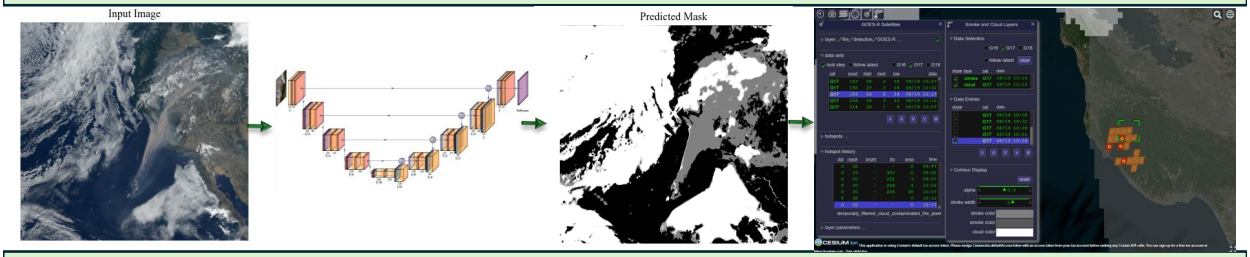
Graphical Abstract

Smoke or Cloud: Real-time Satellite Image Segmentation in a Wildfire Data Integration Application

Sequoia Andrade, Nastaran Shafiei, Peter Mehlitz

Smoke or Cloud: Real-time Satellite Image Segmentation in an Application for Wildfire Data Integration

- Differentiating smoke from cloud has the potential to improve the accuracy of real-time hot spot detection
- We pre-train and fine-tune a deep learning model for this task and deploy it in a real-time application



True color smoke and cloud detection deep learning model scores 85% accuracy, with real-time implementation from data download to in browser display taking 74 seconds.

Email addresses: sequoia.r.andrade@nasa.gov (Sequoia Andrade), peter.c.mehlitz@nasa.gov (Peter Mehlitz)

Highlights

Smoke or Cloud: Real-time Satellite Image Segmentation in a Wildfire Data Integration Application

Sequoia Andrade, Nastaran Shafiei, Peter Mehlitz

- Development and implementation of a smoke-cloud multi-class image segmentation model for use on GOES-R True Color images.
- Builds, trains, and evaluates a self-supervised pre-trained and fine-tuned U-Net for smoke and cloud segmentation with 85% accuracy.
- Implements the model in an application for real-time wildfire data integration.
- Demonstrates the validity of the model through a case study of the 2020 CZU wildfire complex.

Smoke or Cloud: Real-time Satellite Image Segmentation in a Wildfire Data Integration Application

Sequoia Andrade[a,*], Nastaran Shafiei[b], Peter Mehlitz[b]

[a]HX5, NASA Ames Research Center, Building 269, Moffett Field, 94035, CA, USA
[b]KBR, NASA Ames Research Center, Building 269, Moffett Field, 94035, CA, USA

Abstract

Advanced satellite data is increasingly used for wildfire detection and monitoring, yet near real-time hotspot data products from the GOES-R series often have low confidence due to aerosol contamination. Since aerosol contamination impacts the confidence of the GOES-R hot spot detection algorithm, regardless of contamination from fire-indicating smoke or false positive-indicating clouds, differentiating smoke from cloud has the potential to improve the accuracy of real-time hot spot detection. The primary contribution of this paper is a multi-class smoke and cloud segmentation model that classifies smoke, cloud, and neither pixels from GOES-R true color images in a real-time application. When selecting the final model, we perform an experiment to examine the impact self-supervised learning has on different model architectures. The final model is a U-Net model pre-trained on over 10,000 images using Barlow Twins self-supervised learning and fine-tuned using supervised learning, which exhibits comparable performance to the larger and slower ResUnet model. Our model improves upon existing satellite-based smoke segmentation, with 85% accuracy and 68% mean intersection-over-union on the test set. The model is deployed in an Open Data Integration for wildfire management (ODIN) application, allowing for real-time smoke and cloud detection to improve situational awareness regarding smoke location. From real-time image import to smoke-cloud segmentation display in the browser, the total run time is approximately 74 seconds, with 52 seconds total from the segmentation model pipeline.

Keywords: Machine Learning, Satellite Remote Sensing, Wildfire, Image Segmentation, Data Integration

1. Introduction

As wildfires increase in frequency and severity (Abatzoglou and Williams, 2016; Dennison et al., 2014), there has been a corresponding increase in harnessing real-time data to improve wildfire response (Crowley et al., 2023). Geostationary satellites, such as the Geostationary Operational Environmental Satellites R series (GOES-R), which is the newest series of GOES, provide updated imagery every five minutes over large areas, making them ideal for early fire detection. Existing fire detection products, including the hot spot detection provided by the GOES-R Advanced

*Corresponding Author
Email addresses: sequoia.r.andrade@nasa.gov (Sequoia Andrade), peter.c.mehlitz@nasa.gov (Peter Mehlitz)

Baseline Imager (ABI), have decreased confidence due to cloud or smoke contamination (Xu et al., 2021; Hall et al., 2019). However, incidents of smoke are clear indicators of a fire, whereas incidents of cloud could be indicators of a false positive. Identifying which regions have smoke versus cloud can increase the confidence in fire detection from existing GOES-R ABI products. Thus, there is an opportunity to increase the confidence and accuracy of real-time space-based fire detection by segmenting smoke from clouds.

The main contributions of this research are two-fold. First, we explore the feasibility of separating clouds from smoke in true color satellite imagery via building and training a segmentation model. The second contribution is the implementation of the model in a real-time wildfire data integration application. . To this end, we build a labeled dataset of clouds and smoke in satellite imagery, explore different configurations of U-Net (Long et al., 2015; Ronneberger et al., 2015) segmentation models, and deploy the final model in a real-time application for wildfire response data integration, known as Open Data Integration for wildland fire management (ODIN) (Mehlitz, 2022b). A novel component of the machine learning model development is the evaluation of large-scale self-supervised pre-training on model performance, in turn demonstrating self-supervised learning can bring smaller models, such as the simple U-Net, up to par with larger models, such as the ResUnet. As a result, a multi-class pre-trained and fine-tuned U-Net model is chosen as the final model for its robust performance on image segmentation tasks (Mo et al., 2022; Yuan et al., 2021), with the model exhibiting an accuracy of 85% on the test set. Our model not only outperforms an existing smoke segmentation model with an average intersection-over-union of 68% compared to 57.6% (Larsen et al., 2021), but also segments clouds from smoke unlike existing models. The ODIN architecture is used for implementation due to its concurrent, distributed actor system (Mehlitz, 2022b), which supports both real-time data integration from numerous sources, including satellite imagery imports, and integration with the smoke segmentation model. The model enables the transformation of unstructured satellite imagery into concretely classified pixels, which could be integrated with existing real-time fire detection products to improve hot spot detection accuracy.

The following section provides an overview of the motivating background including satellite-based fire detection, existing research on smoke segmentation, and a brief description of the ODIN framework which is used to implement the model. Next, we describe the smoke-cloud segmentation model, including the dataset creation, training procedure, model selection, and implementation. A demonstration of the model is provided via a case study of the 2020 CZU wildfire.

2. Background

2.1. Satellite-based fire detection

Wildfire events can be monitored via space-based satellites equipped with specialized sensors, where satellites have differing spatio-temporal resolution. Two classes of satellites are geostationary satellites, which remain stationed above Earth in a fixed location relative to the ground, and polar orbiting satellites, which orbit Earth along its polar axis. Due to the orbital path, geostationary satellites produce data with high temporal resolution on a scale of minutes and lower spatial resolution at a scale of a few kilometers. In contrast, polar orbiting satellites only provide data a

couple times a day (i.e., temporal resolution on the scale of hours) at a higher spatial resolution (e.g., 750 km). Satellites that are capable of monitoring wildfires in the western United States include the GOES-R series (NESDIS, 2013) and the polar-orbiting Joint Polar Satellite System (JPSS) (DelFierro and Kilcoyne, 2019), both operated jointly by the National Oceanic and Atmospheric Administration (NOAA) and National Aeronautics and Space Administration (NASA).

