Few-Shot Contrail Segmentation in Remote Sensing Imagery With Loss Function in Hough Space

Sun, Junzi; Roosenbrand, Esther

DOI
10.1109/JSTARS.2025.3525576

Publication date
2025

Document Version
Final published version

Published in
IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

Citation (APA)
Sun, J., & Roosenbrand, E. (2025). Few-Shot Contrail Segmentation in Remote Sensing Imagery With Loss Function in Hough Space. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *18*, 4273-4285. https://doi.org/10.1109/JSTARS.2025.3525576

Few-Shot Contrail Segmentation in Remote Sensing Imagery With Loss Function in Hough Space

Junzi Sun and Esther Roosenbrand

***Abstract*—Condensation trails, or contrails, are line-shaped clouds that are produced by an aircraft engine exhaust. These contrails often impact climate significantly due to their potential warming effect. Identification of contrail formation through satellite images has been an ongoing research challenge. Traditional computer vision techniques struggle with varying imagery conditions, and supervised machine learning approaches often require a large amount of hand-labeled images. This study researches few-shot transfer learning and provides an innovative approach for contrail segmentation with a few labeled images. The methodology leverages backbone segmentation models, which are pretrained on existing image datasets and fine-tuned using an augmented contrail-specific dataset. We also introduce a new loss function, SR loss, which enhances contrail line detection by incorporating Hough transformation in model training. This transformation improves performance over generic image segmentation loss functions. The openly shared few-shot learning library, contrail-seg, has demonstrated that few-shot learning can be effectively applied to contrail segmentation with the new loss function.**

***Index Terms*—Contrail detection, contrail segmentation, few-shot learning, remote sensing, SR loss.**

I. Introduction

AIR transport is critical to global connectivity and the economy, but presents significant environmental challenges. Flights emit greenhouse gases, such as carbon dioxide and nitrogen oxide. Furthermore, condensation trails, or contrails, from aircraft engine exhaust also significantly impact the climate due to their potential global warming contribution [1]. Therefore, effectively monitoring and understanding contrails is essential to managing air transport's climate impacts. Past research studies have focused on detecting contrails and contrail cirrus clouds in satellite remote sensing imagery. However, detecting contrails in satellite images is a challenging task due to the varying conditions of the images, such as lighting, contrast, and perspective.

Traditionally, the detection of contrails is considered a computer vision task, given that it involves identifying linear features. Advancements in computational power have led to early research by [2] and [3], which introduce detection methods in satellite images using various image processing techniques, such as ridge classification, Hough transforms, and contrail line segmentation with linear filters.

In later research by [4], these methods have been expanded with new algorithms to track contrails as they age, drift, and spread. This algorithm detects contrails in an image, computes contrail masks by considering the surrounding pixels, and determines the contrail cluster's overall shape. In another approach, Zhang et al. [5] combined artificial visual inspection and algorithmic contrail identification with meteorological data, leading to a contrail occurrence and persistence index that enables studying contrail coverage.

With image processing algorithms, Minnis et al. [6] investigated linear contrails and contrail cirrus clouds' properties using a blend of remote sensing imagery processing techniques. The study utilized a combination of these approaches, including brightness temperature differences (BTD) [7], infrared bispectral techniques [8], and visible infrared shortwave-infrared split-window techniques [9].

Over the past few years, machine learning, especially supervised machine learning, has been adopted widely in remote sensing research. Thus, research has been applying machine learning for contrail detection tasks. Kulik et al. [10] utilized a convolutional neural network (CNN) model to identify contrails in satellite images. However, despite its success in contrail detection, the model could not exact the location of contrails like later segmentation models. Another similar research by [11] also used a CNN, focusing on whether contrails appear in a frame captured by a ground-based camera.

Later research by [12] has provided a set of human-labeled Landsat images for the research community. Another recent study by [13] aimed to construct an open dataset for contrails primarily over the United States, using GOES-16 satellite imagery. This research employed a convolutional network to take a series of temporally sequenced images as inputs and subsequently detect and outline contrail segments. While the results were promising, the details of the models are not made available in this article. Follow-up research by Meijer et al. [14] uses this dataset to study the contrail formations over the United States.

Several recent research studies aim to better identify contrails from satellite images. A study by Chevallier et al. [15] used the Google contrail dataset for tracking the movement of contrails based on the sequence of imagery inputs. In the work of [16], a CNN similar to U-Net is also developed to locate contrails in GOES-16 satellite imagery. Additional research work has also been developed to estimate their altitude-based Lidar data from other remote sensing products. In a recent paper, Yu et al. [17] introduced a new data source from the SDGSAT-1 satellite, which is the world's first 30-m resolution three-band infrared

Received 25 July 2024; revised 21 September 2024; accepted 28 December 2024. Date of publication 3 January 2025; date of current version 30 January 2025. *(Corresponding author: Junzi Sun.)*

The authors are with the Faculty of Aerospace Engineering, Delft University of Technology Delft, 2628 Delft, The Netherlands (e-mail: j.sun-1@tudelft.nl).

Source code and data are available at: https://github.com/junzis/contrail-seg.
Digital Object Identifier 10.1109/JSTARS.2025.3525576

satellite developed by China. The study also presents a new image-enhancing method to improve the detection of contrails in satellite images. Another study [18] provides a comparison study for the performance of contrail segmentation using different machine learning models.

While recent advancements in machine learning, particularly CNNs, have shown success in detecting contrails, these models often require large, labeled datasets. Even the largest known Google dataset has its limitations when compared to large-scale natural image datasets.

Labeling contrails in satellite images is a time-consuming and resource-intensive process. It poses unique challenges compared to traditional natural image segmentation tasks, which require special knowledge, such as in cases when contrails are hard to distinguish from circus clouds. The objects of interest, contrails, are often elongated and subtle compared to more well-defined objects in natural images like people, animals, or vehicles. Geostationary satellite images often cover large geographical areas, which leads to significant variations in spatial resolution at different latitudes. Contrails can appear faint and stretched due to the perspective of the lens, making them more challenging to detect than objects in typical natural image datasets.

This article tries to address contrail segmentation challenges by introducing a few-shot learning approach for contrail detection and segmentation. Few-shot learning [19] could be a proper approach for this task as it could minimize the reliance on large labeled datasets, enabling the model to generalize from just a few examples. Few-shot segmentation addresses the limitations of conventional deep learning; recent research introduces new concepts like holistic prototype activation networks [20] and divide and conquer strategies [21]. Related studies [22], [23] further enhance segmentation with techniques like base and meta, as well as retain and recover.

In addition to the few-shot learning approach, we introduce a novel loss function, SR loss that specifically targets the linear properties of contrails, improving the model's ability to detect these features in satellite imagery. These research components offer improvements in segmentation by reducing the data requirements and enhancing detection accuracy, specifically for contrails from the remote sensing contrail imagery. We also offer open access to the source code, labeled data, and the neural network model.

In summary, the contribution of this study consists of the following:

1) studying the few-shot learning potential in contrail segmentation;
2) proposing a new loss function for training that takes advantage of the linear shape of the contrails;
3) providing open-source code and data that improves the reproducibility of the research for future applications.

The rest of this article is organized as follows. Section II explains the image data and data augmentation procedures. Section III outlines the architecture of the segmentation neural network model, followed by Section IV with details of the proposed SR loss function. Then, Section V presents the model's experiments and results. Section VI offers the discussion. Finally, Section VII concludes this article.

