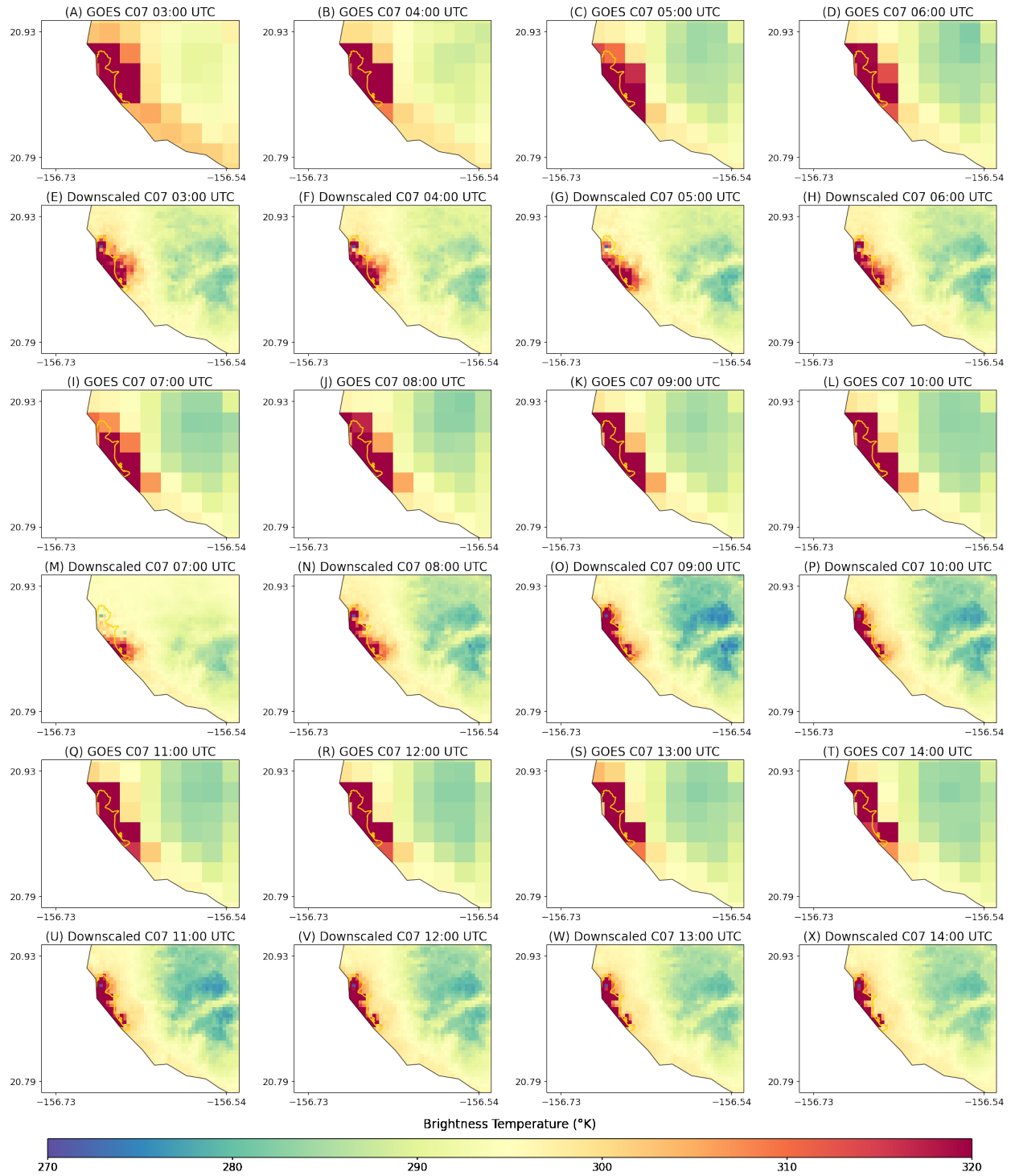


Figure 4.4.6: (a)-(d) GOES ABI C07 from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI C07 from 03:00 UTC to 06:00 UTC, (i)-(l) GOES ABI C07 from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI C07 from 07:00 UTC to 10:00 UTC, (q)-(t) GOES ABI C07 from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI C07 from 11:00 UTC to 14:00 UTC)