Polar orbiting satellite hot spot detection is available via the Moderate Resolution Imaging Spectroradiometer (MODIS) (Justice et al., 2002) instrument on Terra and Aqua satellites and the Visible Infrared Imaging Radiometer Suite (VIIRS) (Schroeder et al., 2014) instrument on JPSS satellites. When compared to GOES-R, the hot spot detection from polar orbiting satellites has higher spatial resolution and higher accuracy. However, polar orbiting satellites only provide data about twice a day and thus cannot be used for near-real time fire monitoring. In contrast, GOES-R provides regular data updated every five minutes and is more suitable for near-real time fire monitoring. Despite having high temporal frequency, even with algorithmic improvements GOES-R hot spot detection fails to detect 68% of low intensity fires when compared to MODIS and incorrectly detects false fires 12% of the time when compared to MODIS (Xu et al., 2021), with higher error rates observed from the baseline algorithm (Hall et al., 2019). This is a major issue: the only satellite usable for near-real time fire monitoring has unsatisfactory performance. At this time, the algorithms used for hot spot monitoring do not leverage machine learning, thus there is an opportunity to augment the existing systems with machine learning in hopes of improving performance. Cloud contamination decreases the confidence in a prediction (Xu et al., 2021; Hall et al., 2019), while smoke is a clear indicator of active fire. Hence, segmenting smoke from cloud in GOES satellite images could improve the accuracy of hot spot detection by reducing omission errors in cases of smoke and decreasing commission errors in cases of clouds.

2.2. Smoke Segmentation

Previous machine learning applications have successfully built image processing convolutional neural networks to detect smoke within a home(Park et al., 2019), as well as trained long-short term memory networks (Jeong et al., 2020) to detect smoke from stationary sensors. Non-neural network based methods, including random forests (Ko et al., 2012) and maximally stable extremal regions (Zhou et al., 2016), have also been used to detect wildfire smoke from ground-based sensors. In 2021, Frizzi et al. built a smoke and fire segmentation model with over 98% accuracy (Frizzi et al., 2021); however, the model is only suitable for on ground image processing and fails to account for clouds. U-Net transformer models have been used for cloud classification and segmentation and exhibit high performance (83% mean IOU) (Roy et al., 2021) compared to standard U-Nets(76% mean IOU), yet smoke is not accounted for in these models. Additionally, Transformer U-Nets and vision transformers in general are larger models than standard U-Nets, thus it is worth examining how to improve standard U-Net accuracy in compute-constrained scenarios. Other successful smoke segmentation models have been trained primarily on synthetic data (Yuan et al., 2019; Yan et al., 2022). Recently, a cubic-cross convolutional attention head has been applied in a pipeline with ResNet and achieved smoke segmentation with a mean intersection-over-union (IoU) over 75% (Yuan et al., 2022). Yan, Zhang, and

Barnes created a Bayesian generative model with a transmission guided loss to improve smoke segmentation in areas of ambiguity, resulting in improved performance compared to other deep learning models (Yan et al., 2022).

While these methods perform well in their respective applications, they have not been extended to detect smoke from satellite imagery and thus can only provide fire detection in the limited region of in-home and stationary sensors. Additionally, smoke segmentation research has largely focused on ground-based sensors rather than satellite imagery, with satellite imagery being inherently different due to the change in scale, spatial and temporal resolution, and impact of clouds. Larsen et al. developed and trained a fully convolutional neural network to segment smoke plumes in satellite data, yet they did not account for clouds and did not use true color imagery (Larsen et al., 2021). There are two primary approaches to identifying smoke using machine learning: physics informed modeling and black-box modeling. Larsen et. al (Larsen et al., 2021) take an approach that is a mixture between the two by considering only specific spectral bands that are related to the physics of smoke while simultaneously using a large physics-blind convolutional neural network. However, true color images are typically composed of a subset of bands that contain both physical smoke information and additional visual information. Thus using true color images rather than only specific smoke-relevant bands moves more towards the black-box method, while still retaining the physical information of smoke present in the visual light spectrum. Li et. al. used a modified YOLOv5 model on Sentinel-2 data to identify the red, green, and blue spectra alongside the 865 nanometer infrared band (band 8a) as the most useful band combination for differentiating smoke from clouds over varied background of bodies of water and vegetationLi et al. (2023). Hence, the true color spectrum can be useful in distinguishing smoke from cloud, especially with an additional infrared band. While Sentinel-2 has high spatial resolution, its low temporal resolution of approximately one pass per every five days makes it impractical for real-time smoke identification. To this end, this paper examines the performance of deep learning methods, and in particular the U-Net, when applied to the satellite true color imagery domain. Additionally, we aim to close the performance gap between standard U-Nets and larger models by investigating the effect of self-supervised pre-training methods. The existing body of research clearly demonstrates that it is possible to segment smoke pixels using deep learning methods, yet there is a gap in implementing these algorithms into real world applications.

2.3. Open Data Integration Framework for Wildland Fire Management (ODIN)

ODIN is a framework that allows for creating real-time, distributed, wildfire data integration applications to support situational awareness and decision making. Figure 1 shows the ODIN framework infrastructure and the existing systems incorporated into ODIN. It is build on top of Runtime for Airspace Concept Evaluation (RACE) (Mehlitz et al., 2016; Mehlitz, 2022a), which is a framework for building configurable, highly concurrent and distributed message-based systems. RACE uses the Akka toolkit (Roestenburg et al., 2015; Akka) that provides an implementation of the actor programming model in Scala. The actor model was proposed in 1973 as a way to deal with concurrency in high performance systems (Hewitt et al., 1973). The primary building blocks in the actor programming model are actors, which are concurrent objects that do not share states and only communicate by means of

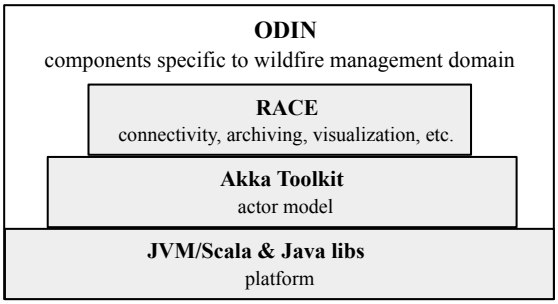


Figure 1: The ODIN framework builds off RACE, which utilizes the Akka toolkit in Scala.

asynchronous messages. Actors are fully independent and autonomous and only become runnable when they receive a message in their buffer. The RACE framework offers a wide variety of actors to import, translate, filter, archive, replay, and visualize live data feeds. ODIN extends RACE towards the application domain of wildfire management. Both RACE and ODIN are open source frameworks written in Scala and publicly available.[1]. ODIN provides specialized import actors to integrate real-time data sources from numerous existing third party web services, including terrain, infrastructure, Delphire fire detection sentinels (DeSalvo and Mehlitz, 2022), micro-wind fields, VIIRS hot spots, GOES hot spots, and tracking information about vehicles and aircrafts. ODIN applications are configurable and can be modified to include or exclude any actors as desired. Considering that machine learning applications require real-time data, ODIN is an ideal framework for deploying relevant models since it can integrate live data from many different sources and display model outputs in a unified interface.

3. Smoke-Cloud Segmentation Model

In this section, we provide details on the U-Net architecture, self-supervised pre-training and supervised training methods, the training data, final model selection, and implementation of the smoke-cloud segmentation model.