TABLE I
GOES-16 ABI Channels, Wavelength, and Primary Uses

Ch.	Wavelength	Primary Use
1	0.47 µm	Aerosols
2	0.64 µm	Clouds
3	0.865 µm	Vegetation
4	1.378 µm	Cirrus cloud
5	1.61 µm	Snow, ice, cloud phase
6	2.25 µm	Cloud particle, snow cloud phase
7	3.90 µm	Dog, stratus, fire, hot spot
8	6.29 µm	Upper-level feature (jet stream, waves)
9	6.95 µm	Mid-level feature (water vapor)
10	7.34 µm	Lower-level feature (fronts)
11	8.5 µm	Cloud-top phase
12	9.61 µm	Ozone
13	10.35 µm	Clean infrared longwave
14	11.2 µm	Standard infrared longwave
15	12.3 µm	Dirty infrared window
16	13.3 µm	CO_2 longwave Infrared

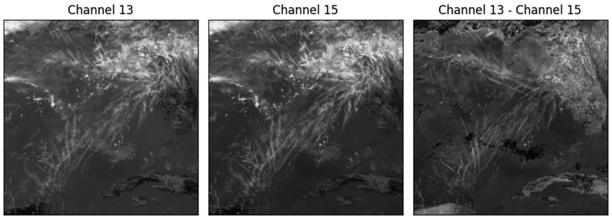

Fig. 1. Example images from channel 13 (12.3 μm) and 15 (10.35 μm) for GOES-16 satellite over the Gulf of Mexico. The BTD of the two channels are shown in the last image.

II. Data Processing

In terms of physical properties, contrails and cirrus clouds share atmospheric similarities in microphysics, allowing for the utilization of infrared channels commonly employed for cirrus cloud identification to detect contrails. This research is conducted with data from the Geostationary Operational Environmental Satellites R Series 16-channel Advanced Baseline Imager (GOES-16 ABI) [24].

The GOES-16 satellite provides high-resolution imagery and atmospheric measurements. The ABI features 16 spectral channels ranging from visible to infrared wavelengths. Table I shows the details of the primary uses of these channels.

A. Data Preprocessing

To isolate the presence of optically thin cirrus clouds and eliminate background and ground interference, we adopt the preprocessing technique involving calculating the BTD.

This technique subtracts one infrared channel from another to generate BTD images. In our case, the difference between 12.3 μm and 10.35 μm for the GOES-16 satellite is obtained. The result is demonstrated in Fig. 1, which showcases a result of a BTD image featuring contrails.

Data from Florida and San Francisco regions over a few days in March and May 2020 is downloaded. These are selected because they are southern coastal regions with more flights. The days are identified with manual visual inspections. The GOES-16 data are obtained using the *goes2go* package. These

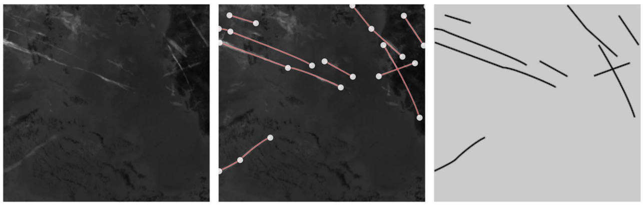

Fig. 2. Process of manual contrail labeling. From left to right: base image, overlay of labeled contrail paths, and final contrail mask image (illustrated with a gray background).

files are processed, where channel 13 (10.35 μm) is subtracted from channel 15 (12.3 μm) to produce the final image used for training and labeling. As a geostationary satellite, GOES-16 data use its projection format to ensure full-Earth disk coverage. We then convert the image into a local projection using the *pyproj* Python library.

The BTD images are processed with the GNU Image Manipulation Program. Contrails are first manually traced as lines based on the authors' visual inspection of the BTD images. The procedure is relatively simple, where thin linear cloud features that the authors can identify are considered contrails.

Based on all identified contrail paths, the mask image is generated with strokes of approximately two pixels on all paths. This masking process is a simplification that improves the labeling speed. In later discussions, we explain why more accurate masking may be trivial.

Fig. 2 shows the process of generating the mask from the original image. Around 40 images from the San Francisco and Florida regions are selected and manually masked with contrails. Within these images, 30% are reserved for evaluating the model's performance, which is not used in training the model.

Furthermore, in this article's experiment section, we also use a public dataset released by Google [13] to compare the performance of the different modeling and training approaches. This dataset is also obtained from the GOES-16 satellite, but it contains thousands of manually labeled satellites, which provides a comprehensive foundation for comparison studies in this article.

B. Image Augmentation

One way of improving contrail detection is to train neural network models with a sufficiently large quantity of remote sensing images. However, large, high-quality datasets are not always available (or necessary), given that manual labeling is time-consuming.

We apply a set of transformations, also known as image augmentations [25], to the image dataset. Image augmentation provides an efficient way to generate training data based on a small labeled dataset. Essentially, this can generate orders of magnitude more input for model training, based on a limited amount of manually labeled data. The augmentation can prevent the model from over fitting due to the following factors.

1) *Locations of contrails*: This results in different locations, orientations, and perspectives of the contrails in the image frame.

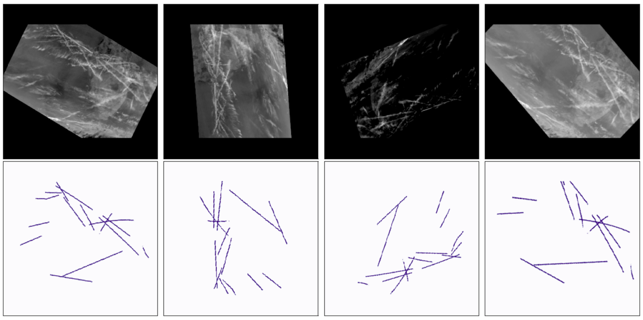

Fig. 3. Image augmentations applied to the same example image.

2) *Lighting variations*: This results in different brightness and contrast.
3) *Viewing angle*: This results in different perspectives of the contrails due to different viewing angles of the optical sensors on the satellites.

During the model's training, we apply a sequence of random augmentations to the BTD image generated from the previous preprocessing step. The augmentation pipeline creates a new image for the training each time an image is loaded. This process aims to train a generalized contrail detection and segmentation model that is robust to varying conditions of the input data. The following transformations are applied in the augmentation pipeline from the training dataset.

1) *Shift, scale, and rotate:* This transformation shifts the image up to 30% of its dimensions, scales it by up to 20%, and rotates it randomly within a 180° range. This set of processes ensures that the contrails can appear in different sections of the input image and with different orientations, providing the model with varying perspectives of contrail positions. The border of the image is filled with a constant value of 0, meaning data will be repeated to fill the entire image.
2) *Padding:* This ensures that images maintain a minimum size of 320 × 320 pixels by padding when needed. This transformation guarantees consistency in image dimensions, even after the random shifts and rotations. A constant value of 0 is also used for filling.
3) *Resize:* After applying the shifts and padding, the image is resized to 320 × 320 pixels to maintain uniformity across the dataset.
4) *Random brightness and contrast:* This randomly adjusts the brightness by up to 20% and the contrast by up to 30% of the image, with a probability of 0.5. It tries to simulate different atmospheric conditions and lighting environments.

The results for the transformation, brightness, and contrast adjustments from a single image can be seen in Fig. 3. The random rotation, scale, and position are applied to each batch (with a probability of 1). The random brightness and contrast adjustments are performed with a probability of 0.5 for each batch in the training.

During the training validation, the process is similar to the training augmentation but without any adjustments to brightness

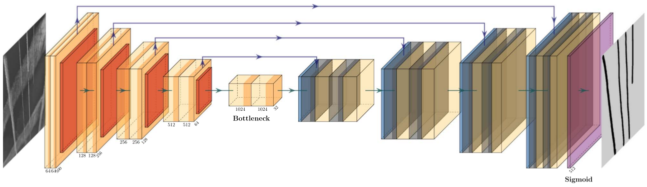

Fig. 4. Illustration of the U-Net neural network.

or contrast. The validation images are shifted, scaled, rotated, padded, and resized to maintain consistency. This ensures that the validation process evaluates the model's performance on geometrically transformed images, without introducing additional variability from lighting conditions. For the testing, only padding and resizing are applied. This ensures that the test data maintains its original form without introducing randomness, allowing the model to be evaluated on images that closely resemble the original dataset while maintaining consistent dimensions.