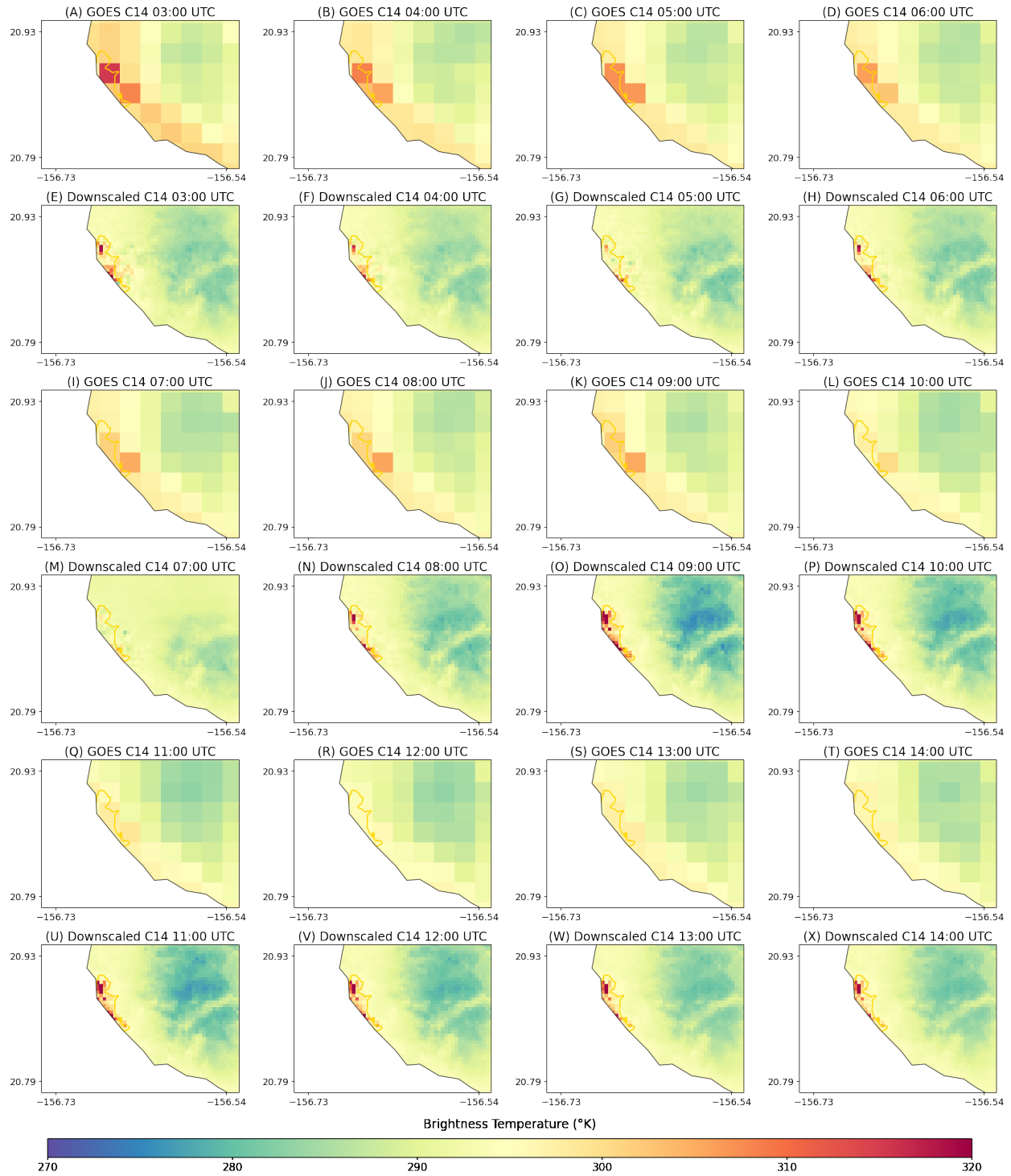


Figure 4.4.7: (a)-(d) GOES ABI C14 from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI C14 from 03:00 UTC to 06:00 UTC), (i)-(l) GOES ABI C14 from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI C14 from 07:00 UTC to 10:00 UTC), (q)-(t) GOES ABI C14 from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI C14 from 11:00 UTC to 11:00 UTC)

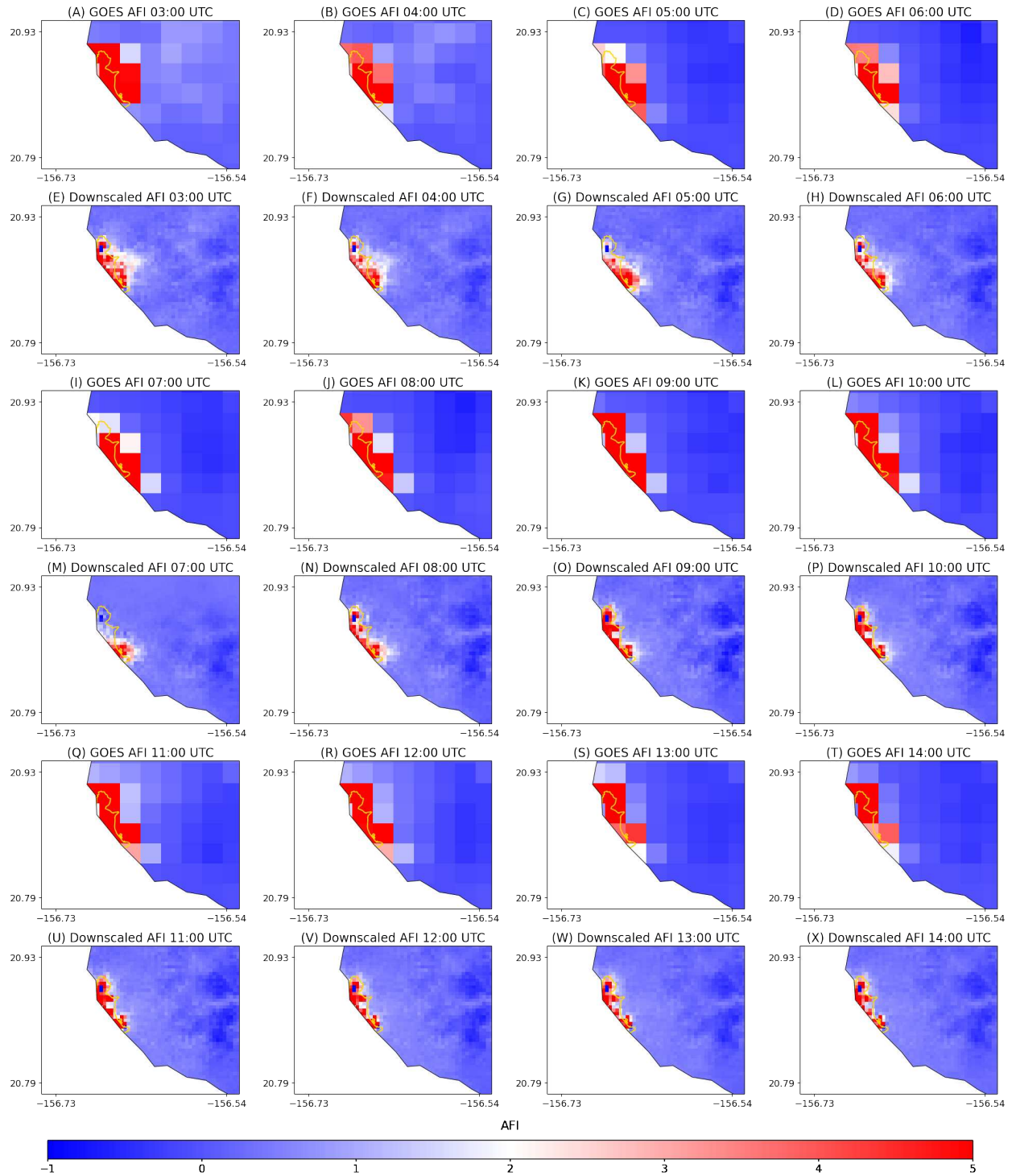


Figure 4.4.8: (a)-(d) GOES ABI AFI from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI AFI from 03:00 UTC to 06:00 UTC, (i)-(l) GOES ABI AFI from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI AFI from 07:00 UTC to 10:00 UTC, (q)-(t) GOES ABI AFI from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI AFI from 11:00 UTC to 14:00 UTC)

Similarly to the previous Brightness Temperatures and AFIIs shown for the full island, the zoomed Lahiana fires shown in Figures 4.4.6, 4.4.7, and 4.4.8 clearly show the fire's development from 03:00 UTC to 14:00 UTC. These images are accompanied by the ground-based wildfire boundary overlay, which serves to highlight the spatial accuracy of the super-resolved Brightness Temperatures (BTs) and AFIIs. Notably, there is visible growth of fire pixels extending beyond the ground-based fire boundary, particularly in Figures 4.4.6 and 4.4.8. While some fire pixels appear outside the reference boundary, this still represents a significant improvement over the original GOES ABI imagery, which exhibited even greater spatial inaccuracy. The super-resolved outputs demonstrate enhanced accuracy in locating general fire hotspots.