3.1. Model Architecture

Smoke and cloud segmentation is considered as a multi-class pixel classification problem, where there are three possible pixel classes: smoke, cloud, and neither. Two model architectures are considered for this task: a U-Net model (Ronneberger et al., 2015) and a ResUnet model (Diakogiannis et al., 2020). All models are built and trained using the Keras (Chollet et al., 2015) framework in python. The U-Net model is a fully convolutional neural network consisting of a contraction or encoder path, bottle neck, and expansion or decoder path. Unlike an auto-encoder with similar architecture, the U-Net concatenates layers from the contraction path into the expansion path, resulting in the U-shaped network seen in Figure 2. The original U-Net model has four rounds in the contraction phase (encoder), where each round includes two steps of a 2-dimensional convolution with a (3, 3) kernel, followed by a max pooling layer to decrease the input dimension by half. The number of feature channels doubles after each round, with the

[1] `https://github.com/NASARace`

first round having 16 feature channels. After these four rounds, the bottleneck phase again applies two steps of a 2-dimensional convolution with a (3, 3) kernel, resulting in an expanded feature space. After the bottleneck, the resulting layer is upsampled and concatenated with the last layer from the forth round of the contraction phase to begin the expansion phase (decoder). The resulting concatenated layer undergoes the same block consisting of two steps of 2-dimensional convolutions, then the resulting layer is upsampled and concatenated with the last layer of the third round of the contraction phase to start the next round of the expansion phase. This process is repeated three more times to comprise the four rounds of the expansion phase. Here the number of feature channels is halved in each round, resulting in 16 feature channels after the last round. Each layer in the contraction, bottle neck, and expansion phase has a ReLU activation. Once the expansion phase is completed, an output layer is constructed for the classification task. In our smoke-cloud segmentation model, the output layer has three channels - one for smoke, one for cloud, and one for neither - and a softmax activation. Additionally, our model includes batch normalization after each round in the expansion, bottle neck, and contraction, as well as a drop out layer with 30% of the nodes dropped, as seen in Figure 2.

The ResUnet follows the same architecture as the standard U-Net; however, each 2-dimensional convolution block instead is replaced with a residual block defined by the standard ResNet architecture (Diakogiannis et al., 2020; He et al., 2016). While the ResUnet shows improved performance to the standard U-Net (Diakogiannis et al., 2020), the residual block significantly increases the size of the network. In particular, our ResUnet is significantly larger with 32 million parameters compared to the U-Net's 2 million parameters. Because this application requires real-time processing of large images every five minutes, the ResUnet is nearly too large for the task and fails to complete the segmentation in time on a standard laptop. Thus we do not consider any larger architectures, such as Segment Anything Model with 632 million parameters (Kirillov et al., 2023), Vision Transformers with 86-631 million parameters (Dosovitskiy et al., 2021), or Transformer U-Nets with approximately 100 million parameters (Chen et al., 2024).

3.2. Model Training

To utilize the information in unlabeled images, models are pre-trained in a self-supervised manner using the Barlow Twins architecture (Zbontar et al., 2021). Pre-training allows models to better understand the underlying structure of input images and in turn learn features, such as continuity of shapes and different textures, which generalize to learning a specific task. When applied to a U-Net model, Barlow Twins has been shown to drastically improve performance (Punn and Agarwal, 2022). The Barlow Twins method learns features of images by passing two sets of the same data with different augmentations through an encoder, then computing a custom loss that allows the model to learn similarities in images. For our model, the augmentations applied to the data include a random flip, crop, and rotation. Essentially, the same images should be represented similarly in the networks, while different images should have different representations. This is learned through the custom loss function, which uses the summed squares of elements in a correlation matrix of the outputs from the two models where the diagonal is subtracted by one. Hence,

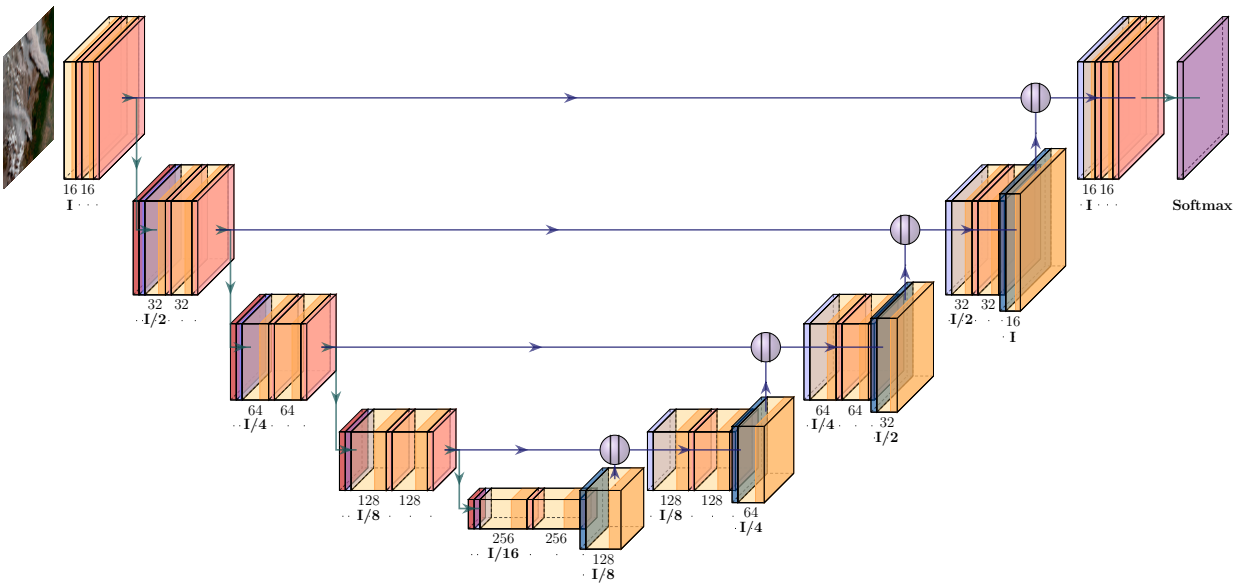


Figure 2: Smoke-cloud segmentation U-Net model architecture, where convolutional blocks are orange, batch normalization is pink, dropout is purple, max pooling is red, and upsampling is navy. Generated using(Iqbal, 2018).

the same images should have a high correlation and different images should have low correlation, resulting in a low loss if the models encode the same images similarly despite augmentations. This process allows the encoder to learn the representations that differentiate two images, even when under augmentation. After pre-training, the encoder can be directly used in a U-Net in the contraction path to improve model performance through transfer learning and fine-tuning. Supervised training follows conventional machine learning where the model is provided labeled training data. Here we consider two main variations of supervised learning after pre-training: transfer learning and fine-tuning. Transfer learning freezes the weights of the pre-trained encoder, whereas fine-tuning allows the model to adjust the weights of the pre-trained encoder during supervised training.

3.3. Training Data

The self-supervised training dataset consists of 10,726 randomly sampled GOES-R images. Images are sampled from 2020-2023 over the days from roughly June 15th-September 15th of each year (day of year 167 to 258). This data range was chosen to cover the wildfire season, hence the pre-training data contains numerous examples of various smoke and cloud configurations. Since the range spans two operational windows for GOES-West, dates after the operational start of GOES-18 use GOES-18 images and days before the start use GOES-17 images. For each day in that range, an hour is selected from 11 a.m. to 5 p.m. eastern time for GOES-16 and from 11 a.m. to 5 p.m. pacific time for GOES-17 and GOES-18. The full scale images are 3000 by 5000 pixels, thus they are tiled into overlapping 512 by 512 tiles before being input to the model. Through this tiling, the original spatial resolution of GOES-R is maintained. Because the pre-training dataset spans the continental U.S., the model is exposed to all configurations potential backdrops to smoke and cloud, such as various vegetation types, land use categories, and natural features.