III. Neural Network-Based Segmentation Model

This section presents the neural network model used for contrail segmentation. The model is based on the U-Net architecture with a residual network (ResNet) backbone. We also introduce the few-shot transfer learning technique and the standard loss function used for image segmentation.

A. ResUNet Model Structure

The U-Net model [26], initially developed for biomedical image segmentation in 2015, has gained significant attention in segmentation research studies due to its high accuracy and efficiency compared to traditional conventional neural networks. It is selected in this article as the underlying model for the contrail segmentation task. The U-Net is a fully CNN and consists of two main components: the contracting and expansive paths, sometimes called encoder and decoder paths.

The encoder or contracting path follows the traditional structure of a CNN, which consists of multiple convolutional and pooling layers. This path is designed to capture the context and features of the input image. The encoder gradually reduces the spatial dimensions of the feature maps while increasing their depth. The encoder captures the abstracted semantics in the input data. The decoder path of a U-Net generates a pixel-wise segmentation map that corresponds to the input image. It upsamples the layers using transposed convolutions (or deconvolutions). This path helps to reconstruct spatial resolutions and, thus, leads to the generation of contrail masks corresponding to the input image. A skip connection is established for each corresponding encoder and decoder layer pair. The skip connection concatenates features from both layers. It allows the network to combine high-level and lower level features for classification tasks, such as the segmentation application in this study.

The overall network is illustrated in Fig. 4, where the input is the satellite image, and the output is the manually created contrail mask.

U-Net is often combined with ResNet [27], a type of network designed to address the challenge of training deep neural networks, especially the diminishing gradients in backpropagation. This study uses the *ResNet50* architecture [28]. Instead of directly learning outputs based on inputs, ResNet learns the difference between the input and desired output.

For image segmentation tasks, it is common to combine the UNet and ResNet as the ResUNet model [29]. The UNet architecture captures the context and features of the input image, while the ResBlock from ResNet improves learning the residual between different neural network layers. Overall, the ResUNet model is designed to capture the context and features of the input image while also maintaining the ability to train deep neural networks effectively.

As suggested in [18], several other model architectures could potentially produce better accuracy. However, as our main focus in this article is few-short learning potential for contrail segmentation, the ResUNet architecture is chosen due to its simplicity and wilder implementation in previous studies [14], [16], [29]. According to the performance comparison by [18], ResUNet is still among the top-performing models for contrail segmentation.

B. Few-Shot Transfer Learning Using Pretrained Models

Few-shot transfer learning allows pertained models to be adapted quickly for specific tasks, and it has become a widely adopted strategy for machine learning. By employing transfer learning, we start with a neural network model with weights already trained with a large, generic public dataset.

The choice of standard UNet architecture allows us to leverage the transfer learning technique, which uses a pretrained ResNet-50 backbone model pretrained on a large existing image dataset, such as ImageNet [30]. Subsequently, the pretrained model undergoes further training with a domain-specific dataset, in our case, a manually labeled contrail dataset. The advantage of this approach lies in its capacity to reduce the time and resources needed for model training.

C. Standard Loss Functions

The choice of loss function is a critical part of training neural network models, as it determines what to optimize during the training. The segmentation of contrail is essential to a binary classification problem. Two conventional loss functions are tested in the neural network training: Focal loss [31] and Dice loss [32]. We also present a new loss function, especially for the contrail detection problem discussed in Section IV.

Here, both focal and Dice loss functions are relevant choices, as the classes (contrail and noncontrail) are highly imbalanced, and these loss functions should not heavily penalize the prediction of the majority class (noncontrail pixels).

Focal loss is designed to focus the learning process on misclassified data (wrong labels) and data that is hard to classify. It applies a specific *modulating* factor to the commonly used cross-entropy loss

$$L_{\text{Focal}}(p_t) = -(1-p_t)^{\gamma}\log(p_t)$$
$$p_t = \begin{cases} p, & \text{if} \quad y=1 \\ 1-p, & \text{otherwise} \end{cases} \tag{1}$$

where $\gamma(\gamma > 0)$ is the scaling factor introduced by focal loss and p represents the probability. $(1-p_t)^{\gamma}$ is the modulating factor. When $\gamma = 1$, the loss is equivalent to binary cross-entropy loss, while increasing γ will give more focus on less well-classified examples. In this article, γ is set to 2.

Dice loss is a loss function that is commonly used in segmentation tasks. It is a metric that measures the dis-similarity of predicted classes and true classes

$$L_{\text{Dice}}(p,g) = 1 - \frac{2gp}{g+p} \tag{2}$$

where p represents the probability of prediction for a pixel belonging to the binary target classes (contrail or noncontrail), g is the ground truth class for that pixel. The Dice loss is also often represented in the logarithmic form, as follows:

$$L_{\text{logDice}}(p,g) = \log\left(\frac{2gp}{g+p}\right). \tag{3}$$

The second term of the Dice loss is the Dice coefficient (or Dice score), which measures the similarity between two data sets

$$C_{\text{Dice}}(p,g) = \frac{2gp}{g+p} \tag{4}$$

where higher coefficient values indicate better similarity between the predicted and ground truth classes. Throughout this article, we use the Dice score based on the validation dataset to evaluate the performance of contrail segmentation tasks.

IV. Designing the SR Loss for Segmentation Model

The previous generic loss functions primarily help the neural network determine whether a pixel should be classified as a contrail, based on differences with adjacent pixel areas. Moreover, they do not explicitly consider the potential shapes of the contrail. In this section, we explain in detail how we design a new loss function, SR loss, to improve the detection of contrails.

In designing our new loss function, we aim to consider this inherent linear shape of contrails. To achieve this, we want to consider the properties of lines in the Hough space using the Hough transformation. The Hough transformation is a technique that converts the standard representation of lines in the Cartesian coordinate system into the polar coordinate format. This transformation allows us to represent lines in the image as points in the Hough space. By comparing the Hough space of the labeled contrail masks and the predicted contrail masks, we can minimize the differences in the linear features of the contrails.

A. Hough Space and Transformation

Hough transformation is a technique proposed by [33] in the 1960s, which has been widely adopted in image processing and computer vision. It is often employed to detect and extract linear features in images. First, the Hough transformation converts the common representation of lines. Let us consider a line in the Cartesian coordinate system

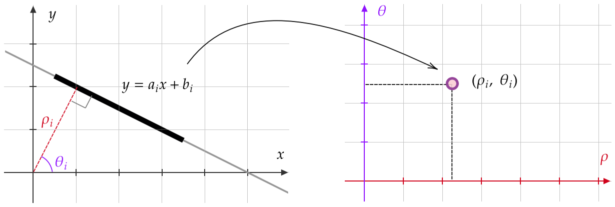

Fig. 5. Example of Hough transform.

$$y = ax + b. \tag{5}$$

The same line can be described in a polar coordinate format

$$\rho = x\cos\theta + y\sin\theta \tag{6}$$

where ρ is the distance between the origin and the closest point on the line and θ is the angle formed by the new line and the horizontal axis.

Essentially, the Hough transform uses a point in the polar coordinate system to represent a line in the Cartesian coordinate system. In this article, we refer to this polar coordinate system as Hough space for convenience. We illustrate the Hough transformation in Fig. 5.

In reality, due to the thickness of the line feature in the image, we can have many points clustered closed in the Hough space, representing a potential contrail mask. The closer the points in Hough space are to each other, the more likely they represent a similar line in angle and location in the image.

To detect multiple contrails in an image, we first discretize the Hough space, where each point, represented by ρ and θ, corresponds to a possible line in an image pixel space. We keep the points in the Hough space with the corresponding line in the image space covering more than 50pixels of contrail masks. It is about 20% of the pixel width of the image. The threshold is determined imperially based on the contrails we observed in the images. It can be reduced to include more short contrails or increased to focus on long contrails.

When training the neural network model, the predicted masks at each iteration are also transformed into the Hough space. The points in the Hough space are then compared with the labeled contrail masks. The difference between the labeled and predicted contrail masks in the Hough space is used to construct the new loss function, SR loss, which is explained in the following section.