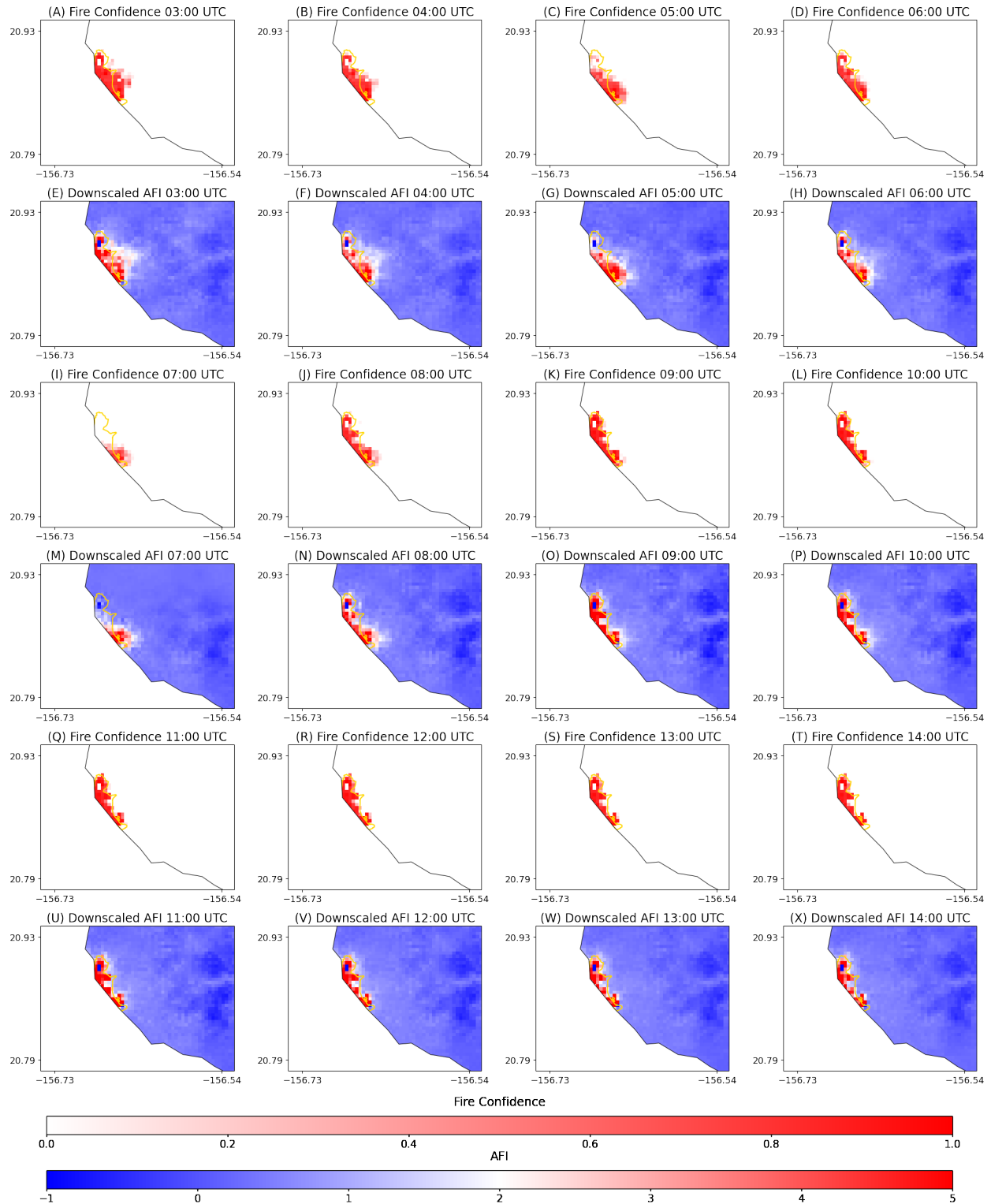


Figure 4.4.9: (a)-(d) GOES ABI AFI from 03:00 UTC to 06:00 UTC, (e)-(h) Fire Confidence from 03:00 UTC to 06:00 UTC, (i)-(l) GOES ABI AFI from 07:00 UTC to 10:00 UTC, (m)-(p) Fire Confidence from 07:00 UTC to 10:00 UTC, (q)-(t) GOES ABI AFI from 11:00 UTC to 14:00 UTC, (u)-(x) Fire Confidence from 11:00 UTC to 14:00 UTC

Next, the zoomed-in Lahaina fire results, shown in Figure 4.4.9, clearly demonstrate the evolution of fire product accuracy from 03:00 UTC to 14:00 UTC. The ground-based wildfire boundary is also overlaid to highlight the spatial accuracy of the super-resolved AFI, along with the corresponding Fire Confidence calculations. Similarly, around 03:00 and 04:00 UTC there was a set of false positive fire pixels that were outside of the fire boundary, but that region of false positives lowers in confidence in the following hours to < 0.3 . Within the fire boundary, areas with moderate confidence values ranging from 0.2 to 0.6 may indicate regions where the fire is either emerging or subsiding, potentially representing partially burning pixels. This information could be highly valuable to emergency responders in assessing the fire's progression and for strategizing localized containment efforts.

4.4.3 Center of Maui Island Results

For reference, the following results focus on a zoomed-in portion of Maui Island, highlighting the Pulehu, Olinda, and Kula wildfires, collectively referred to as the center fires to encompass the overall region. The zoomed area is as follows:

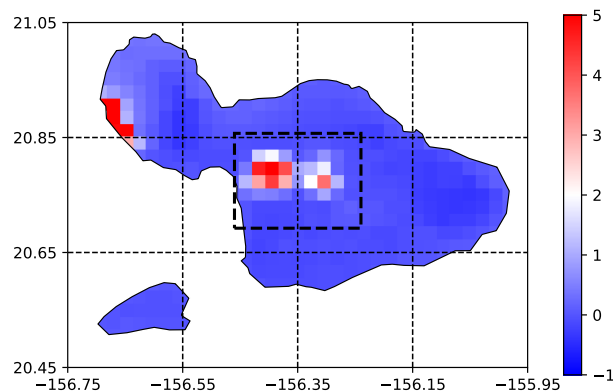


Figure 4.4.10: Zoomed in center region

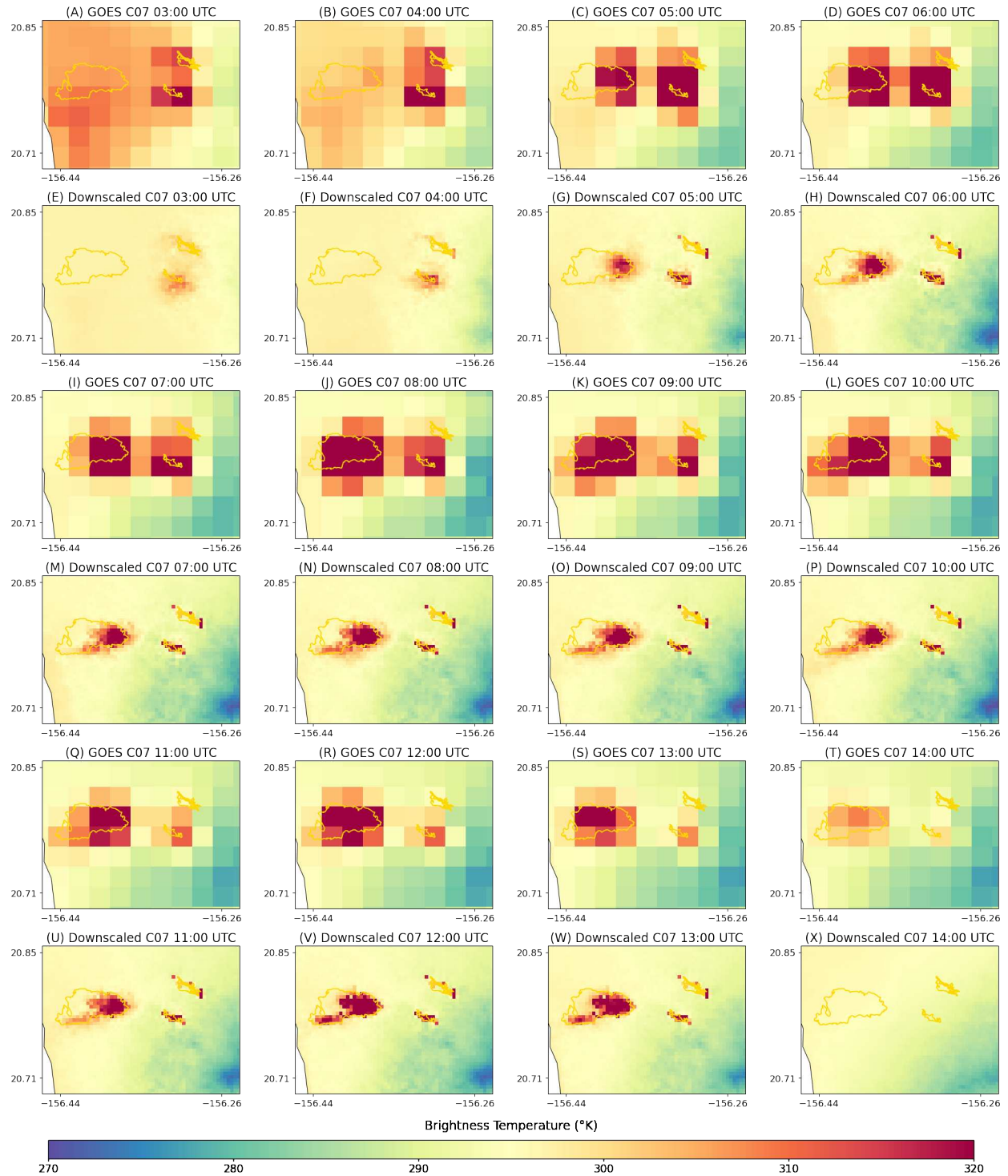


Figure 4.4.11: (a)-(d) GOES ABI C07 from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI C07 from 03:00 UTC to 06:00 UTC), (i)-(l) GOES ABI C07 from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI C07 from 07:00 UTC to 10:00 UTC), (q)-(t) GOES ABI C07 from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI C07 from 11:00 UTC to 14:00 UTC)

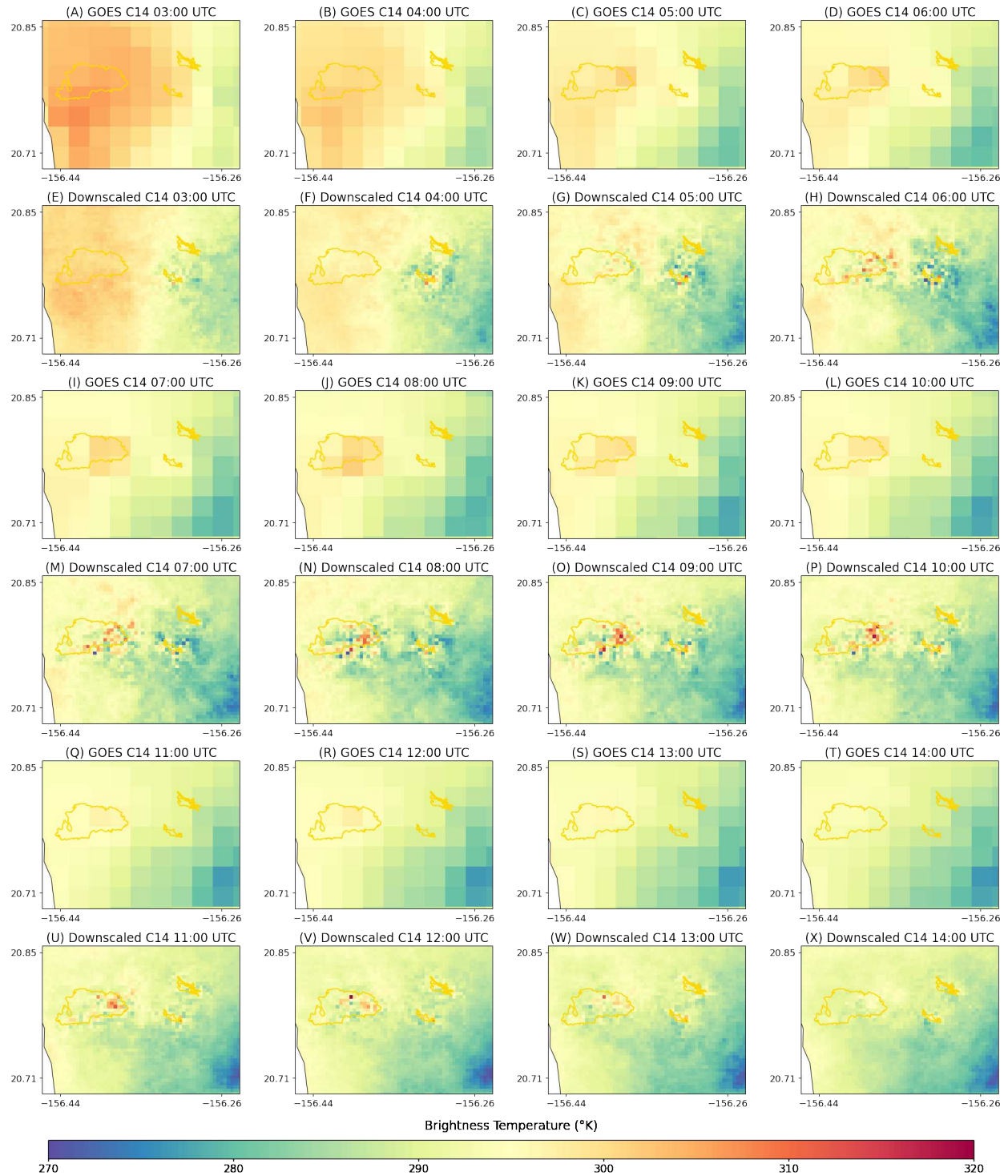


Figure 4.4.12: (a)-(d) GOES ABI C14 from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI C14 from 03:00 UTC to 06:00 UTC), (i)-(l) GOES ABI C14 from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI C14 from 07:00 UTC to 10:00 UTC), (q)-(t) GOES ABI C14 from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI C14 from 11:00 UTC to 11:00 UTC)

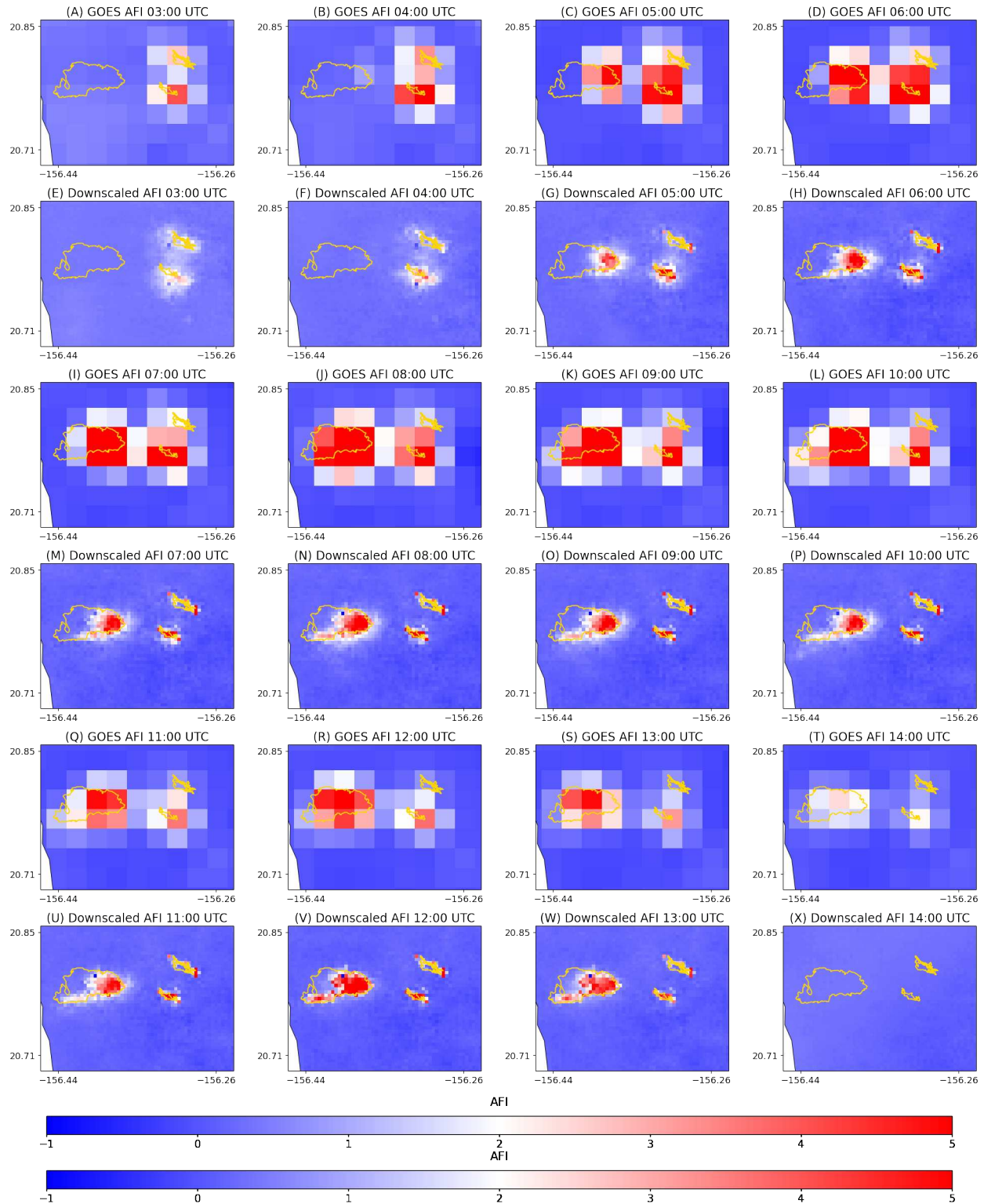


Figure 4.4.13: (a)-(d) GOES ABI AFI from 03:00 UTC to 06:00 UTC, (e)-(h) Downscaled GOES ABI AFI from 03:00 UTC to 06:00 UTC), (i)-(l) GOES ABI AFI from 07:00 UTC to 10:00 UTC, (m)-(p) Downscaled GOES ABI AFI from 07:00 UTC to 10:00 UTC), (q)-(t) GOES ABI AFI from 11:00 UTC to 14:00 UTC, (u)-(x) Downscaled GOES ABI AFI from 11:00 UTC to 14:00 UTC)

Figures 4.4.11, 4.4.12, and 4.4.13 display the Brightness Temperatures and AFIs for the central region of Maui Island, encompassing the Pulehu, Olinda, and Kula fires, over the period from 03:00 to 14:00 UTC. The largest of these fires, the Pulehu fire, exhibits elevated C07 and AFI values that extend beyond the established fire boundary, suggesting possible overprediction or fire spread. Notably, several misleading GOES ABI pixels are visible in Figure 4.4.11 between 06:00 and 10:00 UTC, where high brightness temperature values appear along the southeastern edge of the Pulehu fire. These pixels likely correspond to highly localized, intense heat sources that are small-scale fire activity. In contrast, the Olinda fire, the most northeastern of the three, suffers from under-representation. The generated C07 output fails to fully capture the burn area, indicating a lack of clear fire signal in that region. This represents the opposite issue observed with the Pulehu fire and highlights a key limitation of GOES ABI imagery: the need for a distinguishable hotspot for effective fire detection. However, even without a very distinguishable GOES ABI fire pixel for Olinda the FireGAN still shows some hotspots in that area.

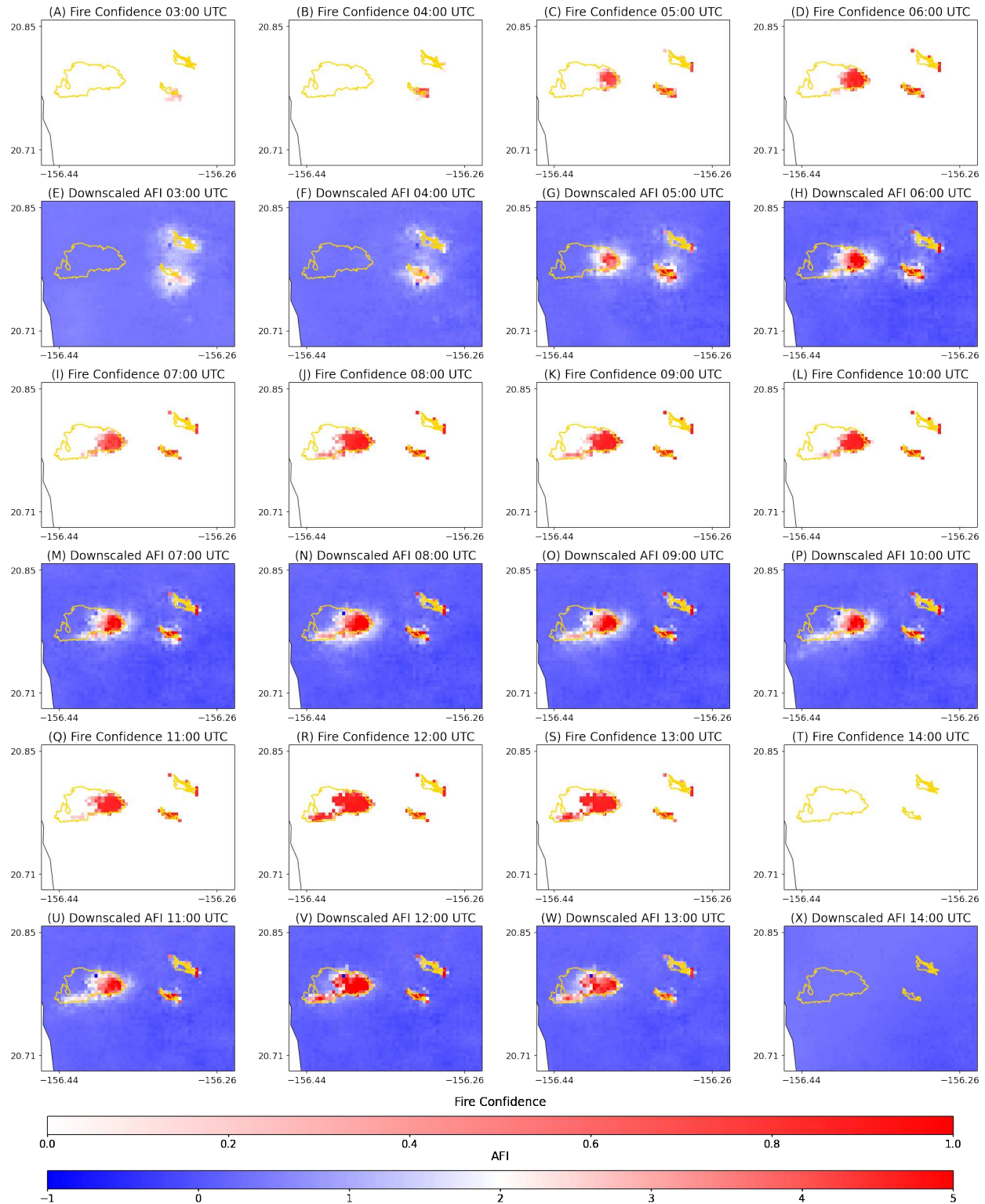


Figure 4.4.14: (a)-(d) GOES ABI AFI from 03:00 UTC to 06:00 UTC, (e)-(h) Fire Confidence from 03:00 UTC to 06:00 UTC, (i)-(l) GOES ABI AFI from 07:00 UTC to 10:00 UTC, (m)-(p) Fire Confidence from 07:00 UTC to 10:00 UTC, (q)-(t) GOES ABI AFI from 11:00 UTC to 14:00 UTC, (u)-(x) Fire Confidence from 11:00 UTC to 14:00 UTC)

Figure 4.4.14 shows the Fire Confidence and Downscaled AFI for the center region of Maui Island, spanning from 03:00 to 14:00 (UTC). The fire confidence makes up for many of the false positives shown in the previous images, showing its strength in identifying specific hotspot locations, including the edges with lower confidence values which are indicative of a fire's growth or shrinkage. This is especially noticeable from times 05:00 to 09:00 UTC where the Pulehu fire is rapidly growing, starting with lower values 0.35, then growing consistently over the following hours to an established and large fire hotspot.

4.5 Conclusion

In conclusion, the FireGAN model demonstrated its ability to effectively downscale GOES ABI C07, C14, and AFI imagery, regardless of a wildfire's presence. This capability highlights the strength of the model in generating GOES ABI imagery to 375 m regardless of Maui's conditions. This was shown with average RMSE values of 0.08 to 0.42, R^2 of 0.999 to 1, PSNR from 20dB to 40dB, and SSIM from 0.98 to 0.99. Overall, these results confirm FireGAN's effectiveness in producing high-resolution satellite imagery while preserving structural details as well.

When wildfires were present, the super-resolved images produced by FireGAN showed a high degree of spatial accuracy. This was validated through direct comparison of the predicted fire pixels with the VIIRS VNP14 fire product which resulted in HSS values of 0.839, CSI of 0.724, POD of 0.828, and FAR of 0.148. Which shows the model's ability to generate high resolution images that actually have to be reduced in resolution to be compared to VIIRS' fire product. Along with the FireGAN's ability to successfully generate 375 m images, the fire product can accurately identify wildfire locations, following GOES ABI's 10 minute temporal resolution which enables near-realtime active wildfire monitoring.

The results indicate that FireGAN not only enhances the spatial quality of GOES ABI imagery but also preserves and improves fire-related information. This makes it a valuable tool for supporting wildfire monitoring, management, and emergency response. The consistent and reliable outputs from 03:00 to 14:00 UTC demonstrate the model's ability to track fire development over time,

reinforcing its potential for real-time situational awareness and demonstrating the value of a high temporal resolution.

The following chapter will summarize the entire thesis, including what work was completed and what work could be done in the future.

Chapter 5

Summary and Future Work

5.1 Summary

This thesis explored the use of artificial intelligence (AI) for enhancing satellite imagery resolution, focusing on a deep learning-based approach, FireGAN. The model used images that were converted to brightness temperature from radiance, set to Maui, HI as the study domain, paired temporally similar GOES ABI and VIIRS sensor's images, and lastly, filtered through paired images with a correlation coefficient calculation so they are ready for training. From there the FireGAN was used to super-resolve GOES ABI 2 km imagery, including the C07 and C14 spectral bands, as well as their respective AFI calculations, to match the resolution of VIIRS sensor's 375 m data from Suomi-NPP, NOAA-20, and NOAA-21 satellites, creating a 6x improvement on GOES ABI spatial resolution while maintaining the 10 minute temporal resolution.

The effectiveness of FireGAN was demonstrated through the accurate reconstruction of satellite images, both in regions with and without wildfires. Over the entire test set, the model achieved: RMSE = 0.08-0.12, $R^2 = 0.99-1$, PSNR = 18-41dB, and SSIM = 0.98-0.99. These metrics confirm that FireGAN can reconstruct high-quality thermal imagery and preserve structural details. However, there is room for improvement around the PSNR metric since the FireGAN did a much better job at downscaling the AFI in comparison to C07 and C14 spectral bands, mainly due to the generalization that is produced by the AFI calculation in non-burning areas.

When evaluating fire detection performance using the VIIRS VNP14 product as ground truth for individual image comparisons: Probability of Detection (POD) = 0.839, False Alarm Rate (FAR) = 0.141, Heidke Skill Score (HSS) = 0.839, Critical Success Index (CSI) = 0.724. When aggregating results across all test cases from August 8–10, 2023: Probability of Detection (POD) = 0.819, False Alarm Rate (FAR) = 0.491, Heidke Skill Score (HSS) = 0.625, Critical Success Index (CSI)

= 0.457. The higher aggregated FAR suggests an accumulation of false detections over some areas, which points to a limitation in model generalization at a large scale.

Regardless, the model was still able to generate images every 10 minutes across August 8th through August 10th, 2023 which shows the high value it has in near-real time wildfire detection. Additionally, the model proved effective in temporal tracking of wildfires. For instance, the Maui wildfire case study demonstrated the model's ability to maintain consistent spatial accuracy from 03:00 to 14:00 UTC in the Testing Maui, HW on August 9th, 2023 section, also supporting real-time wildfire progression analysis.

5.2 Future Work

While this thesis lays a strong foundation for super-resolving wildfire imagery, several challenges remain that, if addressed, could further enhance the model's performance and broaden its applicability.

One potential area for improvement is the incorporation of additional datasets. By expanding the model's training data to include more diverse fire scenarios, varying weather conditions, and different geographic regions, the model's generalization could be improved. This would enable FireGAN to handle a wider range of wildfire types and environmental factors, enhancing its usefulness in other testing applications.

Another potential route involves integrating higher-resolution satellite data, such as Landsat, which offers a spatial resolution of 30 meters, being far higher than VIIRS' 375 meters. The inclusion of Landsat imagery could provide more spatially accurate fire locations, contributing to even more precise wildfire detection and mapping.

Additionally, the integration of predictive models that incorporate other environmental factors, such as wind speed and vegetation density, could enable the system to predict wildfire occurrence ahead of time. Models like U-Net or Convolutional LSTMs, as presented by [53], could be employed to forecast wildfires based on multiple physical factors, shifting from simply detecting ongoing fires to a proactive approach that predicts potential wildfires.

By addressing these challenges, future work could enhance AI-driven wildfire monitoring and prediction systems. Not only could this improve the spatial resolution of satellite-based fire detection systems, but it could also provide insights for predicting wildfire locations. This would allow authorities to take preventative measures, potentially reducing the devastating impact of wildfires on communities.

Bibliography

- [1] S. W. Running, “Is global warming causing more, larger wildfires?” *Science*, vol. 313, no. 5789, pp. 927–928, 2006.
- [2] C. O’Kane, “How much of maui has burned in the wildfires? aerial images show fire damage as containment efforts continue,” *CBS News*, 2023.
- [3] L. Giglio, W. Schroeder, and C. O. Justice, “The collection 6 modis active fire detection algorithm and fire products,” *Remote Sensing of Environment*, vol. 178, pp. 31–41, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425716300827>
- [4] E. P. S. L. Christopher C. Schmidt, Jay Hoffman, “Goes-r advanced baseline imager (abi) algorithm theoretical basis document for fire / hot spot characterization,” https://www.star.nesdis.noaa.gov/atmospheric-composition-training/documents/ABI_FDC_ATBD.pdf, 2020, nOAA NESDIS, Center for Satellite Applications and Research, Version 2.7, Published: October 2020.
- [5] N/A, “Nasa viirs land science investigator processing system (sips) visible infrared imaging radiometer suite (viirs) 375 m 750 m active fire products product user’s guide version 1.4,” https://lpdaac.usgs.gov/documents/427/VNP14_User_Guide_V1.pdf, 2024, nASA, Version 1.4.
- [6] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, “Esrgan: Enhanced super-resolution generative adversarial networks,” in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.
- [7] L. Cai, H. Gao, and S. Ji, “Multi-stage variational auto-encoders for coarse-to-fine image generation,” 2017. [Online]. Available: <https://arxiv.org/abs/1705.07202>

- [8] Clabaut, M. Lemelin, M. Germain, Y. Bouroubi, and T. St-Pierre, “Model specialization for the use of esrgan on satellite and airborne imagery,” *Remote Sensing*, vol. 13, no. 20, 2021. [Online]. Available: <https://www.mdpi.com/2072-4292/13/20/4044>
- [9] Z. Shao, L. Wang, Z. Wang, and J. Deng, “Remote sensing image super-resolution using sparse representation and coupled sparse autoencoder,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 2663–2674, 2019.
- [10] L. Salgueiro, J. Marcello, and V. Vilaplana, “Seg-esrgan: A multi-task network for super-resolution and semantic segmentation of remote sensing images,” *Remote Sensing*, vol. 14, no. 22, 2022. [Online]. Available: <https://www.mdpi.com/2072-4292/14/22/5862>
- [11] M. Badhan, K. Shamsaei, H. Ebrahimian, G. Bebis, N. P. Lareau, and E. Rowell, “Deep learning approach to improve spatial resolution of goes-17 wildfire boundaries using viirs satellite data,” *Remote Sensing*, vol. 16, no. 4, 2024. [Online]. Available: <https://www.mdpi.com/2072-4292/16/4/715>
- [12] Y. Zhao and Y. Ban, “Goes-r time series for early detection of wildfires with deep gru-network,” *Remote Sensing*, vol. 14(17), p. 4347, 09 2022.
- [13] B. Nelson, “Landsat timeline,” <https://landsat.gsfc.nasa.gov/satellites/timeline/>, 2025.
- [14] Anonymous, “History of noaa satellites,” <https://www.nesdis.noaa.gov/our-satellites/related-information/history-of-noaa-satellites>, 2025.
- [15] M. Gure, M. E. Ozel, H. H. Yildirim, and M. Ozdemir, “Use of satellite images for forest fires in area determination and monitoring,” in *2009 4th International Conference on Recent Advances in Space Technologies*, 2009, pp. 27–32.
- [16] P. Gupta and M. Follette-Cook, “Fundamentals of satellite remote sensing,” https://appliedsciences.nasa.gov/sites/default/files/D1P3_Fundamentals.pdf, 2018.

- [17] X. Yang and B. Wei, “Exact research on the theory of the blackbody thermal radiation,” *Scientific Reports*, vol. 6, p. 37214, Nov 2016.
- [18] N/A, “Blackbody radiation,” [https://phys.libretexts.org/Bookshelves/University_Physics/University_Physics_\(OpenStax\)/University_Physics_III_-_Optics_and_Modern_Physics_\(OpenStax\)/06\%3A_Photons_and_Matter_Waves/6.02\%3A_Blackbody_Radiation](https://phys.libretexts.org/Bookshelves/University_Physics/University_Physics_(OpenStax)/University_Physics_III_-_Optics_and_Modern_Physics_(OpenStax)/06\%3A_Photons_and_Matter_Waves/6.02\%3A_Blackbody_Radiation), 2023, libreTexts, University Physics (OpenStax).
- [19] —, “Abi bands quick information guides,” <https://www.goes-r.gov/mission/ABI-bands-quick-info.html>.
- [20] —, “Visible infrared imaging radiometer suite (viirs),” <https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/viirs/>.
- [21] A. Rosebrock, “Imagenet: Vggnet, resnet, inception, and xception with keras,” <https://pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>, 2017, published: March 20, 2017.
- [22] D. Krause, “Alexander mountain fire fully contained after burning for nearly 3 weeks west of loveland,” <https://coloradosun.com/2024/08/18/alexander-mountain-fire-contained-colorado-wildfire/>, 2024, the Colorado Sun, August 18, 2024.
- [23] N/A, “Smith river complex,” <https://www.fire.ca.gov/incidents/2023/8/15/smith-river-complex>, 2023.
- [24] —, “Viirs - level-1 / level-0,” <https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/science-domain/viirs-L0-L1/>, N/A.
- [25] NASA Earthdata, “Fire information for resource management system (firms),” <https://firms.modaps.eosdis.nasa.gov/map/>, 2024, accessed: 2024-07-16.
- [26] N/A, “Turing test in artificial intelligence.”

- [27] B. Kumar, “Convolutional neural networks: A brief history of their evolution,” <https://medium.com/appyhigh-technology-blog/convolutional-neural-networks-a-brief-history-of-their-evolution-ee3405568597>, 2021, appyHigh Blog, Published: August 31, 2021.
- [28] G. Hinton, “Boltzmann machines,” in *Encyclopedia of Machine Learning and Data Mining*. New York: Springer Science+Business Media, 2014. [Online]. Available: <http://people.stat.sfu.ca/~dac5/BoltzmannMachines.pdf>
- [29] J. Schmidhuber, “Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991),” *Neural Networks*, vol. 127, pp. 58–66, 2020, preprint available at <https://arxiv.org/pdf/1906.04493>.
- [30] N/A, “Generative adversarial network (gan),” <https://www.geeksforgeeks.org/generative-adversarial-network-gan/>, 2025, last Updated: March 10, 2025.
- [31] Amanrao, “Image upscaling using bicubic interpolation,” <https://medium.com/@amanrao032/image-upscaling-using-bicubic-interpolation-ddb37295df0>, 2023, medium, Published: September 13, 2023.
- [32] H. Mansourifar, A. Moskovitz, B. Klingensmith, D. Mintas, and S. J. Simske, “Gan-based satellite imaging: A survey on techniques and applications,” *IEEE Access*, vol. 10, pp. 118 123–118 140, 2022.
- [33] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016. [Online]. Available: <https://arxiv.org/pdf/1501.00092>
- [34] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, “Esrgan: Enhanced super-resolution generative adversarial networks,” 2018. [Online]. Available: <https://arxiv.org/abs/1809.00219>

- [35] N/A, “Generative adversarial network (gan),” <https://www.geeksforgeeks.org/generative-adversarial-network-gan/>, 2025, geeksforGeeks, Last Updated: March 10, 2025.
- [36] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014. [Online]. Available: <https://arxiv.org/abs/1406.2661>
- [37] F. Tariq, “Breaking down the mathematics behind cnn models: A comprehensive guide,” <https://medium.com/@beingfarina/breaking-down-the-mathematics-behind-cnn-models-a-comprehensive-guide-1853aa6b011e>, 2023, medium, Published: May 2, 2023.
- [38] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472–2481.
- [39] S. Rallabandi, “Activation functions: Relu vs. leaky relu,” <https://medium.com/@sreeku.ralla/activation-functions-relu-vs-leaky-relu-b8272dc0b1be>, 2023, medium, Published: March 25, 2023.
- [40] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” 2018. [Online]. Available: <https://arxiv.org/abs/1611.07004>
- [41] U. Demir and G. Unal, “Patch-based image inpainting with generative adversarial networks,” 2018. [Online]. Available: <https://arxiv.org/abs/1803.07422>
- [42] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015. [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [43] Serp AI, “Leaky relu,” <https://serp.ai/posts/leaky-relu/>, 2023, accessed: 2025-04-16.

- [44] N/A, “Smokehouse creek fire,” https://en.wikipedia.org/wiki/Smokehouse_Creek_Fire, 2025, published: May 20, 2025.
- [45] S. Bangar, “Vgg-net architecture explained,” <https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f>, 2022, medium, Published: June 28, 2022.
- [46] N/A, “Bcewithlogitsloss,” <https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html>, 2024, © Copyright 2024, PyTorch Contributors.
- [47] C. Sekhar and P. Meghana, “A study on backpropagation in artificial neural networks,” *Asia-Pacific Journal of Neural Networks and Its Applications*, vol. 4, pp. 21–28, 08 2020.
- [48] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [49] Z. Bobbitt, “How to interpret root mean square error (rmse),” <https://www.statology.org/how-to-interpret-rmse/>, 2021, statology, Published: May 10, 2021.
- [50] N/A, “R squared | coefficient of determination,” <https://www.geeksforgeeks.org/r-squared/>, 2024, geeksforGeeks, Last Updated: September 25, 2024.
- [51] —, “Python | peak signal-to-noise ratio (psnr),” <https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/>, 2020, geeksforGeeks, Last Updated: February 6, 2020.
- [52] H. R. S. E. P. S. Zhou Wang, Alan C. Bovik, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, 2004, iEEE, Published: 2004. [Online]. Available: <https://ece.uwaterloo.ca/~z70wang/publications/ssim.pdf>
- [53] N. Alizadeh, M. Mahdianpari, E. Hemmati, and M. Marjani, “Fusionfirenet: A cnn-lstm model for short-term wildfire hotspot prediction utilizing spatio-temporal datasets,” *Remote*

Sensing Applications: Society and Environment, vol. 37, p. 101436, 2025. [Online].
Available: <https://www.sciencedirect.com/science/article/pii/S2352938524003008>