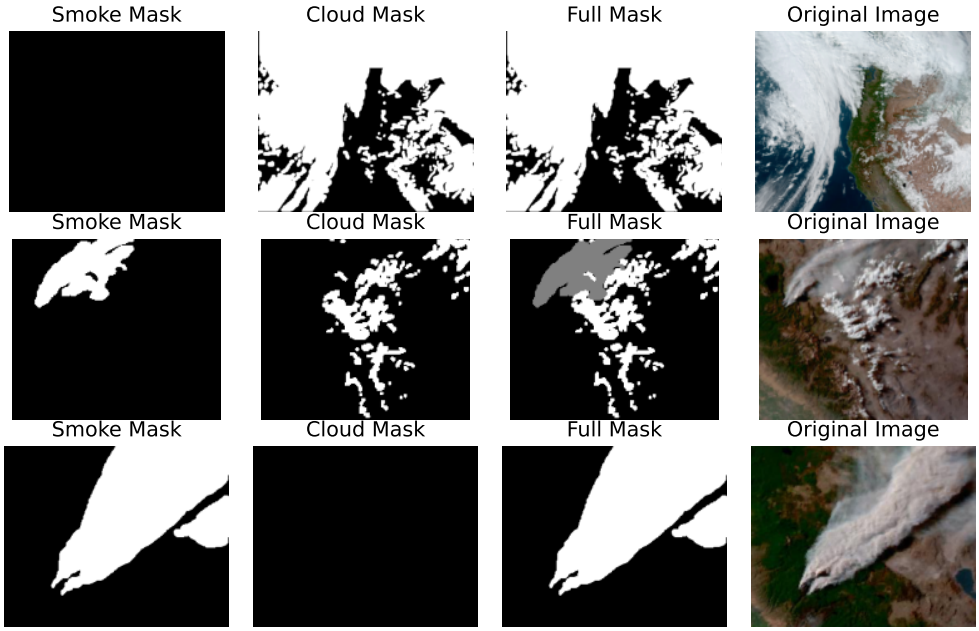


Figure 3: Manually annotated training data examples, where the top row has only cloud, the middle has smoke and cloud, and the bottom has only smoke.

Pre-training thus gives the encoder the opportunity to learn the natural variation of clouds, smoke, and the underlying backgrounds they appear over. Ultimately, this diverse dataset helps the model encode the structure of different features present in smoke and clouds over a variety of geographical regions.

The supervised training dataset is built from 150 manually annotated images, spanning primarily the Western U.S. and Canada over 2020-2023. Fires captured with Terra and Aqua satellites were explored via NASA WorldView application (National Aeronautics and Space Administration) and events with smoke and cloud were screenshot. Additionally, known historical fire dates were used to query the GOES Amazon Web Services S3 bucket (National Oceanic and Atmospheric Administration) to obtain imagery from the GOES series. Archived GOES images are not in true color by default, so the raw bands were first reconstructed into true color images prior to annotation. Images were annotated using the label-studio community labeling application (Tkachenko et al., 2020-2022), with smoke and cloud annotated with different labels. Initial models were built with just smoke annotations; however, the models routine incorrectly classified clouds as smoke when cloud annotations were excluded from the training. Since the segmentation model is interested in class rather than instance, all clouds were considered one instance and all smoke were considered one instance within each image. Examples of manually annotated data are provided in Figure 3. As evident in Figure 3, the training dataset consists of a variety of scenarios, with some images having only smoke, others having only cloud, and some inputs with both smoke and cloud. Images also vary in size and resolution due to the use of Terra, Aqua, and GOES satellites. All images are resized to the same (512,512) input prior to model training, thus some images are down-scaled and some are up-scaled. For model training, the 150 labeled images are randomly

divided into a training and validation set of 125 images and a test set of 25 images.

3.4. Model Selection

A 2x4 experimental design is used to compare two model architectures, ResUnet and U-Net, across four training methods, vanilla, data augmentation, transfer learning, and fine-tuning. The vanilla training involves simply training each model with the given data. The data augmentation variant applies augmentation to the training data, resulting in sixteen variants of each image in the training set through combinations of flips and rotations. Both the transfer learning and fine-tuning variants use a Barlow Twins pre-trained encoder in addition to data augmentation. The encoders are pre-trained with Barlow Twins for a maximum of 25 epochs, with early stopping triggered when the loss increases for two consecutive epochs.

Models are evaluated using 5-fold cross validation on the 125 training and validation data set. For each fold, the 125 images are split into a set of 100 for training and 25 for validation, with performance of the trained model also measured on the 25 image test set. Performance is measured using accuracy, mean IoU, precision, recall, and f-1. Since smoke-cloud segmentation is considered multi-class, a sparse categorical cross entropy loss is used. Each model is trained using the labeled supervised dataset on a batch size of sixteen for a maximum of one hundred epochs with early stopping, where model training is prematurely stopped when the validation loss is the same for ten epochs.

Results from the cross validation experiments are presented in Table 1. While the vanilla and data augmented ResUnets clearly out perform the standard U-Nets, the fine-tuned U-Net has the best performance across the training and validation sets, with comparable performance to the data augmented ResUnet on the test set. Recall, precision, and f-1 are all in similar ranges, indicating the model balances false positives and false negatives well. Performance across the folds on the test set is visible in Figure 4, where the performance gap between the ResUnet and U-Net shrinks from the vanilla variant to the fine-tuned variant. Both architectures experience a performance boost from data augmentation, though the ResUnet has greater improvement. The standard U-Net steadily improves with pre-training in the transfer learning and fine-tuning variants, resulting in similar performance to the ResUnet. Interestingly, pre-training the ResUnet does not provide any significant performance gains, perhaps due to over fitting. Since the ResUnet is a significantly larger model with 32,779,379 parameters compared to the U-Net's 2,164,627 parameters, the pre-trained and fine-tuned U-Net is selected for the final model to optimize both accuracy and inference speed.

Prior to implementing the model in production, the selected fine-tuned U-Net architecture is retrained on the full 125 image training and validation set. The model is trained with a batch size of 16 and learning rate of $1e-5$ for 15 epochs, with learning curves visible in Figure 5. From the curves we can see that the model generalizes to the unseen test set well, with a lower loss and higher accuracy and IoU scores. The training is stopped right before over fitting to the training data occurs, evident by the convergence of the training and test loss. Performance for the final model on the 25 image test set can be seen in Table 2 and the confusion matrix in Figure 6. The model performs best at distinguishing cloud, where 80.2% of pixels are correctly classified, with most confusion occurring between smoke and cloud, where 13% of smoke pixels are misclassified as clouds and 12.9% of cloud pixels misclassified as smoke.

Table 1: Classification metrics for the model and training experiments over training, validation, and test sets. Averages and standard deviations over the five folds are shown ($\mu \pm \sigma$).

		Vanilla		Data Augmentation		Transfer Learning		Fine-Tuning	
	Metric	ResNet	Unet	ResNet	Unet	ResNet	Unet	ResNet	Unet
Train	Accuracy	0.66±0.14	0.59±0.19	0.85±0.02	0.76±0.04	0.82±0.01	0.87±0.01	0.83±0.02	**0.88±0.01**
	Precision	0.57±0.08	0.56±0.14	0.83±0.03	0.74±0.05	0.75±0.03	0.84±0.01	0.8±0.03	**0.86±0.02**
	Recall	0.54±0.05	0.54±0.07	0.77±0.02	0.66±0.05	0.73±0.03	0.82±0.02	0.75±0.04	**0.85±0.02**
	F-1	0.52±0.09	0.49±0.12	0.79±0.02	0.65±0.08	0.73±0.04	0.83±0.01	0.76±0.04	**0.85±0.02**
	IoU	0.39±0.1	0.37±0.11	**0.76±0.03**	0.51±0.07	0.67±0.05	0.72±0.02	0.7±0.04	0.75±0.02
Val	Accuracy	0.66±0.14	0.59±0.19	0.85±0.02	0.75±0.06	0.82±0.01	0.84±0.01	0.83±0.02	**0.86±0.02**
	Precision	0.57±0.08	0.56±0.13	**0.83±0.03**	0.74±0.06	0.75±0.03	0.81±0.02	0.8±0.03	**0.83±0.03**
	Recall	0.54±0.05	0.54±0.07	0.77±0.02	0.65±0.04	0.73±0.03	0.78±0.02	0.75±0.04	**0.82±0.02**
	F-1	0.52±0.09	0.49±0.12	0.79±0.02	0.64±0.09	0.73±0.04	0.79±0.02	0.76±0.04	**0.82±0.02**
	IoU	0.39±0.1	0.37±0.11	**0.76±0.03**	0.51±0.07	0.67±0.05	0.71±0.02	0.7±0.04	0.74±0.02
Test	Accuracy	0.66±0.14	0.55±0.16	**0.85±0.02**	0.71±0.05	0.82±0.01	0.82±0.01	0.83±0.02	0.84±0.01
	Precision	0.57±0.08	0.55±0.12	**0.83±0.03**	0.69±0.04	0.75±0.03	0.75±0.01	0.8±0.03	0.79±0.01
	Recall	0.54±0.05	0.53±0.06	0.77±0.02	0.65±0.04	0.73±0.03	0.75±0.0	0.75±0.04	**0.78±0.02**
	F-1	0.52±0.09	0.46±0.1	**0.79±0.02**	0.61±0.08	0.73±0.04	0.75±0.01	0.76±0.04	0.78±0.01
	IoU	0.39±0.1	0.36±0.11	**0.76±0.03**	0.51±0.07	0.67±0.05	0.69±0.01	0.7±0.04	0.73±0.02