B. SR Loss—Combining Dice Loss at Original and Hough Spaces

Fig. 6 shows how we constructed the new loss function, SR loss, to improve the detection of contrails. The detailed process is as follows.

1) *Step 1*: The labeled contrail masks are first transformed into the Hough space using the Hough transformation. The labeled contrail masks are represented as points in the

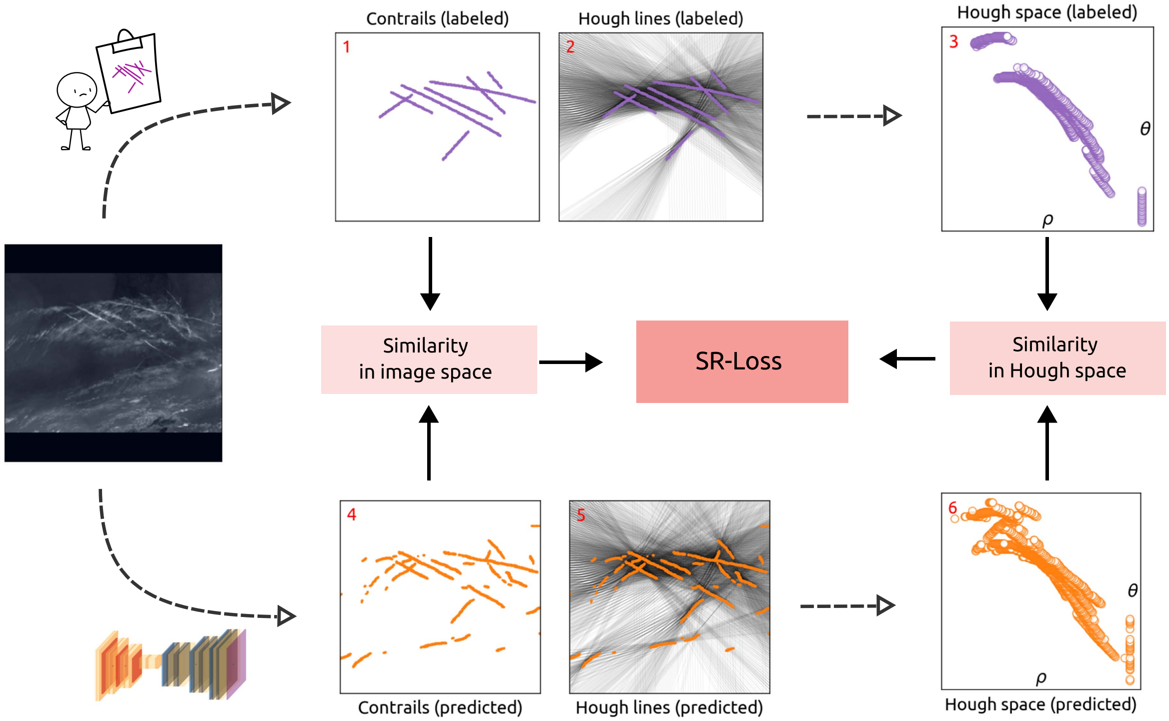

Fig. 6. Design of the SR loss combining similarities in both image space and Hough space. 1) Labeled contrail masks. 2) Lines in the Hough space from labeled contrail masks. 3) Points in the Hough space corresponding to the lines in 2. 4) predicted contrail masks. 5) Lines in the Hough space from predicted contrail masks. 6) Points in the Hough space corresponding to the lines in 5.

Hough space. In subplot 1, the contrail masks are shown. Subplot 2 shows all the lines that contain at least 50pixels of contrail masks. Subplot 3 shows the points associated with these dense lines in the Hough space.

2) *Step 2:* The predicted contrail masks during the training of the neural network model are also transformed into the Hough space. The predicted contrail masks are represented as points in the Hough space. In subplot 4, we can see the predicted contrail masks. Subplot 5 shows all the lines that contain at least 50pixels in the predicted masks. Subplot 6 shows the points associated with these dense lines in the Hough space.
3) *Step 3:* The new loss function, SR loss, is constructed by comparing the Dice loss of labeled and predicted contrail masks in the Hough space.
4) *Step 4:* Finally, we combine the loss in the Hough space with the Dice loss in the pixel space to produce the final SR loss. A weight is introduced to adjust the importance of the two loss functions. In this article, we use the default value of 0.5 for the weight.

Through Fig. 6, we can observe the difference between labels and predictions (subplot 3 versus subplot 6). The new loss function aims to minimize both the difference between subplots 1 and 4, as well as between subplots 3 and 6. The new loss function is defined in (7), as follows:

$$L_{\text{SR}}(p, g) = (1-\alpha)L_{\text{Dice}}(p, g) + \alpha L_{\text{Dice}}(p_h, g_h) \tag{7}$$

where p represents the prediction for a pixel belonging to a contrail and g is the ground truth from the labeled contrail mask. p_h and g_h are predictions and ground truth in the Hough space. α is an adjustable parameter that can be declared to increase or decrease the weight of the loss at two different spaces, which is set to 0.5 for the experiments of this article. This new loss function has two terms. The first incorporates the Dice loss in the pixel space, which minimizes differences between the labeled and predicted contrail masks. However, the linear feature of the contrails is not explicitly considered in the first term, but is addressed in the second term. The second term of the SR loss function deals with the similarity in the Hough space. We minimize the difference between linear features in labeled and training image masks.

V. Experiments and Results

To evaluate the proposed neural network models and the new loss functions, we design the following experiments.

1) Visually examining the performance of the proposed segmentation model.
2) Visually comparing the prediction performance of difference loss functions.
3) Qualitatively analyzing model performances and loss functions using Google open contrail dataset.
4) Qualitatively analyzing the few-shot capabilities using a small subset of training data.

The results from the first two experiments provide illustrative examples of the proposed model. In comparison, the last two experiments are validations on the Google open contrail dataset, which contains around 2000 images for validations.

TABLE II
MODEL AND PARAMETER CONFIGURATIONS

Parameter	Value
Features	single channel, 320×320 pixels, grayscale
Targets	single channel, 320×320 pixels, binary
Backbone	ResNet-50
Model parameters	32.5 million
Model size	around 140 MB (vRAM)
Batch size	16
FLOPs	197.8 billion
Loss functions	Dice loss, Focal loss, and SR loss
Optimizer	Adam optimizer
Learning rate	0.001 (default rate)
Evaluation metric	Dice score

A. *Data, Models, and Parameters*

In these experiments, models are trained with either our data or the Google dataset. To examine the applicability of models on new image sources, we use validation images from the Google contrail dataset [13], which are not used in any training.

Like our own labeled data, the contrail images from the Google dataset are also from GOES-16 satellite but differ in image size, time, and location. The labeling process for the Google dataset is also different. Multiple labels from different labelers are first created. Then, these different versions of labeled contrail masks are combined to create the most common mask for each satellite image. The quality of labels is higher than our labeling method since our contrail masks are the same width in pixels, but in the Google data, they are more accurate and are voted among different labelers.

For the experiment, we use the ResUNet model with the ResNet-50 backbone. The foundational mode is pretrained by the `segmentatio_model.pytorch` libaray [34]. To simplify the model structure for discovering the few-shot capabilities, our ResUNet model only has one input channel and one output channel, which are the input grayscale BTD image and the binary contrail mask. The detailed parameters of the model are in the following Table II.

During the experiment, we used two different GPUs to train the model, including a standard desktop NVIDIA T1000 and a higher performance NVIDIA A10 G for cases requiring more VRAM.

Once trained, the inference time for our models is approximately 0.1 s per image on a CPU (i9-10980XE) or 0.02 s per image on a GPU (NVIDIA T1000).

B. *Evaluating the Model With Example Validation Images*

The first experiment is to validate the model on unseen satellite images used for training and validation. A model is trained with a standard Dice loss function based on the 27 images in our training dataset. It is then used to predict a few examples (with different amounts of contrails) from the Google validation dataset.

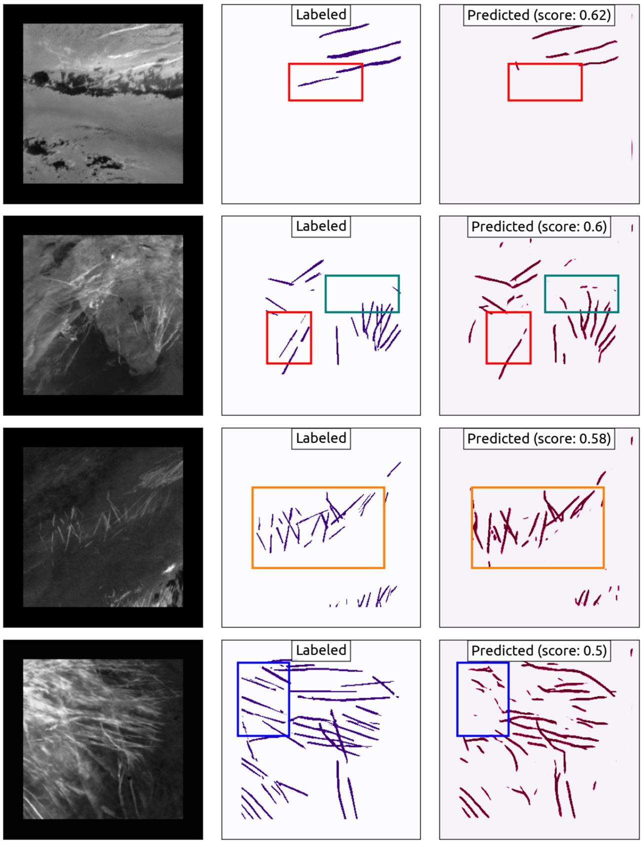

Fig. 7. Application of the contrail detection model with unseen GOES images for Google dataset. The model trained for 1000 epochs with Dice loss. The model is trained on 27 images labeled by the authors with image augmentations.

Fig. 7 shows four examples containing different amounts of contrails and complexities of contrails. The BTD image and labeled contrails are shown in the first two columns, and predicted contrails using our neural network model trained are shown in the last column.

The Dice coefficient (or Dice score), measuring the similarity of the labels and predictions, is calculated for each image, which varies around 0.5 to 0.6 among these four examples. Based on visual inspection of these images, the observations are made as follows.

1) The first image shows a relatively simple case where the contrails are relatively easy to distinguish. We can see that the model can detect most labeled contrails, except the part that overlaps with the dark background (marked with red boxes).
2) The second image shows a more complex situation where contrails are not presented. Our model can identify some (but not all) contrails that are labeled by humans. It also identified some segments of contrails that are missing in the labels. These missing labels and mismatches cause the lower Dice score. The mismatches are marked with red and teal boxes.
3) The third image also shows a more complex situation, where many short contrails are presented densely in two regions in the image. The model can identify contrails with a high level of agreement with human labels from Google contrail data. However, identified contrails are generally thicker than the labels (in orange boxes). Hence, even if

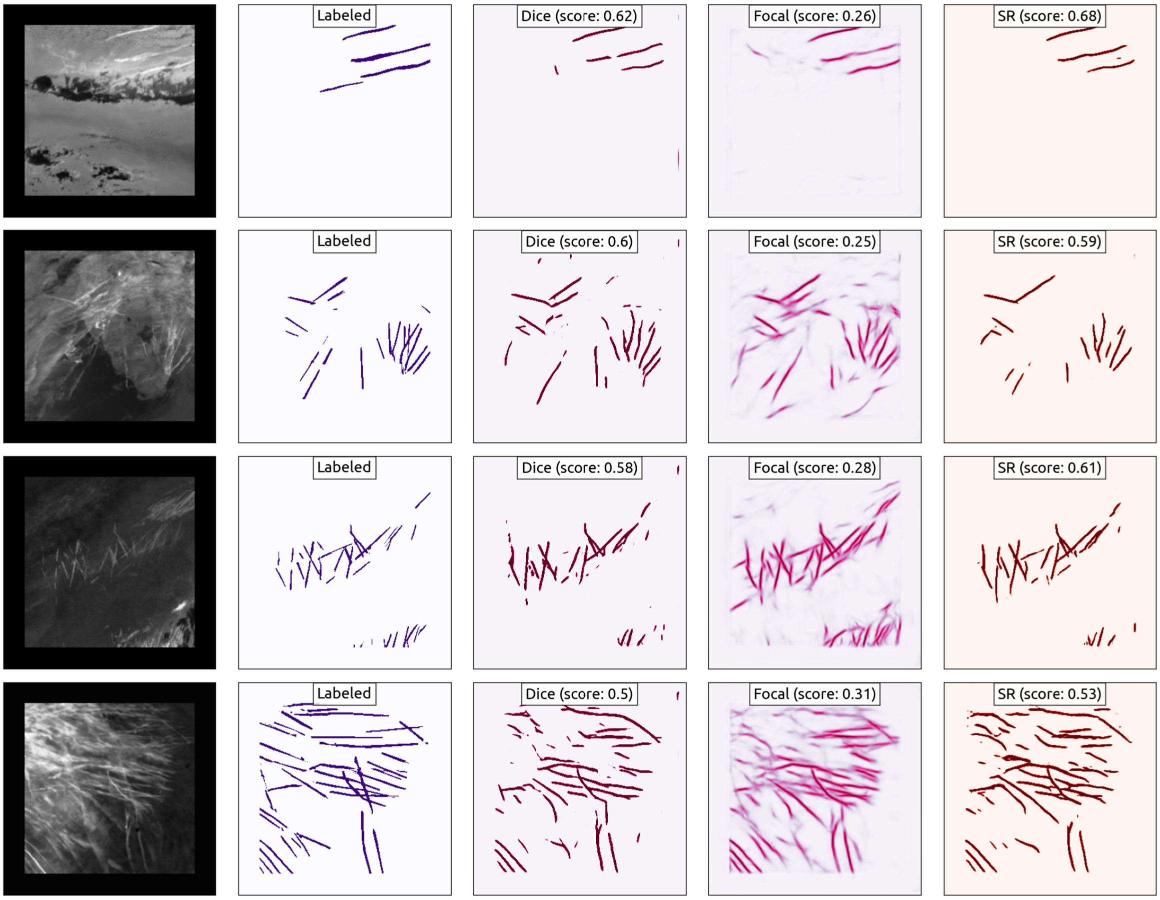

Fig. 8. Difference in output between Dice, focal, and SR losses using samples from the testing dataset. Models are trained on 27 manually labeled images with these loss functions and evaluated on the Google dataset images.

there is a higher level of agreement on the number of contrails, the dice score is lower due to these mismatches.

4) The fourth image shows the most complicated case in this set of examples, where contrails and contrail cirrus clouds exist. The model fails to identify some labeled contrails on the top left region (marked with blue boxes), where the distinction between contrails and contrail cirrus is unclear.

Judging from these images, we can see that the Dice scores can reflect the quality of the models, but the absolute values do not always directly tell the model's performance. A lower Dice score can be caused by predicted contrails missing from the labels, or vice versa, or a difference in the widths between prediction and labels. Across different images, it can be seen that the Dice score can reflect the relative performance among different prediction results. In general, a higher score indicates better agreement with the label. In the following parts, we will use the Dice score to measure the performance differences among different models.

C. Prediction Performance Under Different Loss Functions

This experiment tests three loss functions: Dice loss, focal loss, and the SR loss function. To evaluate the performance and influence of these loss functions, we train three models with the same inputs and assess the model against the same images from previously. Fig. 8 shows the predicted contrails from these models alongside the labels.

We can observe that the detection performance varies for different loss functions across the images. The performances between Dice and SR losses are comparable, where SR loss shows a slightly higher, but not significant, Dice score. The model trained with the focal loss function has a lower score. However, it provides more probabilistic predictions. The score is comparable to the other two if a threshold is used.