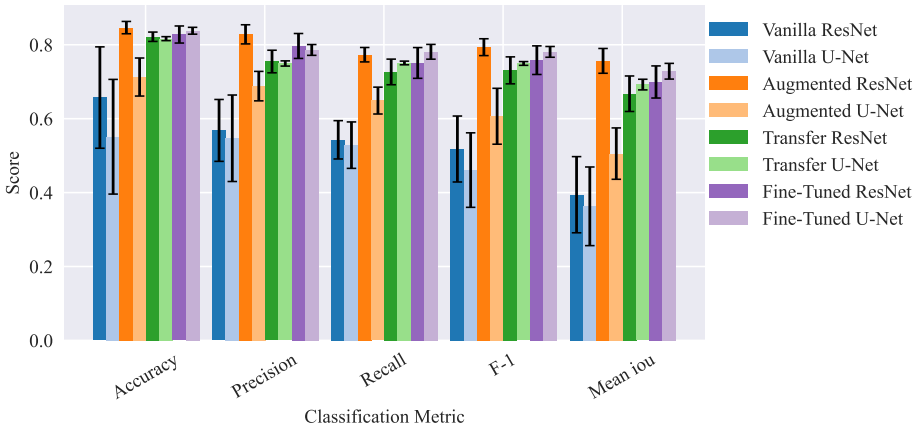


Figure 4: Performance metrics on the test set for model experiments, with the fine-tuned U-Net performing as good as the ResUnet.

Table 2: Performance of the final model on the test set.

Class	precision	recall	f1-score	support
neither	0.93	0.89	0.91	3,650,286
smoke	0.61	0.76	0.68	979,478
cloud	0.84	0.80	0.82	1,923,836
average (macro)	0.79	0.82	0.80	6,553,600

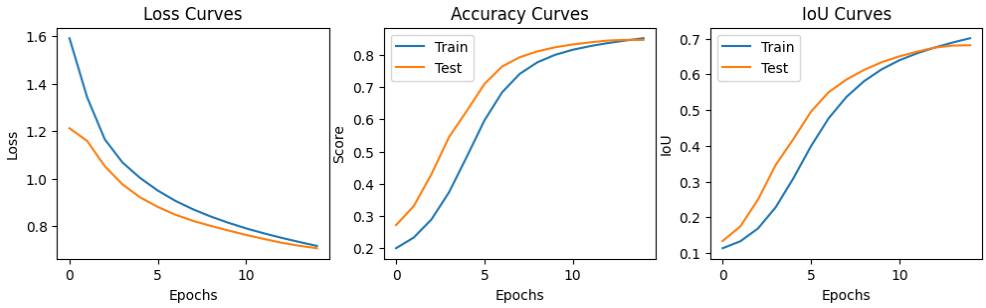


Figure 5: Learning curves during final training for the fine-tuned U-Net model.

Some confusion also occurs between smoke and neither pixels, with 11.5% of smoke pixels misclassified as clouds. This is likely due to the fluid and non-distinct boundary of dispersed smoke. Ultimately the model performs well, with average precision of 0.79, recall of 0.82, f-1 of 0.80, IoU of 0.68, and accuracy of 0.85. When looking at smoke pixels versus any other class the mean IoU increases to 0.695, which is an improvement over the 0.576 score obtained by Larsen et. al. (Larsen et al., 2021) Example outputs from randomly selected images from the test set are visible in Figure 4. The model generally distinguishes smoke and cloud well in the first two examples; however, it struggles more in the third example and misclassifies a cloud as smoke.

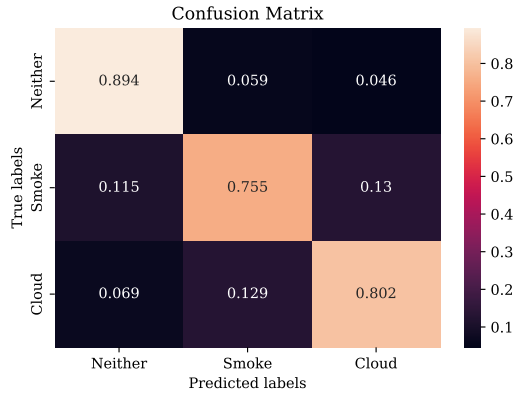


Figure 6: Confusion matrix for the final model on the test set.

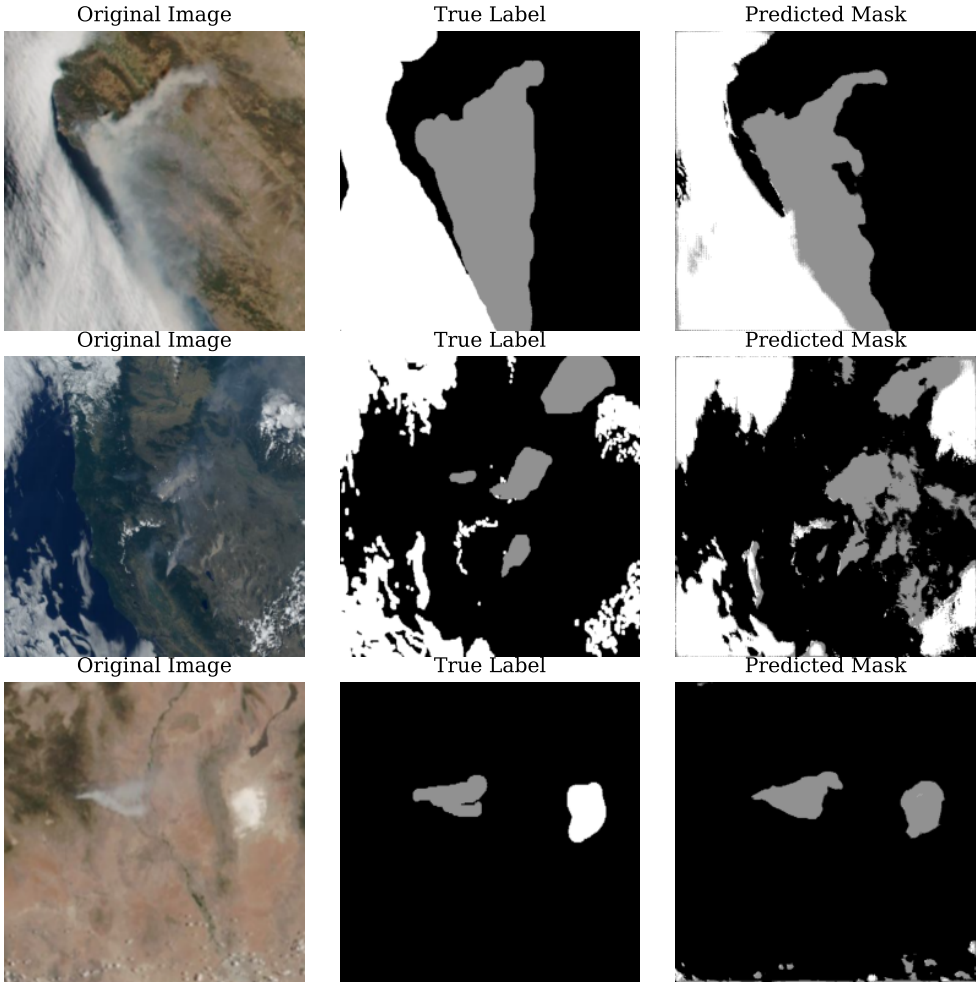


Figure 7: Example image segmentation from randomly selected samples from the test set.

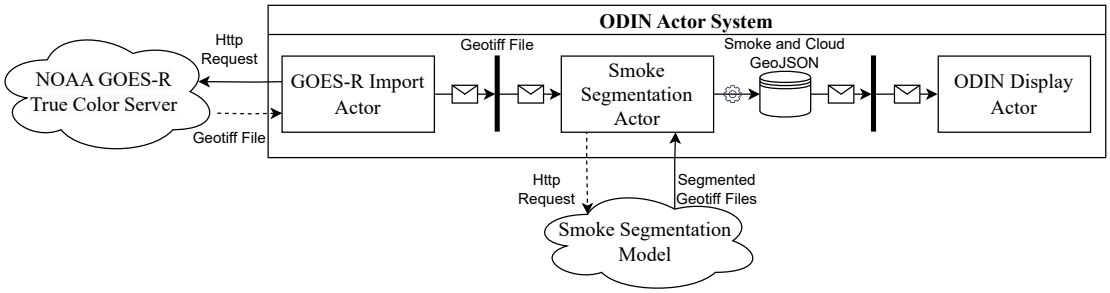


Figure 8: Implementation pipeline for the smoke-segmentation model in real-time ODIN application

3.5. Implementation

Using different configuration settings, the smoke-cloud segmentation model can be used in real-time, processing live data, or in a replay mode, processing recorded data. Figure 8 provides an overview of the ODIN application that implements the real-time use case. The actors within the ODIN application are represented by rectangles. The GOES-R import actor is responsible for importing `geotiff` files from the NOAA GOES-R true color server at `NOAA/NESDIS/STAR` into the ODIN application. It submits http requests to the server and downloads up-to-date `geotiff` files.[2]. The import actor then publishes messages encapsulating information about the downloaded `geotiff` files into a publish-subscribe communication channel (represented by a vertical line in Figure 8) which the smoke segmentation actor is subscribed to. By subscribing to the channel, the smoke segmentation actor receives the messages published to the channel. It then sends http requests including the raw file to a sever which hosts the smoke and cloud segmentation model (referred to as Smoke Segmentation Model in Figure 8). Once the server receives a request, the input image file is processed by the segmentation model and a segmented `geotiff` file is returned back to the smoke segmentation actor, where one layer of the file has smoke pixels and one layer has cloud pixels. To ensure the output files are the same resolution as the input files, the 5000 by 3000 input images are tiled into (512, 512) chunks with overlap and padding added as needed. The returned file is then processed within the smoke segmentation actor and converted into a `geojson` format for display. Resulting smoke and cloud pixels, stored by the smoke segmentation actor, are then published, accessed, and visualized by the ODIN display actor, with options to toggle between just smoke, cloud, or both. The total run time from real-time image download to in the browser segmentation display is approximately 74 seconds, where around 52 seconds are due to the smoke and cloud segmentation pipeline and the remaining 22 seconds are due to the post-processing `geojson` conversion, as seen in Table 3. A similar process occurs for replay demonstrations; however, instead of downloading the most recent file from the web, pre-downloaded true-color GOES images must be already available and stored in a directory specified by the user.

[2] `https://cdn.star.nesdis.noaa.gov/GOES18/ABI/CONUS/GEOCOLOR/GOES18-ABI-CONUS-GEOCOLOR-5000x3000.tif`

Table 3: Execution time in seconds for the real-time smoke and cloud segmentation model in an ODIN application on an Intel Core i7. Averages and standard deviations are calculated from thirty runs.

Source	Execution Time (sec)
Image Import	0.132 ± 0.009
Segmentation	52.342 ± 3.373
Post-Processing	21.813 ± 6.175
Total	74.288 ± 7.576

Table 4: Model performance on the CZU case study example.

Class	precision	recall	f1-score	support
Neither	0.83	0.81	0.82	569,561
Smoke	0.38	0.72	0.50	115,098
Cloud	0.94	0.84	0.88	866,011
average (macro)	0.71	0.79	0.73	1,550,670

4. Case Study: 2020 CZU Fire

On August 16th, 2020, a dry lightning storm ignited a series of wildfires across California that resulted in over 1.5 million acres burned between hundreds of fire events. The CZU lightning complex began as separate fires in San Mateo and Santa Cruz Counties, which combined into one massive fire that burned over 85,000 acres before the fire was fully contained in late September. By August 18th the separate fires had joined into one and high variable winds fueled massive fire growth. Hot spot detectors on the VIIRS instrument could not capture the growth as it unfolded due to the low temporal resolution, hence the fire spread rapidly with little satellite detection. During a critical suppression phase of the fire, excessive smoke was present and resulted in grounded aerial assets for multiple days. To demonstrate the smoke-segmentation model in an ODIN application, we present a case study from the early days of the CZU fire. Figure 9 is a screenshot of an ODIN application with data displayed from 10:16 am on Auguest 19th, 2020. Users can select to view hot spots in the "GOES-R Satellites" sub-menu. GOES-R hot spots are pictured in the colored parallelograms, where the yellow shading indicates a low confidence hot spot. Smoke and cloud pixels generated from the segmentation model can be displayed using the options in the "Smoke Layer" submenu, where smoke pixels are shown in grey and cloud pixels are white. In Figure 9, hot spots are mostly overlapped with smoke pixels rather than cloud pixels. While cloud contamination decreases the confidence in a hot spot, the presence of smoke over a hot spots is a clear indication of an active fire. For example, the hot spot highlighted by the green box is considered low confidence over the past hour due to cloud contamination (NESDIS, 2013), visible in the hot spot history submenu. Thus the low confidence of the hot spots can be augmented by the smoke and cloud segmentation model to provide better situational awareness to decision makers and increase the confidence in active fire locations.

Figure 10 shows the complete smoke and cloud segmentation output over the region spanning California on August 19th, 2020 at 10:16 am. The model successfully segments the large smoke plume traveling over the coast and across California, while also correctly discriminating most of the clouds. There is some misclassification in the southern cloud formations, where less dense clouds are incorrectly classified as smoke. Exact performance metrics for this

Figure 9: Smoke (grey pixels) and cloud (white pixels) segmentation and GOES-R hot spots in an ODIN application are shown with data from the CZU fire on August 19th, 2020 at 10:16 am.

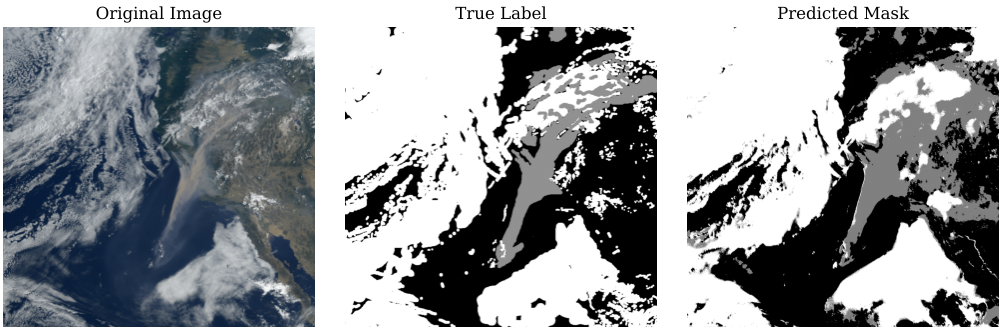


Figure 10: Raw smoke (grey) and cloud (white) segmentation output compared to input image and annotations for the CZU fire case study.

example are available in Table 4, with the model obtaining an overall accuracy of 0.82 and mean IoU of 0.61. The model performs slightly worse on this example than on the test set as a whole, with an f1-score of 0.73 on the CZU example versus 0.80 on the test set. In this example, there are many complex features: overlapping smoke and cloud, dispersed smoke, and a distinct smoke plume. Here the model seems to mostly struggle with mislabeling thinner clouds as smoke and extends the dispersed smoke boundary more than the annotation. Hence the model is more likely to have a false positive for smoke (precision of 0.38), which is preferred to a false negative (recall of 0.72). The model performs similarly on the CZU example to the test set in terms of recall. Despite breaking the input image into (512, 512) sized tiles prior to processing with the model, no performance degradation around the edges is visible due to the use of overlapping tiles. Overall, the predicted mask in Figure 10 maps the smoke and cloud formations of the input image well by capturing the general patterns, even in regions where smoke and cloud overlap.

5. Discussion

With wildfire management personnel ingesting large amounts of data from various sources, using machine learning to distill information, including smoke, in an integrated application improves situational awareness while reducing work load. The U-Net model built for smoke and cloud segmentation successfully differentiates between the two aerosol classes, producing a simplified and focused data product when compared to true color imagery. The immediate impact of this model is real-time smoke location alongside other relevant data. While hot spots were the only other data displayed in the case study, ODIN applications can also import high-resolution wind fields, infrastructure maps, aircraft locations, and vegetation maps. Our model provides easy-to-interpret smoke and cloud locations in the form of pixel classification, which can also be leveraged by additional processing methods if desired. For example, smoke location can be paired with wind fields to understand smoke drift patterns or smoke can be considered when planning flight paths. Ultimately the smoke and cloud segmentation model should be considered alongside other relevant data to maximize impact and usefulness.

Existing smoke segmentation models consider the physics of smoke more than our model by incorporating spectral bands separate from true color compositions (Larsen et al., 2021). However, out model achieves comparable, if not better, performance than these models despite only using true color bands. This implies that our method of self-supervised pre-training could also benefit models with more relevant input bands. While true color imagery does contain information on smoke as evident by its visual distinction, true color spectrum imagery is not useful during nighttime, and thus our model can only be used during day light hours.Although the model was trained on standard true color images, the real-time model implementation utilizes the enhanced GOES GeoColor product (Miller et al., 2020). The GeoColor product blends red (640 nanometers), blue (470 nanometers), near-infrared (860 nanometers), and simulated green into a three-banded image to produce better differentiation between aerosols, such as smoke and various cloud types (Miller et al., 2020). This combination of bands is consistent with previous studies that identified the ideal bands for discerning smoke from clouds and environmental backgrounds (Li et al., 2023). Hence the model may have less accurate performance when using standard true color versus infrared enhanced true color products.

Certain environmental backgrounds that resemble smoke or clouds, such as snowy mountains, may be difficult to distinguish in true color alone. Instead, using the additional infrared band, or other bands useful for background detection (Li et al., 2023), could improve these limitations. The GOES series satellites were chosen due to their high temporal resolution; however, their spatial resolution ranges from 0.5-2.0 km (Miller et al., 2020). Since the pre-training data set consisted only of GOES images, the model learned the underlying distribution at the resolution of GOES. However, the fine-tuning training phase exposed the model to various spatial resolutions through imagery from Terra(0.25-1km), Aqua (0.25-1km), and GOES satellites. The model is only tested for this limited range of resolutions and thus should only be used for images within 0.25-2km resolution.

Despite achieving accuracy and IoU scores higher than existing satellite-based models, there is still some misclassification in smoke entities. The model has only been trained and evaluated on 150 labeled data points and could benefit from further evaluation on more unseen labeled data; however, labeling the data manually is time consuming and prone to human error. Additionally, a limited set of models have been considered at this time and other architectures, such as transformer based U-Nets (Roy et al., 2021) or ensembles of segmentation models, should be explored further for scenarios with increased computational resources. The intended use of the model is to augment existing satellite-based wildfire hot spot detection data products with information on smoke and cloud, which is inherently a high-risk use case. While the model is currently not integrated into hot spot detection and rather is a supplement, integration poses some risks. If outputs from the model are directly integrated into hot spot detection algorithms, such as the GOES-R product, then there is a risk of accumulating error. The main negative impact could be a false negative account of smoke, where the model fails to detect smoke and incorrectly reduces the probability of fire when there is in fact an active fire. While we currently plan to treat the smoke and cloud segmentation as a separate output, we do foresee a potential integration of the model into existing hot spot detection algorithms and these risks must be considered and mitigated if integration occurs.

6. Conclusions and Future Work

In this paper, we present a multi-class smoke and cloud segmentation model with a U-Net architecture, implement the model in a real-time actor-based ODIN application, and demonstrate the model with a case study of the CZU August 2020 fire. We found that the ideal model configuration is a pre-trained and fine-tuned U-Net. Our model performs satisfactory with 68% IoU, 85% accuracy, 82% recall, 79% precision, and 80% f-1 on the test set. When deployed into a real-time application, information from the smoke and cloud segmentation model can complement existing hot spot detection algorithms by providing real-time outputs in just over a minute. In turn, situational awareness is improved by coupling accurate smoke location with other relevant data sources, such as hot spot locations.

Future work will center around applying the combined self-supervised pre-training and supervised fine-tuning method to physics informed models, such as models where inputs include only bands relevant to smoke and clouds. Additional future work will be focused on improving the portability of the smoke and cloud segmentation model, as well as the ODIN application as a whole. Specifically, the model server and application will be migrated to the

Rust programming language, rather than the current Python model server and Scala application. We will also explore options for integrating the model output with the existing GOES-R hot spots to automatically consider smoke when determining hot spot confidence. Other machine learning applications for wildfire data analysis will be pursued, including models for real-time cloud coverage and pyrocumulonimbus detection.

Acknowledgments

This research is supported by the System-Wide Safety (SWS) project in the Airspace Operations & Safety program (AOSP) in the NASA Aeronautics Research & Mission Directorate (ARMD) and partially funded Under Prime Contract No. 80ARC020D0010 with the NASA Ames Research Center. Any opinions or findings of this work are the responsibility of the authors and do not necessarily reflect the views of sponsors or collaborators. The U.S. government retains, and by accepting the article for publication, the publisher acknowledges that the U.S. government retains, a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. government purposes.

Computer Code Availability

All code mentioned in this paper is either already open-source and on github, or available on request. The entire RACE framework is open-source with an Apache 2.0 License and available at `https://github.com/NASARace/race/tree/master`. RACE-ODIN code is also open-source with an Apache 2.0 License and available at `https://github.com/NASARace/race-odin`. Details about the infrastructure, building ODIN applications, and more is documented extensively at `https://nasarace.github.io/race-odin/` and `https://nasarace.github.io/race/`. To run the model in ODIN, users must first start a python server following the instructions in the repository (`https://github.com/NASARace/race/tree/master/race-earth/src/main/python/smoke-segmentation`), then start ODIN using the Scala Build Tool and proper configuration files (e.g., `https://github.com/NASARace/race/blob/master/config/smoke/cesium-smoke.conf`). The model developed in this paper is open-source and available for use via Huggingface at `https://huggingface.co/sequoiaandrade/Smoke-Cloud-Segmentation-RACE-ODIN`. Scripts used to train the model and training datasets are available upon request but have been omitted from the GitHub repository since they are not necessary for operating the software.

References

Abatzoglou, J.T., Williams, A.P., 2016. Impact of anthropogenic climate change on wildfire across western us forests. Proceedings of the National Academy of Sciences 113, 11770–11775.

Akka, 2020. Akka. URL: `http://doc.akka.io/docs/akka/current/scala.html`.

Chen, J., Mei, J., Li, X., Lu, Y., Yu, Q., Wei, Q., Luo, X., Xie, Y., Adeli, E., Wang, Y., et al., 2024. Transunet: Rethinking the u-net architecture design for medical image segmentation through the lens of transformers. Medical Image Analysis , 103280.

Chollet, F., et al., 2015. Keras. `https://keras.io`.

Crowley, M.A., Stockdale, C.A., Johnston, J.M., Wulder, M.A., Liu, T., McCarty, J.L., Rieb, J.T., Cardille, J.A., White, J.C., 2023. Towards a whole-system framework for wildfire monitoring using earth observations. Global change biology 29, 1423–1436.

DelFierro, A., Kilcoyne, H., 2019. Joint Polar Satellite System (JPSS) Ground Segment Data Product Specification (GSegDPS), JPSS GS Systems Engineering, 474-01543, Revision A, October 2019 Electronic Approval. Technical Report.

Dennison, P.E., Brewer, S.C., Arnold, J.D., Moritz, M.A., 2014. Large wildfire trends in the western united states, 1984–2011. Geophysical Research Letters 41, 2928–2933.
DeSalvo, G., Mehlitz, P., 2022. Wildfire ai: Real-time detection powered by ai. URL: https://fsapps.nwcg.gov/nirops/docs/upload/4_3_TFRSAC-ultimately-final.pdf.
Diakogiannis, F.I., Waldner, F., Caccetta, P., Wu, C., 2020. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. ISPRS Journal of Photogrammetry and Remote Sensing 162, 94–114.
Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. URL: https://arxiv.org/abs/2010.11929, arXiv:2010.11929.
Frizzi, S., Bouchouicha, M., Ginoux, J.M., Moreau, E., Sayadi, M., 2021. Convolutional neural network for smoke and fire semantic segmentation. IET Image Processing 15, 634–647.
Hall, J.V., Zhang, R., Schroeder, W., Huang, C., Giglio, L., 2019. Validation of goes-16 abi and msg seviri active fire products. International Journal of Applied Earth Observation and Geoinformation 83, 101928.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
Hewitt, C., Bishop, P., Steiger, R., 1973. A universal modular ACTOR formalism for artificial intelligence, in: Proceedings of the 3rd International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. pp. 235–245.
Iqbal, H., 2018. Harisiqbal88/plotneuralnet v1.0.0 (v1.0.0). URL: https://doi.org/10.5281/zenodo.2526396.
Jeong, M., Park, M., Nam, J., Ko, B.C., 2020. Light-weight student lstm for real-time wildfire smoke detection. Sensors 20, 5508.
Justice, C., Giglio, L., Korontzi, S., Owens, J., Morisette, J., Roy, D., Descloitres, J., Alleaume, S., Petitcolin, F., Kaufman, Y., 2002. The modis fire products. Remote sensing of Environment 83, 244–262.
Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R., 2023. Segment anything. URL: https://arxiv.org/abs/2304.02643, arXiv:2304.02643.
Ko, B., Kwak, J.Y., Nam, J.Y., 2012. Wildfire smoke detection using temporospatial features and random forest classifiers. Optical Engineering 51, 017208–017208.
Larsen, A., Hanigan, I., Reich, B.J., Qin, Y., Cope, M., Morgan, G., Rappold, A.G., 2021. A deep learning approach to identify smoke plumes in satellite imagery in near-real time for health risk communication. Journal of exposure science & environmental epidemiology 31, 170–176.
Li, J., Liu, H., Du, J., Cao, B., Zhang, Y., Yu, W., Zhang, W., Zheng, Z., Wang, Y., Sun, Y., et al., 2023. Detection of smoke from straw burning using sentinel-2 satellite data and an improved yolov5s algorithm. Remote Sensing 15, 2641.
Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.
Mehlitz, P., 2022a. RACE. URL: http://nasarace.github.io/race/.
Mehlitz, P., Shafiei, N., Tkachuk, O., Davies, M., 2016. Race: building airspace simulations faster and better with actors, in: 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), IEEE. pp. 1–9.
Mehlitz, P.C., 2022b. Odin-fire: Open data integration framework for wildland fire management .
Miller, S.D., Lindsey, D.T., Seaman, C.J., Solbrig, J.E., 2020. Geocolor: A blending technique for satellite imagery. Journal of Atmospheric and Oceanic Technology 37, 429 – 448. URL: https://journals.ametsoc.org/view/journals/atot/37/3/JTECH-D-19-0134.1.xml, doi:10.1175/JTECH-D-19-0134.1.
Mo, Y., Wu, Y., Yang, X., Liu, F., Liao, Y., 2022. Review the state-of-the-art technologies of semantic segmentation based on deep learning. Neurocomputing 493, 626–646.
National Aeronautics and Space Administration, . NASA worldview. URL: https://worldview.earthdata.nasa.gov/. accessed: 2022-10-24.
National Oceanic and Atmospheric Administration, . NOAA geostationary operational environmental satellites (goes) 16, 17 & 18. URL: https://registry.opendata.aws/noaa-goes/. accessed: 2023-02-10.
NESDIS, N., 2013. Goes-r advanced baseline imager (abi) algorithm theoretical basis document for fire/hot spot characterization 2.6.
Park, J.H., Lee, S., Yun, S., Kim, H., Kim, W.T., 2019. Dependable fire detection system with multifunctional artificial intelligence framework. Sensors 19, 2025.
Punn, N.S., Agarwal, S., 2022. Bt-unet: A self-supervised learning framework for biomedical image segmentation using barlow twins with u-net models. Machine Learning 111, 4585–4600.
Roestenburg, R., Bakker, R., Williams, R., 2015. Akka in Action. 1st ed., Manning Publications Co., Greenwich, CT, USA.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. CoRR abs/1505.04597. URL: http://arxiv.org/abs/1505.04597, arXiv:1505.04597.
Roy, R., Ahan, M., Soni, V., Chittora, A., 2021. Towards automatic transformer-based cloud classification and segmentation, in: NeurIPS 2021 Workshop on Tackling Climate Change with Machine Learning, p. 60.
Schroeder, W., Oliva, P., Giglio, L., Csiszar, I.A., 2014. The new viirs 375 m active fire detection data product: Algorithm description and initial assessment. Remote Sensing of Environment 143, 85–96.
Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N., 2020-2022. Label Studio: Data labeling software. URL: https://github.com/heartexlabs/label-studio. open source software available from https://github.com/heartexlabs/label-studio.
Xu, W., Wooster, M.J., He, J., Zhang, T., 2021. Improvements in high-temporal resolution active fire detection and frp retrieval over the americas using goes-16 abi with the geostationary fire thermal anomaly (fta) algorithm. Science of Remote Sensing 3, 100016.
Yan, S., Zhang, J., Barnes, N., 2022. Transmission-guided bayesian generative model for smoke segmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3009–3017.
Yuan, F., Dong, Z., Zhang, L., Xia, X., Shi, J., 2022. Cubic-cross convolutional attention and count prior embedding for smoke segmentation. Pattern Recognition 131, 108902.
Yuan, F., Zhang, L., Xia, X., Wan, B., Huang, Q., Li, X., 2019. Deep smoke segmentation. Neurocomputing 357, 248–260.

Yuan, X., Shi, J., Gu, L., 2021. A review of deep learning methods for semantic segmentation of remote sensing imagery. Expert Systems with Applications 169, 114417.
Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S., 2021. Barlow twins: Self-supervised learning via redundancy reduction, in: International Conference on Machine Learning, PMLR. pp. 12310–12320.
Zhou, Z., Shi, Y., Gao, Z., Li, S., 2016. Wildfire smoke detection based on local extremal region segmentation and surveillance. Fire Safety Journal 85, 50–58.

List of Tables

List of Figures