Overall, the model trained with SR loss tends to generate fewer short contrail segments. This is due to the threshold used for line detection in the Hough transform, and the model can focus on predicting contrails with longer linear shapes.

D. Qualitative Analysis of the Model and Loss Functions Performance

In this analysis, we evaluate the previously obtained few shot learning models trained with different loss functions on our own labeled images. The validation images are from the Google validation dataset. There are less than 2000 images in the validation dataset, and around 70% of these images contain no contrails, and the analysis focuses on these images with contrails.

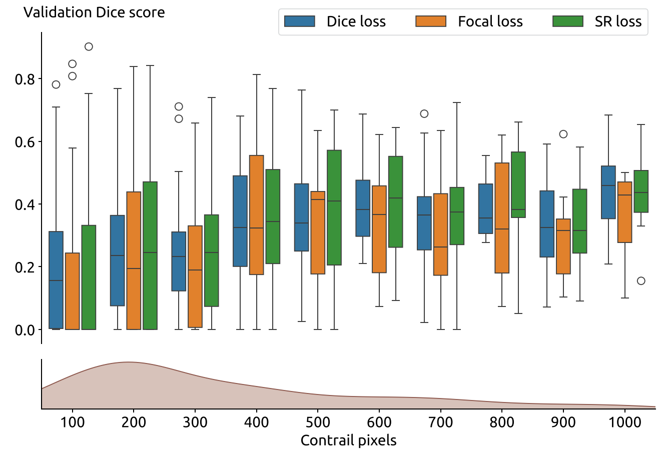

Fig. 9. Validation scores from models trained with Dice, focal, and SR losses. Models are trained on 27 manually labeled images with these loss functions and evaluated on the Google dataset validation images.

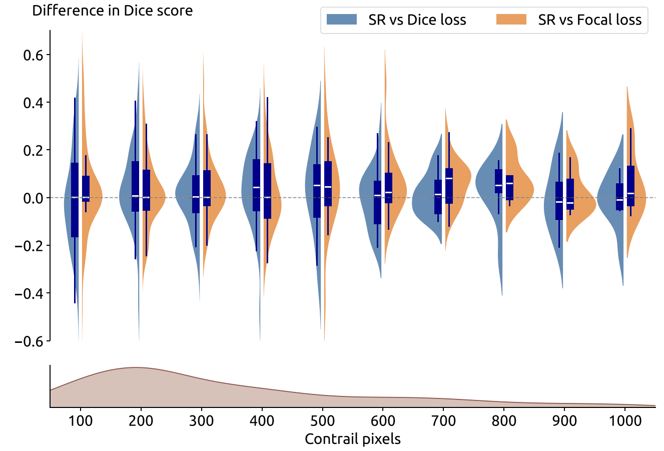

Fig. 10. Difference in performance between models trained with SR loss and Dice or focal losses. Models are trained on 27 manually labeled images with these loss functions and evaluated on the Google dataset validation images.

Fig. 9 shows the performance of Dice scores from three models trained with Dice, focal, and SR losses. In the main section of the figure, the scores are grouped by contrail pixels (in hundreds) based on the labels. In the bottom part of the figure, the distribution of the number of contrail mask pixels from all validation images is shown. For reference, in this validation dataset, the 25th, and 75th percentiles of the contrail pixel numbers are around 200 and 700 pixels, respectively.

This figure shows that the overall accuracy is lower for images with small contrails (smaller than 400 pixels). The Dice score is higher for images with larger contrail masks (larger than 400 pixels), around 0.4 to 0.5. The model trained with SR loss also tends to have a slightly higher Dice score. To better visualize the difference between SR loss and the other two standard loss functions, Fig. 10 shows the performance difference between the model trained with these three loss functions.

In Fig. 10, the distributions are also grouped by the number of pixels. The blue distributions show the differences between models trained with SR loss and Dice loss, and the orange distributions show the differences between models trained with SR loss and focal loss. These differences are also shown using box plots to provide more information on the quantiles of the differences.

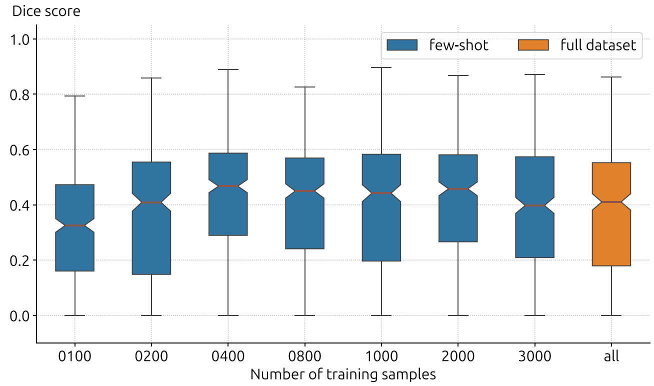

Fig. 11. Contrail segmentation performance from few-shot models trained with the amount of random samples from the Google dataset. All models are trained with SR loss within 30 min on an NVIDIA A10 G GPU and evaluated on the Google dataset validation images.

We can see that the performance of SR loss is similar to Dice and focal for images with short contrails (less than 300 pixels). For images with more (often also longer) contrails, SR loss generally performs better, reflected by the positive median difference in most distribution groups.

E. Evaluating the Few-Short Performances With Different Sample Sizes

In this experiment, we examine the performance of few-shot models against the full model trained on the Google dataset. Unlike previous models trained with our dataset, this test is conducted using only the Google dataset to provide a consistent comparison for models trained with different samples of images.

Six few-shot models are trained with 100, 200, 400, 800, 1000, 2000, and 3000 images, respectively. All models are trained for the same time (30 min) on an NVIDIA A10 G GPU, with the SR loss. The best checkpoint during the training was selected to compare the prediction accuracy.

In all sampled datasets, 30% of randomly selected images contain contrail masks between 0 and 1000 pixels, while the remaining 70% of random samples contain contrail masks larger than 1000 pixels. This choice aims at a more balanced dataset since most of the images in the dataset contain no contrails.

We also train an additional model with the full dataset using the same GPU and training time to compare the performance. The dataset is also balanced, where all images containing contrail masks greater than 200 pixels are considered for training. In addition, 2000 random samples of images with less than 200 pixels of contrail masks are used to balance the dataset. The downsampling on noncontrail images is due to the limited VRAM available on the GPU.

In Fig. 11, the Dices scores of few-shot models are shown in blue, while the scores from the fully trained model are shown

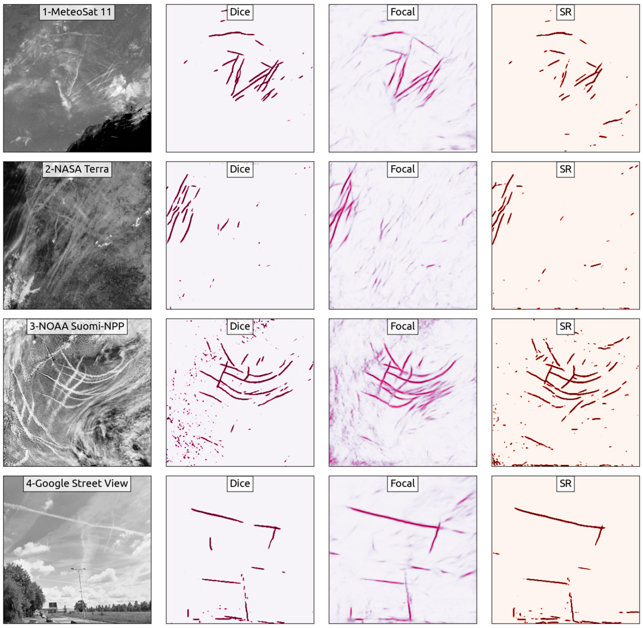

Fig. 12. Contrails detected on different types of image sources. The segmentation model was trained with three losses, including the customized SR loss function.

in orange. Note that the validation is applied to the Google validation images that contain contrail labels.

We can observe that the model trained with 400 random samples performs similarly to the model trained with the entire dataset. Even if the model is trained only with 100 samples, it performs relatively well. Based on these results, one may further hypothesize that a small set of images containing a broad representation of contrail conditions may provide a significantly accurate model without the need for a large number of hand-labeling image data.

F. *Evaluating the Model With Other Image Sources*

Since we apply augmentation techniques in training the neural network model, the resulting contrail detection model has demonstrated an ability to work with an extensive range of images. The contrail detection model can be directly applied to different image sources without additional training.

Fig. 12 presents examples of our model applied to four image sources. The first image originates from MeteoSat, which shares similar image properties with the GOES satellite imagery used during our training phase. The second image is a color photograph from the NASA Terra satellite, where the model has proven capable of managing a broad dynamic range of color inputs.

The third image in Fig. 12 illustrates contrails from ships, as observed by the NOAA Suomi-NPP satellite. Our model still identifies these ship contrails despite their curved shapes, influenced by wind patterns. In the final image, we experiment with an extreme example - a random Google Street View photograph taken near Amsterdam's Schiphol Airport. It can be seen that the models maintain a good level of performance, even when applied to nonsatellite images.

VI. Discussions

A. *Few-Shot Learning and Generalization Capabilities*

The challenge of generating abundant training data is a well-recognized bottleneck in supervised machine learning, especially for image labeling tasks. This study addresses this issue through a two-pronged approach: leveraging few-shot learning and utilizing image augmentation. Unlike traditional end-to-end training that requires large datasets, the few-shot learning technique employed in this study allows for model generalization with a significantly smaller number of samples. This approach begins with a generalized segmentation model pretrained on a large dataset and fine-tunes it specifically for contrail segmentation. Augmenting these few-shot images further amplifies the training set by several orders of magnitude, enhancing the model's robustness and adaptability.

The few-shot learning capabilities demonstrated in this study show that a large dataset may not always be required for effective contrail segmentation. Even with as few as hundreds of training samples, the models showed reasonable performance, with only marginal improvements when the dataset was increased to 400 or more samples. This finding suggests that a carefully selected subset of diverse images may be sufficient for training robust contrail detection models.

However, this also raises important questions about the model's true generalization capabilities. The experiments were limited to satellite imagery with specific properties (i.e., from GOES-16 and similar sources). While we demonstrated that our model works on other image types, such as NASA Terra images or even nonsatellite images like Google Street View, the consistency of these results across a broader spectrum of imagery and atmospheric conditions remains unclear.

To further create models capable of coping with different remote sensing images, we recommend compiling a small dataset of annotated images from multiple satellites to refine the models. This expanded dataset could then be subjected to our proposed image augmentation and few-shot learning strategies, likely resulting in a more robust and generalized contrail segmentation model.

B. *Contrail Masks*

Our labeling process follows a very simplistic approach, in which line segments of the same width are drawn over different contrails. However, the resulting contrail predictions based on visual inspection in Fig. 8 still match well with the carefully hand-labeled contrail masks in the Google dataset.

This observation can be counter-intuitive. We hypothesize that the network can detect the boundaries without being specifically trained. The uncertainty in the boundaries can be better observed in the results from the models trained with focal loss functions due to the probabilistic nature of the predictions during the training under this loss function.

Furthermore, in many aviation applications, very accurate determination of contrail boundaries in the imagery may be optional. The uncertainty in the image can be caused by the resolutions of the geostationary satellites, where the quality of the imageries degrades at the higher latitude region. Another challenge the literature raises is the difficulty of correlating contrails to specific flights, where large uncertainties are expected in relating the segmented contrails to flights.

C. Implications of Loss Function Selection

In this article, we evaluated three loss functions to examine the performance of the model: Dice loss, focal loss, and SR loss. The first two classical loss functions are commonly used in image segmentation tasks. The SR loss is a new customized loss function designed in this study to take advantage of the shapes of contrails. It considers the information in the Hough space and uses the similarity in the Hough space between the target and prediction to improve the model training.

Our experiments with these different loss functions highlighted subtle but considerable tradeoffs. The SR loss performed better in more complex cases, particularly where contrails were longer or more complex, because of its focus on line structures in the Hough space. However, its computational complexity and higher training time may present challenges in practical applications where computational resources or time are limited.

We provide an implementation of SR loss in PyTorch, which allows it to be computed on GPU during training. However, compared to the other two loss functions, SR loss requires additional transformation to the Hough space, which is more computationally expensive and thus can lead to a slower training process.

Part of the calculation can not be fully vectorized in PyTorch, as the matrix representing the points in the Hough space has different sizes. Thus, in the batch operation, we still have to process each image separately, which significantly reduces the training speed. Future efforts should focus on developing a more efficient version of SR loss code to improve computational efficiency.

D. Failure Cases

In the experiments, we can also observe failure cases where the model failed to identify contrails. These cases are often related to images with few contrails or a strong presence of other clouds. We also notice that large contrast in the background features can also cause the model to fail to identify contrails. This may be caused by the normalization of each input image, which is based on brightness and darkest pixels.

Fig. 13 shows three examples where the model failed to identify contrails. In the first example, the model failed to identify the contrail due to the presence of large clouds that obscured the contrail. In the second example, we can see the contrails are quite faint compared to the contrast of other features. In the third example, the model detected many smaller segments of contrails that were not considered contrails in the labeled data.

E. Accuracy of Contrail Detection and Labeling Challenges

This study uses the Dice coefficient (or Dice score) to measure the similarity between the labels and predictions. The results from using the Dice coefficient to evaluate segmentation performance show some inherent limitations in the labeling process, both in our dataset and the Google Contrail dataset.

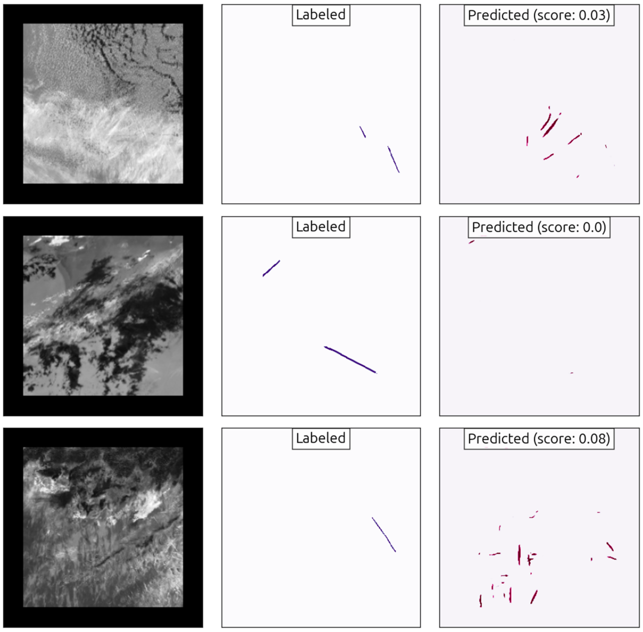

Fig. 13. Examples where our segmentation model failed to identify labeled contrails. The model is trained with SR loss on 400 Google images from the training dataset. These images are from the validation dataset.

These datasets rely on human-labeled contrails, which are prone to inconsistencies. For instance, human labelers often miss shorter or fainter contrails, leading to an incomplete ground truth. As a result, the Dice score might not entirely reflect the model's ability to detect contrails, especially when the model predicts contrails that are missing from the labels.

Given the variability in contrail labels, an important direction for research would be the development of more objective and automated labeling techniques, possibly incorporating semisupervised learning. In addition, using flight data and meteorological data, in conjunction with satellite imagery, could help create more accurate contrail labels and allow for better evaluation metrics that account for both the presence of contrails and their potential environmental impacts. This requires more future research, especially on how to more accurately relate contrails to flights.

F. Limitations of This Study

Several limitations should be acknowledged. First, the size and diversity of the training dataset are constrained to GOES-16 satellites. This can potentially limit the generalizability of the model to other geographical regions or different atmospheric conditions. While we employed various augmentation techniques, the model's performance could still be biased toward the specific image characteristics of the GOES-16 satellite. Expanding the training dataset to include images from other satellite platforms (like Landsat, Sentinel, and MTG satellites) would likely improve the robustness of the model across a wider range of conditions.

The contrail masks in our labeled dataset are relatively simplistic, with uniform widths applied across all contrails. This approximation might limit the model's ability to accurately segment contrails with variable widths, such as those affected by wind shear or different atmospheric conditions. Developing a more sophisticated labeling approach that reflects the actual variability in contrail structures could lead to better contrail segmentation performance.

VII. Conclusion

This study presents the few-shot machine learning approach for detecting flight contrails in remote sensing imagery data. We adopt a U-Net segmentation model and a customized SR loss function. By applying the few-shot models on a larger dataset from Google, we have proved its effectiveness in handling a variety of image quality in terms of location, time, and composition of clouds.

The few-shot contrail segmentation models in this article are trained on a small set of hand-label labeled images in combination with image augmentation techniques. This generates diverse transformations and qualities of satellite images. Despite the limited training data, the model demonstrates strong performance with unseen new image sources, and it would be capable of further improvement with a richer dataset from different satellites.

Moreover, the creation of the SR loss function, specifically designed to leverage the linear nature of contrails in the Hough space, represents another contribution of this study. This new loss function has performed better than traditional Dice loss and focal loss functions, particularly in complex situations involving a mix of many long and short contrail segments.

The software implementation for this model, including the contrail model, training procedures, contrail detection examples, and the actual model weights trained with different parameters, has been open-sourced under the GNU General Public License, allowing future researchers to use or contribute to the model freely.

VIII. Code and data Availability

The model is implemented in PyTorch [35] with the pretrained Segmentation Models [34]. The repository includes the contrail model, training procedures, contrail detection examples, and model weights trained with different parameters.

Author contributions

JS: Conceptualization, Funding Acquisition, Methodology, Software, Writing- Original draft, Data curation, Writing—Original draft, Writing—Review and Editing, Visualization, Data Curation

ER: Conceptualization, Methodology, Writing—Review and Editing, Data Curation

References

[1] V. Grewe et al., "Mitigating the climate impact from aviation: Achievements and results of the DLR wecare project," *Aerospace*, vol. 4, no. 3, 2017, Art. no. 34.

[2] J. M. Weiss, S. A. Christopher, and R. M. Welch, "Automatic contrail detection and segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 5, pp. 1609–1619, Sep. 1998.

[3] H. Mannstein, R. Meyer, and P. Wendling, "Operational detection of contrails from NOAA-AVHRR-data," *Int. J. Remote Sens.*, vol. 20, no. 8, pp. 1641–1660, 1999.

[4] M. Vazquez-Navarro, H. Mannstein, and B. Mayer, "An automatic contrail tracking algorithm," *Atmospheric Meas. Techn.*, vol. 3, no. 4, pp. 1089–1101, 2010.

[5] J. Zhang, J. Shang, and G. Zhang, "Verification for different contrail parameterizations based on integrated satellite observation and ECMWF reanalysis data," *Adv. Meteorol.*, vol. 2017, pp. 1–11, 2017.

[6] P. Minnis et al., "Linear contrail and contrail cirrus properties determined from satellite data," *Geophys. Res. Lett.*, vol. 40, no. 12, pp. 3220–3226, 2013.

[7] S. A. Ackerman, "Global satellite observations of negative brightness temperature differences between 11 and 6.7 μm," *J. Atmospheric Sci.*, vol. 53, no. 19, pp. 2803–2812, 1996.

[8] S. T. Bedka, P. Minnis, D. P. Duda, T. L. Chee, and R. Palikonda, "Properties of linear contrails in the northern hemisphere derived from 2006 aqua modis observations," *Geophys. Res. Lett.*, vol. 40, no. 4, pp. 772–777, 2013.

[9] P. Minnis et al., "Ceres edition-2 cloud property retrievals using TRMM VIRs and terra and aqua modis data—Part I: Algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4374–4400, Nov. 2011.

[10] L. Kulik, "Satellite-based detection of contrails using deep learning," Master's thesis, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, USA, 2019.

[11] N. Siddiqui, "Atmospheric contrail detection with a deep learning algorithm," *Scholarly Horiz.: Univ. Minnesota, Morris Undergraduate J.*, vol. 7, no. 1, p. 5, 2020.

[12] K. J. F. McCloskey et al., "A human-labeled landsat contrails dataset," in *Proc. ICML Workshop Climate Change*, 2021, pp. 1–6.

[13] J. Y.-H. Ng et al., "Opencontrails: Benchmarking contrail detection on goes-16 abi," 2023, *arXiv:2304.02122*.

[14] V. R. Meijer et al., "Contrail coverage over the United States before and during the COVID-19 pandemic," *Environ. Res. Lett.*, vol. 17, no. 3, 2022, Art. no. 034039.

[15] R. Chevallier, M. Shapiro, Z. Engberg, M. Soler, and D. Delahaye, "Linear contrails detection, tracking and matching with aircraft using geostationary satellite and air traffic data," *Aerospace*, vol. 10, no. 7, 2023, Art. no. 578.

[16] V. R. Meijer, "Satellite-based analysis and forecast evaluation of aviation contrails," Ph.D. thesis, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, USA, 2024.

[17] J. Yu et al., "High-resolution thermal infrared contrails images identification and classification method based on sdgsat-1," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 131, 2024, Art. no. 103980.

[18] S. N. Nobel, M. A. Hossain, M. M. Kabir, M. Mridha, S. Alfarhood, and M. Safran, "Segx-net: A novel image segmentation approach for contrail detection using deep learning," *PLoS One*, vol. 19, no. 3, 2024, Art. no. e0298160,.

[19] A. Parnami and M. Lee, "Learning from few examples: A summary of approaches to few-shot learning," 2022, *arXiv:2203.04291*.

[20] G. Cheng, C. Lang, and J. Han, "Holistic prototype activation for few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4650–4666, Apr. 2023.

[21] C. Lang, B. Tu, G. Cheng, and J. Han, "Beyond the prototype: Divide-and-conquer proxies for few-shot segmentation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 1024–1030.

[22] C. Lang, G. Cheng, B. Tu, C. Li, and J. Han, "Base and meta: A new perspective on few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10669–10686, Sep. 2023.

[23] C. Lang, G. Cheng, B. Tu, C. Li, and J. Han, "Retain and recover: Delving into information loss for few-shot segmentation," *IEEE Trans. Image Process.*, vol. 32, pp. 5353–5365, 2023.

[24] T. J. Schmit and M. M. Gunshor, "Abi imagery from the goes-r series," in *The GOES-R Series*. New York, NY, USA: Elsevier, 2020, pp. 23–34.

[25] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020, Art. no. 125.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv.–MICCAI 2015: 18th Int. Conf.*, 2015, pp. 234–241.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[28] A. Shabbir et al., "Satellite and scene image classification based on transfer learning and fine tuning of resnet50," *Math. Problems Eng.*, vol. 2021, no. 1, 2021, Art. no. 5843816.

[29] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 162, pp. 94–114, 2020.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[32] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Berlin, Germany: Springer, 2017, pp. 240–248.

[33] P. V. Hough, "Method and means for recognizing complex patterns," US Patent 3 069 654, Dec. 1962.

[34] P. Iakubovskii, "Segmentation models pytorch," 2019. [Online]. Available:https://github.com/qubvel/segmentation_models.pytorch

[35] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

Junzi Sun received the Ph.D. degree in aircraft performance from TU Delft, in 2019. He is currently a Professor with the Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands.

Esther Roosenbrand is currently working toward the Ph.D. degree in assessment and mitigation for air traffic management with the Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands.