

D1.2 Contrail Tracking Algorithm

Deliverable ID:	D1.2
Project acronym:	E-CONTRAIL
Grant:	101114795
Call:	HORIZON-SESAR-2022-DES-ER-01
Topic:	HORIZON-SESAR-2022-DES-ER-01-WA1-6
Consortium coordinator:	UC3M
Edition date:	14 June 2024
Edition:	01.00
Status:	Official
Classification:	PU

Abstract

This document aims to provide a comprehensive overview of the approach followed for the task of contrail detection and the study of its evolution into Aviation Induced Cloudiness (AIC). It encompasses an in-depth review of the methodology employed, including the use of various sources of data, models and image processing techniques. The primary approach for contrail detection involves training Neural Networks with satellite image data, attempting to learn the underlying patterns in the shape, motion and brightness temperature of these features. The document distinguishes between contrail detection models that yielded successful outcomes and those that encountered challenges, explaining the workaround solutions devised to address these issues and the additional processing needed to achieve the best possible results. Additionally, the document introduces a novel methodology for simulating the formation and evolution of contrails, based in the physics underlying the phenomenon. This model revolves around particle transport in the atmosphere and the temporal evolution of particle size, to simulate the transition of these linear features into AIC. It includes an evaluation of preliminary test cases by comparing the simulated scenarios with real-world observations.

Authoring & approval

Author(s) of the document

Organisation name	Date
UC3M	11/06/2024

Reviewed by

Organisation name	Date
UC3M	11/06/2024
BIRA	06/06/2023

Approved for submission to the SESAR 3 JU by¹

Organisation name	Date
BIRA	13/06/2024
KTH	13/06/2024
RMI	13/06/2024
UC3M	13/06/2024

Rejected by²

Organisation name	Date
-------------------	------

Document history

Edition	Date	Status	Company Author	Justification
00.01		Initial Draft	UC3M	Draft
00.02		Internal Review	UC3M	Sent for Internal Review
01.00		Submission	UC3M	

Copyright statement © (2024) – (UC3M, BIRA, KTH and RMI). All rights reserved. Licensed to SESAR 3 Joint Undertaking under conditions.

¹ Representatives of all the beneficiaries involved in the project

² Representatives of the beneficiaries involved in the project

E-CONTRAIL

ARTIFICIAL NEURAL NETWORKS FOR THE PREDICTION OF CONTRAILS AND AVIATION INDUCED CLOUDINESS

E-CONTRAIL

This document is part of a project that has received funding from the SESAR 3 Joint Undertaking under grant agreement No [Click or tap here to enter Grant No.] under European Union's Horizon Europe research and innovation programme.



We provide now a high-level summary of the project E-CONTRAIL:

Contrails and aviation-induced cloudiness effects on climate change show large uncertainties since they are subject to meteorological, regional, and seasonal variations. Indeed, under some specific circumstances, aircraft can generate anthropogenic cirrus with cooling. Thus, the need for research into contrails and aviation-induced cloudiness and its associated uncertainties to be considered in aviation climate mitigation actions becomes unquestionable.

We will blend cutting-edge AI techniques (deep learning) and climate science with application to the aviation domain, aiming at closing (at least partially) the existing gap in terms of understanding aviation-induced climate impact.

The overall purpose of E-CONTRAIL project is to develop artificial neural networks (leveraging remote sensing detection methods) for the prediction of the climate impact derived from contrails and aviation-induced cloudiness, contributing, thus, to a better understanding of the non-CO2 impact of aviation on global warming and reducing their associated uncertainties as essential steps towards green aviation.

Specifically, the objectives of E-CONTRAIL are:

- ❖ O-1 to develop remote sensing algorithms for the detection of contrails and aviation-induced cloudiness.
- ❖ O-2 to quantify the radiative forcing of ice clouds based on remote sensing and radiative transfer methods.
- ❖ O-3 to use of deep learning architectures to generate AI models capable of predicting the radiative forcing of contrails based on data- archive numerical weather forecasts and historical traffic.
- ❖ O-4 to assess the climate impact and develop a visualization tool in a dashboard.

Executive Summary

The E-CONTRAIL project aims to develop artificial neural networks that utilize remote sensing detection methods to predict the climate impact of contrails and aviation-induced cloudiness. This deliverable is specifically aligned with E-CONTRAIL's Objective 1 (O-1), which focuses on creating remote sensing algorithms for detecting contrails and aviation-induced cloudiness.

Purpose and Goals: The primary goal of this deliverable is to present the methodology for implementing a model that identifies and tracks contrails using high temporal resolution data sources. Additionally, it introduces a physics-based approach for simulating the transition of contrails into cloudiness, explaining how this simulation will help forecast future contrail locations based on real-world observations.

Methodology: The methodology section details the techniques used to accurately identify regions of contrail formation in satellite imagery. It explores various model architectures and configurations, along with preprocessing and post-processing techniques to optimize results by combining expert knowledge with model detections. The models were validated using original GOES-16 data and extrapolated to European satellite data from MSG/SEVIRI experiments. Furthermore, this section examines the fundamental equations governing ice crystal transportation by wind and the thermophysical characteristics of contrails. A novel term addressing gravity's impact on contrail dissipation is introduced. A model simulating the evolution of contrails into non-linear clouds is also presented.

Results and conclusions: This deliverable has successfully developed a model capable of detecting contrails in satellite imagery and another for characterizing the evolution of contrails into aviation-induced cloudiness. A nearly 10% improvement in the Dice score for contrail detection models was achieved. Necessary pre- and post-processing steps for optimal performance were outlined. The models were demonstrated to be applicable to European data, though further quantitative validation with MSG/SEVIRI images is pending. Performance is expected to improve with the availability of MTG data.

The physics-driven contrail simulation model implemented a novel transport equation to address slip mechanisms, simulating the advection-diffusion of ice particles. Persistent contrail propagation is governed by a comprehensive advection-diffusion equation, essential for precise simulations and radiative forcing assessments. While the model requires further refinement, it shows reasonable consistency with methods like CoCiP.

Future enhancements will focus on:

1. **Physics Representation:** Updating the model to incorporate additional physical processes.
2. **Large-Scale Simulation:** Reducing computational time for tracking contrails over extensive areas such as Europe, improving algorithm efficiency.

This deliverable lays the groundwork for future advancements in understanding and mitigating aviation's climate impact through improved detection and simulation techniques.

Table of contents

Abstract	1
Executive Summary	4
1 Introduction	11
1.1 Scope of the Deliverable	11
1.2 State of the art critique	11
1.2.1 Studies on contrail detection	12
1.2.2 Studies on physic-driven contrail simulation	15
1.3 Content of the Document	16
2 Data	17
2.1 Data used for the contrail detection task	17
2.1.1 Training Data	17
2.1.2 Experimental Data	19
2.1.3 Validation Data	19
2.2 Data used for the Physic-Driven contrail simulation task	23
3 Methods	24
3.1 Data-Driven Contrail Detection Models	24
3.1.1 Methodology for Instance Retrieval	26
3.1.2 U-Net Model	32
3.1.3 YOLO Model	39
3.1.4 Mask-RCNN Model	43
3.1.5 Contributions in terms of Optimization of the U-Net models	46
3.1.6 Transfer Learning	47
3.1.7 Model Ensembling	48
3.1.8 Post-processing Techniques	49
3.2 Physics-Driven Contrail Simulation Model	54
3.2.1 Schmidt-Appleman criteria	54
3.2.2 <i>Contrail Cloud Microphysics Model</i>	55
3.2.3 <i>Particle-Transport Model</i> :	56
3.2.4 <i>Numerical Solution to Particle-Transport Model</i> :	58
4 Results	61
4.1 Data-Driven Contrail Detection Models	61
4.1.1 Metrics used for evaluation	61
4.1.2 Model Qualitative Results	62
4.1.3 Model Performance Results	65
4.1.4 Comparison across all trained models	73
4.1.5 Comparison with Previous Results	74
4.1.6 Preliminary experiments with European data	75
4.2 Physics-Driven Contrail Simulation Model	79
4.2.1 non-persistent contrails	79
4.2.2 Persistent contrails	80
4.2.3 Random Traffic:	81

5 *Conclusions*..... 83

6 *References*..... 84

7 *List of acronyms*..... 89

List of figures

Figure 1. Temperature difference between channels 4 (range of wavelengths between 10.3 and 11.3 μm) and 5 (between 11.5 and 12.5 μm) of AVHRR/2 sensor onboard NOAA-12, 4 May 1995, 07:43 UT. [3]	12
Figure 2. Stacked contrail masks of 1996 indicate frequency and predominant bearing of air traffic (derived from 357 AVHRR noon-passages) [3].....	13
Figure 3. The illustration of (a) the ContrailMod architecture and (b) example of 8×8 sub-grid boxes with detected contrails [4].....	13
Figure 4. : IoU scores of segmentation models trained with different backbone architecture proposed in [10]	14
Figure 5. Class distribution within the training set of OpenContrails	17
Figure 6. Class distribution within the validation set of OpenContrails.....	18
Figure 7: The recipe for the Ash RGB composite [15]	18
Figure 8: A false-colour RGB image with its corresponding contrail mask.	19
Figure 9. Example Ash RGB Image from MSG/SEVIRI.	19
Figure 10. Contrail labelling using MSG/SEVIRI data.....	20
Figure 11. Plotting, filtering and advection aircraft trajectories based on ERA5 reanalysis data.....	21
Figure 12. ERA5 Temperature map.	21
Figure 13. ERA5 Relative Humidity map.....	22
Figure 14. ERA5 Wind flow field.....	22
Figure 15. Contrail Bounding Box Detection Labels	24
Figure 16. Contrail Keypoint Detection Segments	25
Figure 17. Contrail Semantic Segmentation Mask	25
Figure 18. Contrail Instance Segmentation Polygons.....	26
Figure 19. Binary ground truth masks in low (left) and high (right) contrail coverage scenario.	27
Figure 20. Examples of the ellipsoids aiming to approximate the proposed methodology.	27
Figure 21. A highly curved advected contrail shape (left) and a slightly curved linear contrail.	28
Figure 22. Transformation of the binary mask to the $\rho - \theta$ space.....	29
Figure 23. Lines detected with the Hough transform.	30
Figure 24. Lines detected with Hough transform after removing similar segments.	31

Figure 25. Subsegments retrieved from the lines detected.....	31
Figure 26. Independent contrail instances retrieved from the original binary mask.	32
Figure 27. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations [21].	34
Figure 28. Comparison among Next-ViT and efficient Networks, in terms of accuracy-latency trade-off on the ImageNet-1K for classification on TensorRT [22].	36
Figure 29. Overall architecture of the NextViT model	37
Figure 30. . Illustration of the conv-attentional technique [23].....	37
Figure 31. Schematic illustration of the parallel group in CoaT [23].....	38
Figure 32. CoaT Model Architecture	38
Figure 33. Overall architecture of the MaxViT model.....	39
Figure 34. Temporal evolution of the distinct YOLO versions.....	39
Figure 35. Overall Architecture of the YOLOv8 model.....	40
Figure 36. A comparison between the C2f Block of YOLOv8 (left) and C3 Block of YOLOv5 (right).	41
Figure 37. Bottleneck Block in YOLOv8	41
Figure 38. Spatial Pyramid Pooling Fusion Block in YOLOv8	41
Figure 39. Detect Block in YOLOv8	42
Figure 40. Mask R-CNN overall architecture	43
Figure 41. Architecture of ResNet-50.....	44
Figure 42. A comparison of the predicted probability masks by the U-Net models when using Dice Lovasz Hinge Loss and the ground truth mask.....	47
Figure 43. Example of randomly selected image samples from ImageNet.....	48
Figure 44. The optical flow field computed between two consecutive MSG/SEVIRI images	50
Figure 45. Histogram of the sizes of contrails detected by a model	51
Figure 46. Example of filtering of large detected features	52
Figure 47. Decomposition of the Ash-RGB product into H,S,V channels.	52
Figure 48. Example mask for filtering out lighter value pixels	52
Figure 49. The result of applying the two filters to the model detections on two different MSG/SEVIRI scenes from the same day.....	53

Figure 50. Computation of the image gradients using Sobel operator	53
Figure 51. . Ground truth contrail labels with 2 keypoints	63
Figure 52. Contrails detected with YOLOv8-pose	63
Figure 53. Masks generated by the different models and compared against the ground truth mask. Here are shown examples in which U-Net models outperform the rest.	63
Figure 54. Masks generated by the different models and compared against the ground truth mask. Here are shown examples in which all of the models have a good performance.	64
Figure 55: Masks generated by the different models and compared against the ground truth mask. Here are shown examples in which Mask-RCNN outperform the rest of the models.....	64
Figure 56: 2-keypoint contrail labels.	65
Figure 57: Precision-Recall Curve (left) and F1-score Curve (right) of YOLOv8-pose with 2 keypoint labels	65
Figure 58: Evolution of the loss functions and the different metrics across epoch both in the training and validation set for YOLOv8-pose with 2 keypoints. Here B stands for Bounding Box results and P for Pose or Keypoint Detection results.	66
Figure 59: 3-keypoint contrail labels	66
Figure 60: Precision-Recall Curve (left) and F1-score Curve (right) of YOLOv8-pose with 3 keypoint labels	67
Figure 61: Evolution of the loss functions and the different metrics across epoch both in the training and validation set for YOLOv8-pose with 3 keypoints. Here B stands for Bounding Box results and P for Pose or Keypoint Detection results.	67
Figure 62: 20-keypoint contrail labels.	67
Figure 63: Precision-Recall Curve (left) and F1-score Curve (right) of YOLOv8-pose with 20 keypoint labels	68
Figure 64. Evolution of the loss functions and the different metrics across epoch both in the training and validation set for YOLOv8-pose with 20 keypoints. Here B stands for Bounding Box results and P for Pose or Keypoint Detection results.	68
Figure 65. Metrics obtained with the CoT-U-net model	69
Figure 66. Metrics obtained with the NextViT U-Net model	70
Figure 67. Metrics obtained with the YOLOv8 model	71
Figure 68. Metrics obtained with the Mask R-CNN model	72
Figure 69. Metrics obtained with the Weighted Ensemble model	73
Figure 70. Comparison between of the Dice Score as a function of the probability threshold across models	73

Figure 71. Best Dice Score for each model for a selected threshold	74
Figure 72. Dice Score of MaxViT U-Net 1 st place solution of the Google Research competition.....	74
Figure 73. Dice Score of the models presented in the 2 nd place solution of the Google Research competition	75
Figure 74. Area of coverage of the 1 st experiment conducted in Europe along with the location of the ground camera contrail observation. The scene is an Ash-RGB MSG/SEVIRI image.	76
Figure 75. MSG/SEVIRI Ash-RGB image on the 25/09/2023 with a red circle covering Toulouse.	76
Figure 76. Contrails detected by the models on the 25/09/2023 in MSG/SEVIRI Ash-RGB images.	77
Figure 77. An example of contrails labelled on an MSG/SEVIRI Ash-RGB image on the 26/01/2024 by visual inspection.....	77
Figure 78. Contrails detected by the models on the 26/01/2024 in MSG Ash-RGB images.....	78
Figure 79. Ensuring temporal consistency throughout an image sequence making optical flow corrections on the MSG Ash-RGB images from the 26/01/2024.....	78
Figure 80: Single non-persistent contrail	79
Figure 81: non-persistent contrails for 4 aircraft after 150 seconds	79
Figure 82: non-persistent contrails from 4 aircraft after 90 seconds	79
Figure 83: persistent contrails for a single flight after about 1 hour (colourbar: kg/m ³).....	80
Figure 84: persistent contrail for a single flight after about 2 hours	81
Figure 85: contrail's energy forcing from CoCiP model.....	81
Figure 86: Contrails generated by a traffic after about 30 minutes flights.....	82
Figure 87: Persistent contrail spots after about 30 minutes (axes are in meters).....	82

List of tables

Table 1: List of acronyms.....	90
--------------------------------	----

1 Introduction

The concern about climate change has escalated in recent years as we are increasingly experiencing noticeable and impactful consequences in our everyday lives. As these effects become more pronounced and widespread, the urgency to address climate change grows stronger. Aviation emerges as one of the primary contributors to climate change, not only due to Carbon Dioxide (CO₂) and Nitrogen Oxides (NO_x) emissions but also because of other factors, such as the generation of condensation trails, commonly referred to as contrails. These are artificial clouds, formed because of aircraft emissions. Even the exact impact of contrails on global warming is still being studied, it is certain that they affect the climate. A contrail forms when an aircraft flies through an area of the atmosphere with very low temperatures and high humidity. The water vapour emitted by the aircraft engines undergoes a process of rapid freezing into tiny ice crystals due to the low temperatures, which are typically around -40 degrees Celsius [1]. These ice crystals serve as nuclei around which more water vapour condenses, forming a visible cloud-like trail behind the aircraft. Contrails frequently scatter and merge as they are carried by wind, gradually developing into a layer of cloud cover. When this contrail formations and subsequent cloudiness persist over extended durations, they entrap the longwave infrared radiation seeking to exit Earth, consequently altering temperatures in the lower troposphere [2].

To mitigate this warming effect, an effective solution involves identifying regions where weather conditions meet the criteria for potential contrail formation, named Ice Supersaturated Regions (ISSR). Upon identification, aircraft can alter their flight paths to avoid traversing these areas.

Several approaches have been investigated in research, mainly using satellite imagery to identify contrails [3], [4], [5] by their characteristic linear form and cooler temperature relative to the surrounding air, making them discernible in particular infrared wavelengths. While some methods have demonstrated effectiveness under specific conditions, they frequently encounter constraints and fail to generalize for application across varied scenarios or with alternative data sources. Consequently, we have examined these limitations and propose a method that partially circumvents some of these challenges, offering a valuable tool and serving as a foundation for future work.

1.1 Scope of the Deliverable

The main goal of this deliverable is to present the methodology considered to implement a model which identifies and tracks contrails using data source with a high temporal resolution. Moreover, it aims to introduce our physics-based approach for simulating the transition of contrails into cloudiness, explaining how this simulation will form the basis for understanding and forecasting future contrail locations derived from the model's detections in real-world observations.

1.2 State of the art critique

The radiative effects of contrails, their properties and dynamics as well as techniques for their detection and the identification of potential formation areas, have been thoroughly investigated in the scientific literature over the past few decades. Contrails were first observed in 1919, when aircrafts reached high enough altitudes to meet the required weather conditions for water vapour to freeze into ice crystals and have since then been an object of study. In this section we will be providing a short overview of the existing work and research gaps bot in the contrail detection modelling as well as in the contrail simulation area.

1.2.1 Studies on contrail detection

When addressing the task of detecting and segmenting contrails in satellite imagery, most approaches rely on data from geostationary satellites. This preference arises because they offer significantly higher temporal resolution compared to low and medium earth orbit satellites. The extended revisit periods of the latter, often surpassing the lifespan of these clouds, make tracking impractical. However, the limited spatial resolution of geostationary satellite images often presents challenges in identifying the contrails within the visible spectrum channels.

Due to the inability to observe contrails in visible channel images, early contrail detection methods relied on passive remote sensing techniques, taking advantage of the lower temperature of ice crystals in specific thermal infrared bands for identification. For instance, owing to their smaller ice crystal sizes [6], contrails often display higher transmissivity within bands centred around wavelengths of 10.3 to 11.3 μm rather than bands around 11.5 to 12.5 μm , resulting in a more prominent appearance in the Brightness Temperature Difference (BTD) image between these two bands (see Figure 1). Building on this understanding, various studies attempted contrail detection using these BTD images.

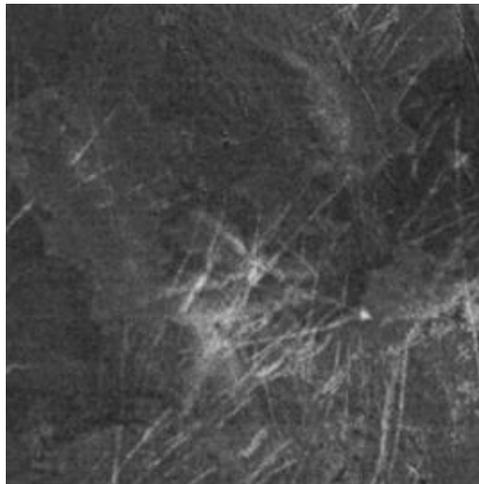


Figure 1. Temperature difference between channels 4 (range of wavelengths between 10.3 and 11.3 μm) and 5 (between 11.5 and 12.5 μm) of AVHRR/2 sensor onboard NOAA-12, 4 May 1995, 07:43 UT. [3]

For instance, Pratt (1991) proposed employing the Hough Transform to detect contrails based on their distinct linear shape [7]. While these and similar approaches performed effectively in certain scenarios, they also yielded numerous false detections and necessitated manual post-processing adjustments on a case-by-case basis. For a long time, the state of the art of the contrail detection model has been the Contrail Detection Algorithm (CDA) proposed by Mannstein et al. (1999) in [3]. Their scheme provided a whole new pipeline that utilized different image processing operations, line detection kernels and brightness temperature thresholds to be able to handle all case scenarios and avoid misdetections providing final contrail masks like the one shown in Figure 2.

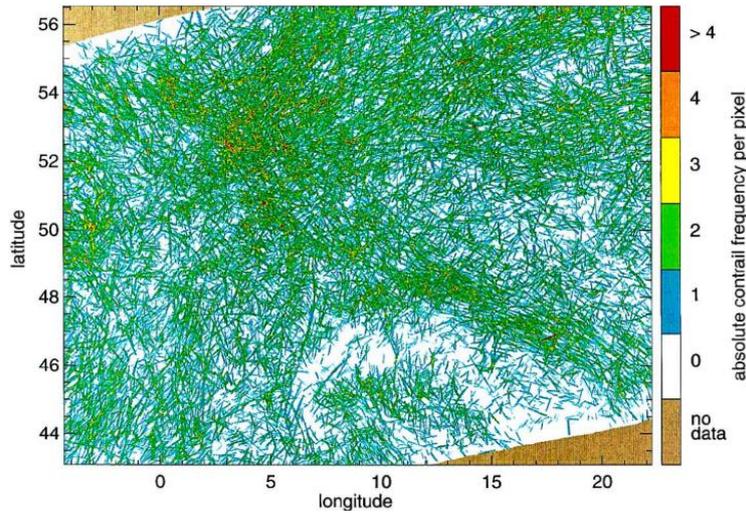


Figure 2. Stacked contrail masks of 1996 indicate frequency and predominant bearing of air traffic (derived from 357 AVHRR noon-passages) [3].

In parallel with these methods, initial efforts to address this issue using Neural Networks have surfaced [8]. However, achieving accurate contrail detection results and developing a broadly applicable algorithm proved challenging due to the absence of a comprehensive dataset encompassing the full range of contrail variability. Over the past few decades, a growing number of studies have emerged focusing on contrail detection methods using Neural Networks. For instance, Zhang et al. (2018) employed a Convolutional Neural Network (CNN), known as *ContrailMod* [4] to identify contrails in a dataset of Himawari-8 BTM images spanning 12.3 to 10.35 μm . The network consists of several convolutional operations followed by three fully connected layers in its architecture (refer to Figure 3) ultimately producing a one-pixel width segment for each contrail. However, unlike most of these authors, they have not made their datasets with the labelled contrails publicly available, so no other models have been developed on this data.

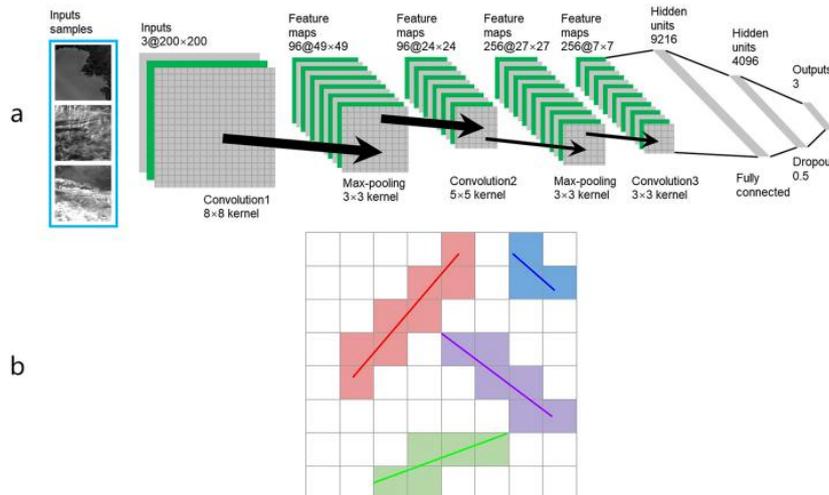


Figure 3. The illustration of (a) the ContrailMod architecture and (b) example of 8×8 sub-grid boxes with detected contrails [4].

The first major publicly accessible dataset was the Landsat Contrails Dataset [9], offering a total of 4,289 spectral images, primarily from 2018, within the viewable range of the GOES-16 satellite. Although Landsat-8's 16-day revisit cycle results in temporal sparsity, its high spatial resolution allows for precise pixel-level contrail labelling. Various studies have published the outcomes of different neural networks trained on this dataset. For example, in [10] several popular segmentation models were benchmarked, including U-Net [11], PSPNet [12], DeepLab V3 [13], and DeepLab V3+ [14], using different combinations of loss functions and encoder backbones.

		UNet	PSP Net	DeepLabV3	DeepLabV3+
ResNet 101	Training	0.6479	0.5032	0.4714	0.7048
	Test	0.3410	0.3788	0.4143	0.4015
ResNext 101-32x4d	Training	0.5411	0.6500	0.7211	0.6657
	Test	0.4224	0.4044	0.4339	0.4266
Xception 71	Training	0.6887	0.7272	0.7730	0.6290
	Test	0.4395	0.4027	0.4246	0.4230

Figure 4. : IoU scores of segmentation models trained with different backbone architecture proposed in [10]

The combination of the U-Net architecture with the Xception 71 backbone achieved the highest Intersection over Union (IoU) of 0.4395. Despite the relatively low IoU, the model produced satisfactory contrail masks. This lower IoU is attributed to the thin and elongated shape of the labels, making the IoU calculation highly sensitive to minor prediction deviations.

The primary issue with this dataset is the insufficient number of labels, which limits the ability to capture enough variability. Additionally, the low temporal resolution of the images hinders the subsequent extension of models for contrail tracking or studying the evolution into AIC.

To deal with the limitations of Landsat-8 Dataset, in 2023 the first comprehensive dataset of labelled geostationary satellite images was released to the public. The OpenContrails dataset contains a significant number of images, totalling 22,410, and includes temporal context for each image, providing both preceding and subsequent snapshots. Subsequently, in May 2023, a contrail detection competition was launched by Google Research [12], inviting experts to develop models trained on this data to achieve optimal performance results based on the Dice Score Metric given by

$$DS = \frac{2 \cdot |X \cap Y|}{|X| + |Y|}$$

where X is the entire set of predicted contrail pixels for all observations in the test data and Y is the ground truth set of all contrail pixels in the test data.

As a result, a multitude of attempts have emerged to train Neural Networks for contrail detection using this dataset, predominantly utilizing the well-known semantic segmentation architecture U-Net combined with various backbones, loss functions, and hyperparameter configurations. The top-performing solution in the competition employed a U-Net network with a MaxViT [13] encoder and was trained using binary cross-entropy loss. However, its superiority in performance was largely attributed to the identification of a 0.5-pixel shift in the labels, enabling appropriate corrections during training.

While only one image in the sequence includes contrail labels, some participants have endeavoured to incorporate the temporal dimension by testing window transformers like Video Swin Transformers [14] or 3D CNNs [15]. However, they reported suboptimal performance due to significant changes between consecutive frames, with a 15-minute difference. Despite this, some suggested leveraging

temporal mixing layers and Long Short-Term Memory (LSTM) [16] modules. Nonetheless, the observed performance improvement was marginal, approximately 0.01%, which did not justify the additional computational time required to process the entire image sequence as input.

Despite the existence of promising solutions, there remains a significant research gap in this domain. This is because numerous architectures beyond U-Net based models warrant exploration, and alternative methods for integrating temporal information into networks could potentially yield satisfactory performance if an innovative approach is identified.

1.2.2 Studies on physics-driven contrail simulation

Contrail simulation models span from detailed small-scale models to comprehensive large-scale climate models, each with varying focuses and complexities. Initially, we review key models and their distinguishing features, with a particular emphasis on the specialized Contrail Cirrus Prediction (CoCiP) model and Ames Contrail Simulation Model (ACSM).

Small-scale models focus on ice nucleation, growth, and sublimation within contrails. For example, Lewellen [16a] utilized a Large Eddy Simulation (LES) model to examine contrail formation and persistence under varying atmospheric conditions. Detailed microphysical schemes, such as those by Kärcher and Lohmann [16b] are often embedded within larger models to enhance their accuracy in simulating contrail microphysics.

Large-scale models, such as global climate models incorporate contrails to evaluate their climate impact, typically using simplified parameterizations. Burkhardt and Kärcher [16c] developed such a parameterization to study the radiative forcing of contrail cirrus on a global scale.

CoCiP model is a specialized large-scale contrail model with its dedicated focus on predicting the formation, evolution, and climate impact of contrail cirrus. It incorporates detailed microphysical processes, integrates real-time aviation data, and operates at high temporal and spatial resolutions. Schumann [16d] demonstrated CoCiP's capability to predict global contrail distributions and their radiative impacts.

ACSM (Jinhua Li, 2023) is a more recent specialized large-scale contrail model which shares many similarities with CoCiP w.r.t. contrail formation, evolution and climate impact of contrail cirrus. Nevertheless, there are some distinctive features as outlined in (Jinhua Li, 2023), such as calculating contrails by aircraft occupancy in a gridded airspace rather than for individual aircraft trajectories, as well as the approach to model the cross-sectional area of contrails.

Here, we have developed a novel specialized large-scale model to simulate the temporal evolution of contrails, leveraging a new transport equation. This model integrates foundational elements from established frameworks such as CoCiP and ACSM particularly in addressing the microphysics of contrail formation and growth. The key innovation lies in the use of a new advection-diffusion equation, which enhances the accuracy of simulating contrail evolution. Additionally, the model incorporates a neural network scheme to reconstruct wind components and other atmospheric variables from ReAnalysis data, providing a more precise representation of atmospheric conditions. This enhanced approach aims to improve the fidelity of contrail simulations, contributing to a better understanding of their impact on radiative forcing.

1.3 Content of the Document

The document is organized as follows: Section 2 offers detailed descriptions of the data sources used for both contrail detection and the simulation of contrail dynamics. Section 3 delves into the specifics of these approaches, while Section 4 discusses the results obtained. Finally, Section 5 addresses the issues encountered, presents conclusions from our findings, and outlines future steps for refining these models as we transition to the next phases of the project. Section 6 provides a list of acronyms, and finally, Section 7 contains the references.

2 Data

In this section, we provide a concise overview of the data sources that proved instrumental in the development of our contrail detection and simulation models, as well as those identified as valuable for subsequent validation or ground-truth label generation purposes.

2.1 Data used for the contrail detection task

In this section, we provide an overview of the data used for training the models and conducting experiments to test their performance on alternative image sources. Additionally, we review the data used for labelling contrails to generate ground truth, which allows us to compare the model predictions and assess their performance.

2.1.1 Training Data

The dataset used for training our models is the OpenContrails dataset [5] which incorporates data from the Geostationary Operational Environmental Satellites - 16 Series (GOES-16) as previously mentioned. We consider this dataset to be of great importance due to its substantial number of labelled scenes, as training neural networks necessitates sufficient data to encompass the variability of the objects we aim to detect. In Figure 5 and Figure 6 is shown the number of images within each set as well as the number of images containing contrails in each of them.

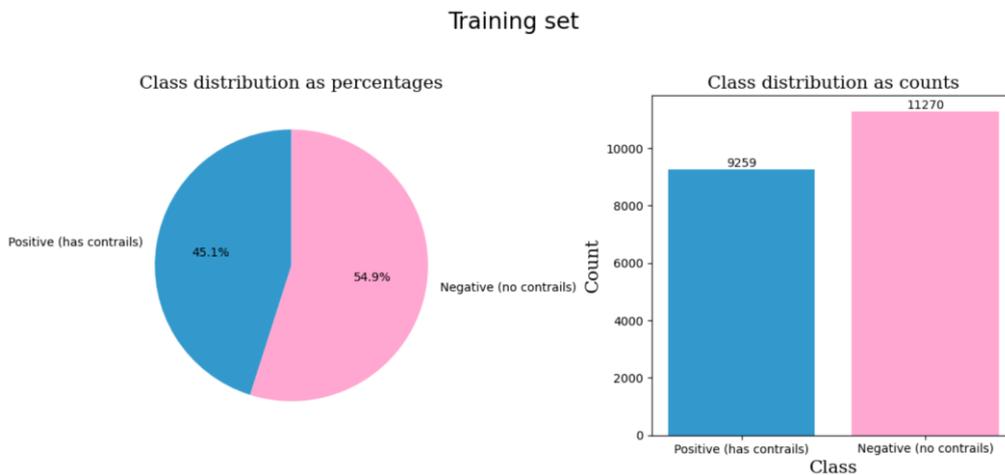


Figure 5. Class distribution within the training set of OpenContrails

Validation set

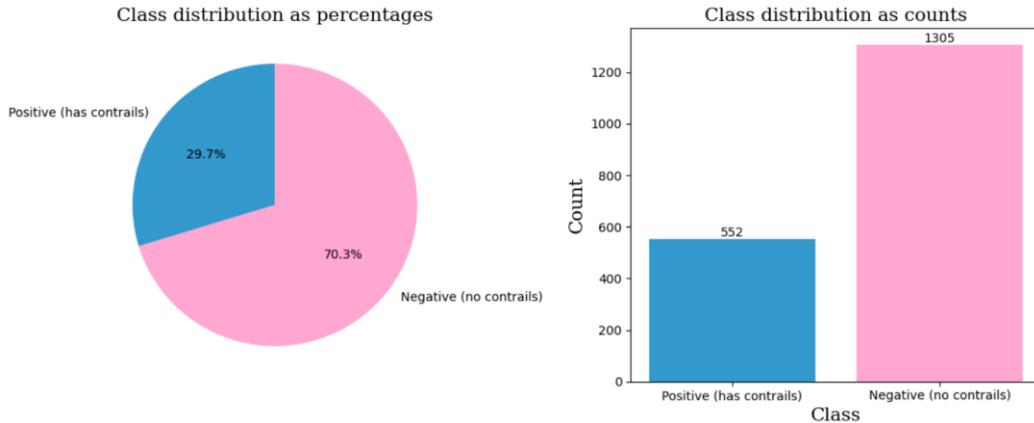


Figure 6. Class distribution within the validation set of OpenContrails

Given the intricate nature of scenes, coupled with disparities in images arising from atmospheric, lighting, and weather conditions, as well as low resolution, having a vast quantity of data becomes imperative. This abundance of data enables the network to effectively capture the underlying patterns associated with contrail shapes, brightness levels, and dynamics.

The images we have decided to use are the false-colour RGB image product, known as Ash RGB, composed of the BT channels described in Figure 7.

Ash RGB Recipe

Color	Band / Band Diff. (µm)	Min – Max, Gamma	Physically Relates to...	Small contribution to pixel indicates...	Large contribution to pixel indicates...
Red	12.3 – 10.3	-6.7 to 2.6 K 1	Optical depth / cloud thickness	Thin clouds	Thick clouds, ash plume
Green	11.2 – 8.4	-6.0 to 6.3 K 1	Particle phase/size	Large water or ice particles	Small water or ice particles, sulfur dioxide gas
Blue	10.3	243.6 - 302.4 K 1	Temperature of surface	Cold surface	Warm surface

Figure 7: The recipe for the Ash RGB composite [15]

The models are trained to detect contrails on just a single frame since this dataset does not provide sequence-labels. The labels are binary masks like the one shown in Figure 8.

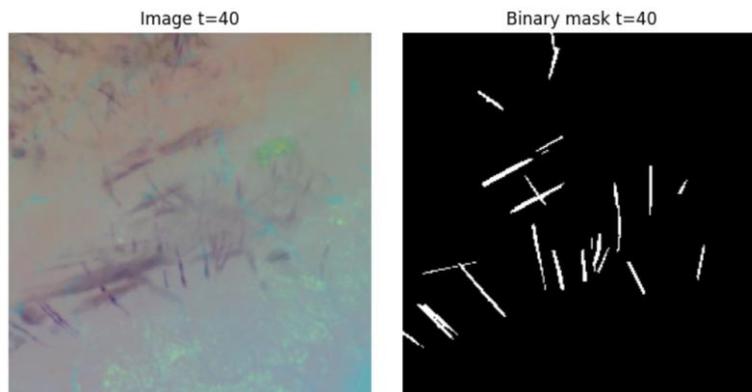


Figure 8: A false-colour RGB image with its corresponding contrail mask.

2.1.2 Experimental Data

We conducted preliminary experiments using data from the Spinning Enhanced Visible and InfraRed Imager (SEVIRI) aboard the Meteosat Second Generation (MSG) satellite. This data has a resolution of 3 km x 3 km in the infrared, which is significantly lower than the resolution of the data used for training, so we do not expect the results to be as accurate. Additionally, these images have a lower temporal resolution, providing a snapshot every 15 minutes. Here, as input for the detections (see Figure 9), we also use the false-colour Ash RGB product, which is presented briefly earlier.

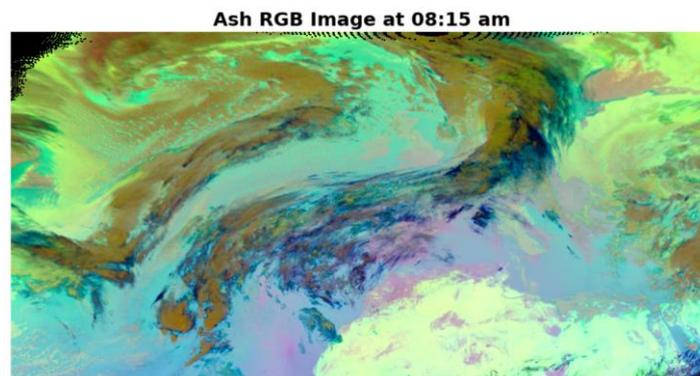


Figure 9. Example Ash RGB Image from MSG/SEVIRI.

2.1.3 Validation Data

To validate the detection performance on the experimental data, we need to generate ground-truth contrail labels by manually creating polygons that contain each of the contrails of an image as shown in Figure 10.

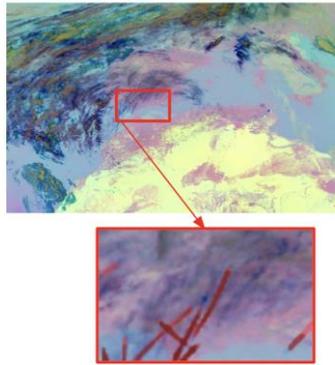


Figure 10. Contrail labelling using MSG/SEVIRI data

We have decided to follow the methodology used by the team at Google Research, who created the OpenContrails Dataset. The steps are as follows:

1. Plotting Trajectories: We begin by plotting all aircraft trajectories obtained from ADS-B data over a series of selected previous time steps onto the image we aim to label. This allows us to visualize the paths taken by aircraft in the region of interest.
2. Filtering Trajectories: Next, we filter out trajectories that do not pass through areas where contrails are likely to form. Specifically, we apply a threshold where the temperature must be -40 degrees Celsius or lower and the relative humidity must be 90% or higher. These conditions are conducive to contrail formation. We set these conservative thresholds to ensure that no potential contrails are mistakenly discarded.
3. Simulating Advection: We then simulate the movement of the contrails generated by these aircraft using the wind velocity field. For this simulation, we employ the second-order Runge-Kutta method [16]. This step accounts for the wind-driven advection of the contrails, providing a more accurate representation of their actual positions over time.

This process ensures an accurate matching of the observed contrails in the images with those potentially formed by each of the aircrafts that have flown that region. The process of generating these labels is summarized in Figure 11.

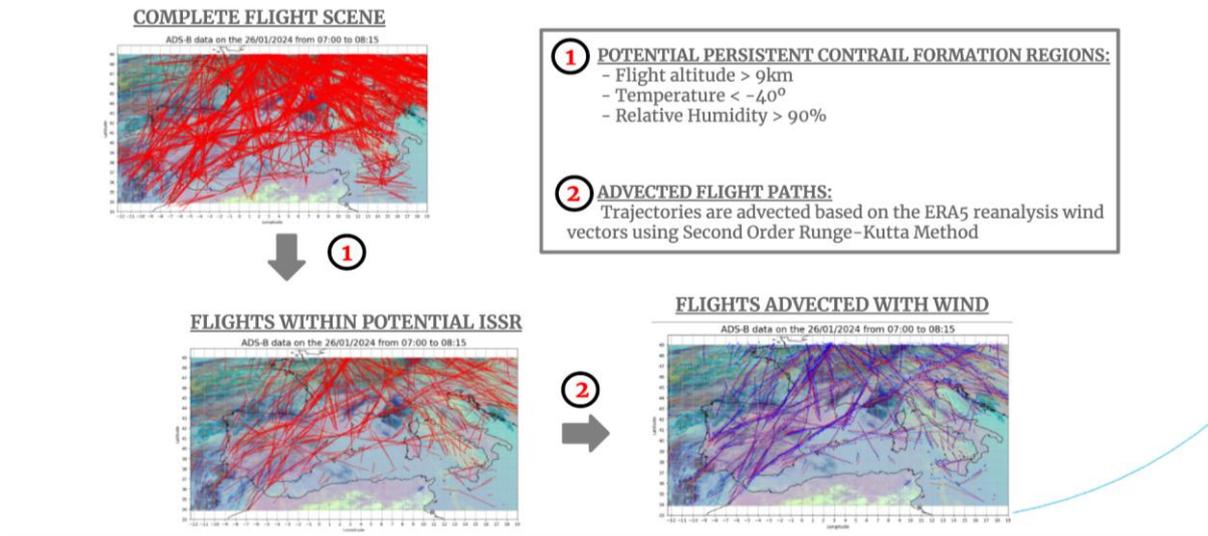


Figure 11. Plotting, filtering and advection aircraft trajectories based on ERA5 reanalysis data.

To follow this steps, ERA5 data is used for filtering and advection of the flight paths. ERA5 uses a sophisticated data assimilation system to integrate observational data from various sources (satellites, weather stations, aircraft, buoys) with a numerical weather prediction model. It offers a temporal resolution of 1 hour and a spatial resolution of approximately 30 km grid, providing detailed and accurate representations of atmospheric variables over time and space.

Here, we are focusing on:

- ❖ Temperature: It represents the thermal state of the atmosphere at different levels. A visualization example for a given pressure level is shown in Figure 12.

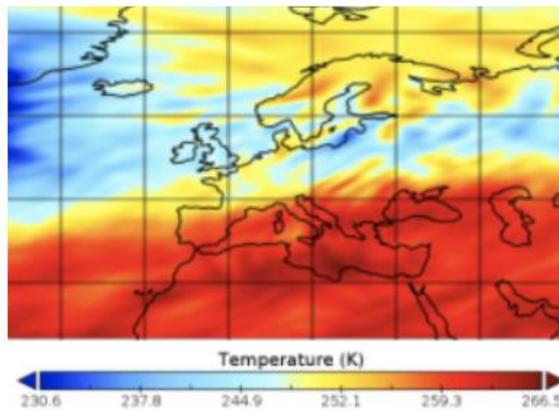


Figure 12. ERA5 Temperature map.

- ❖ Relative Humidity: It is the ratio of the current amount of water vapour in the air to the maximum amount of water vapour the air can hold at the same temperature, expressed as a percentage (%). A visualization example is shown in Figure 13.

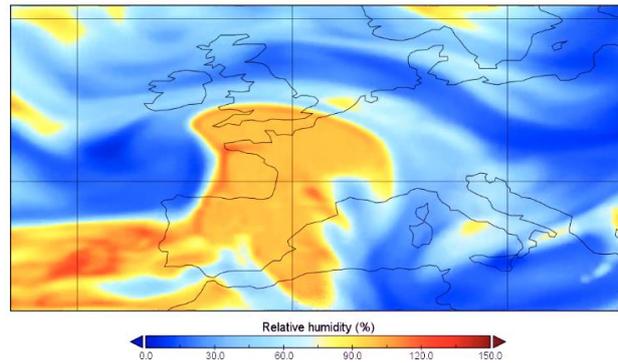


Figure 13. ERA5 Relative Humidity map.

- ❖ Wind Vectors: The east-west (zonal) component of wind is given by the U vector and the north-south (meridional) component of wind is given by the v component (see Figure 14).

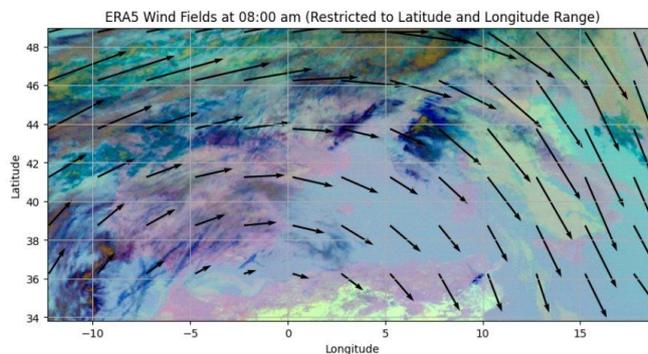


Figure 14. ERA5 Wind flow field.

Advection of potential contrails

To simulate the advection of contrails due to wind by advecting each aircraft's position, using wind vector fields and the second-order Runge-Kutta (RK2) method, you need to follow a step-by-step procedure that we are going to describe here.

The advection of potential contrails can be described by the partial differential equation:

$$\frac{dX}{dt} = V(X, t)$$

Where $X(t) = (x(t), y(t))$ is the position vector and $V(X(t), t) = (u(X(t), t), v(X(t), t))$ the wind velocity vector.

To do this, for each trajectory extracted from the ADS-B data of the form $W = \{X_0, X_1, \dots, X_T\}$ (where T is the time at which the image has been taken), for each waypoint given initially as X_i , to compute the advected \hat{X}_i need to iterate $X(t)$ over $t = i, \dots, T$ following these steps:

1. Initial setup:

- Set the aircraft position at an initial time t (ADS-B data): $X(t) = (x(t), y(t))$
- Set the velocity vector at the initial position $X(t)$ (ERA5 data):

$$V(X(t), t) = (u(X(t), t), v(X(t), t)), \quad \mathbf{k1} = V(X(t), t)$$

- Define as time step: $\Delta t = 1$

2. Estimate the **mid-point position** between the initial position $X(t)$ and the position at the following time step $X(t + \Delta t)$:

$$X\left(t + \frac{\Delta t}{2}\right) = X(t) + \frac{\Delta t}{2} \mathbf{k}_1$$

3. Estimate the **velocity at the mid-point** position:

$$V\left(X\left(t + \frac{\Delta t}{2}\right), t + \frac{\Delta t}{2}\right) = \left(u\left(X\left(t + \frac{\Delta t}{2}\right), t + \frac{\Delta t}{2}\right), v\left(X\left(t + \frac{\Delta t}{2}\right), t + \frac{\Delta t}{2}\right)\right),$$

$$\mathbf{k2} = V\left(X\left(t + \frac{\Delta t}{2}\right), t + \frac{\Delta t}{2}\right)$$

4. **Update** the position at the **following time step**:

$$X(t + \Delta t) = X(t) + \Delta t \mathbf{k}_2$$

5. **Iterate** this process until reaching $X(T)$ where T is the time of the image

6. **Store** $\hat{X}_l = X(T)$

Note that the ERA5 wind vectors have lower **spatial and temporal resolution than ADS-B so we need to **interpolate** each grid of wind vectors both **in the spatial and temporal domain**.*

2.2 Data used for the Physic-Driven contrail simulation task.

In our physics-driven model, we relied on ReAnalysis data (ERA5) to represent atmospheric variables such as wind, relative humidity, and temperature. However, given limitations in resolution of the data, we employed neural network techniques to reconstruct these quantities over the target domain.

3 Methods

In this section, we outline the methodology employed to attain the goal of correctly identifying regions of contrail formation in satellite imagery. We present various alternatives explored, elucidating the efficacy of each approach and the workarounds to the issues found. This includes a comprehensive examination of diverse model architectures and configurations, as well as the preprocessing and post-processing techniques utilized to achieve best possible results combining knowledge with model detections. Furthermore, we discuss the performance metrics obtained from validating models using the original GOES-16 data with which we have trained the models, as well as a validation of the extrapolation of the techniques to European satellite data from experiments conducted with MSG/SEVIRI.

Furthermore, we conduct an examination of the fundamental equations governing the phenomenon of ice crystal transportation caused by wind and the thermophysical characteristics of contrails. Additionally, we introduce a novel term to be integrated into these equations, addressing the impact of gravity on the dissipation of contrails. Subsequently, we introduce a model to simulate the evolution of contrails into non-linear clouds based on the afore-mentioned equations.

3.1 Data-Driven Contrail Detection Models

For this task, various approaches can be utilized to detect objects in images. For instance, one straightforward method involves detection using **bounding boxes**, which indicate the region of the image containing the object. However, for the task of contrail detection, this approach is suboptimal for two main reasons:

- Firstly, models may struggle to discern the underlying patterns of contrails solely from information about the region they occupy, especially since these regions can be quite large and may contain other objects such as different types of clouds.
- Additionally, for our purposes, simply receiving a prediction of the region where the contrail is located is insufficient. In the case of a geostationary satellite, being on one side of a region as opposed to the other could result in a difference of hundreds of kilometres, underscoring the need for additional information beyond just the detected region.

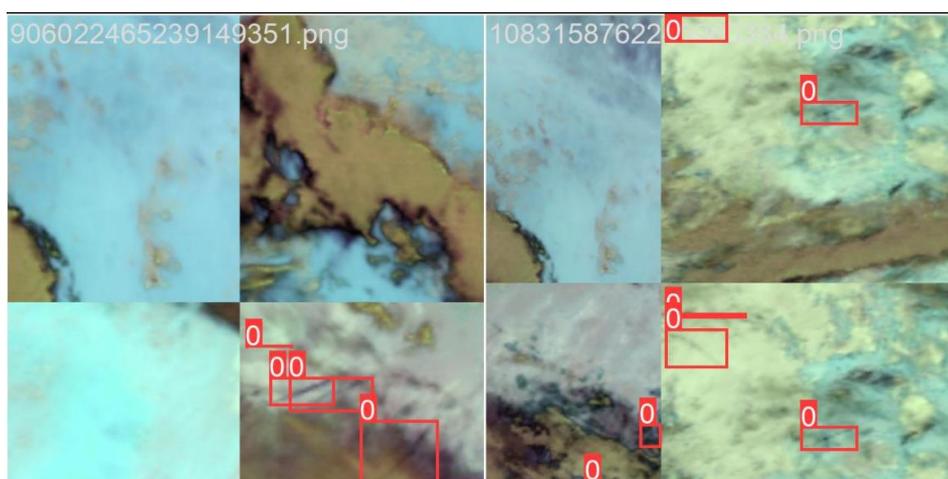


Figure 15. Contrail Bounding Box Detection Labels

Another approach involves **keypoint detection**. This method entails training the model to concentrate solely on the crucial pixels that define an object, enabling it to recognize the key characteristics. In our scenario, an initial idea was to train a model to execute this task, utilizing the extremes and certain intermediate points of the segment that delineates the primary axis of the contrail's direction as keypoints. While this approach yielded relatively satisfactory performance, further exploration of alternative methods resulted in superior outcomes.

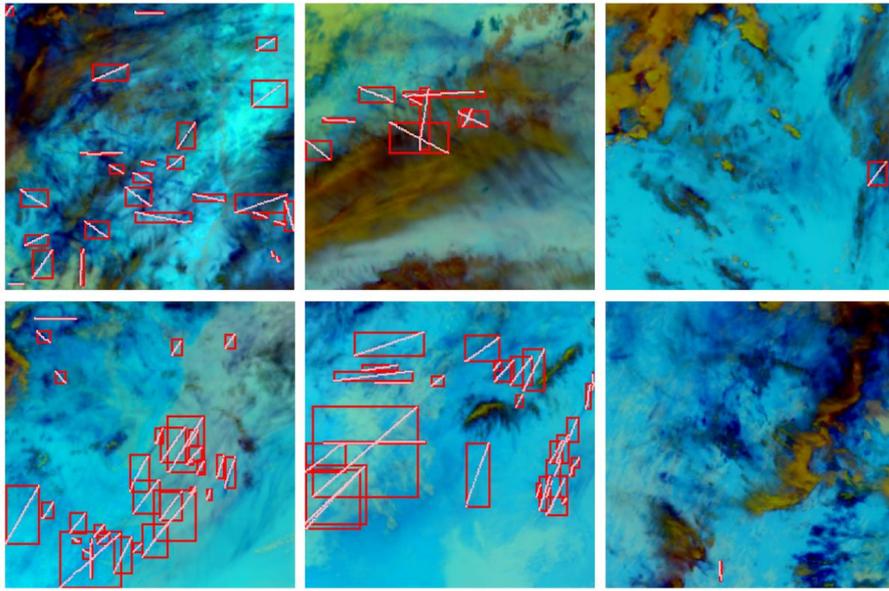


Figure 16. Contrail Keypoint Detection Segments

In the context of utilizing geostationary satellite imagery, where each pixel corresponds to a couple of square kilometres, even a small error of a few pixels can be translated to a significant error in terms of distance, potentially spanning many kilometres. Furthermore, clouds are dynamic entities; they undergo continuous changes and are better described as processes rather than static objects. Given this variability in shape and behaviour, employing **semantic segmentation** emerges as a more fitting strategy for accurately identifying them at their specific locations and capturing their precise shapes. Semantic segmentation involves determining, for each pixel location in the image, whether that pixel belongs to a contrail or not. This approach allows for a more granular understanding of the cloud distribution and facilitates precise delineation of their boundaries within the images.

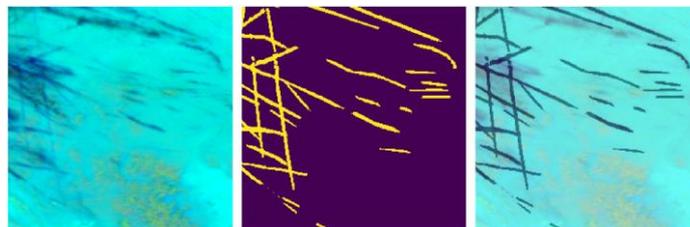


Figure 17. Contrail Semantic Segmentation Mask

While the semantic segmentation method appears suitable for our current problem, it presents a significant drawback when considering future applications: the model does not distinguish between individual contrail instances. In cases where regions of contrails overlap, the model is unable to provide information for each separate object. This limitation has several implications, such as:

- We cannot differentiate between aviation-induced cloudiness and regions with linear contrails occurring at different altitudes.
- We cannot track the evolution of each individual contrail over time.
- We cannot match each contrail to the aircraft that generated it.
- We cannot accurately study the dynamics of cloud dissipation phenomenon.

To address this challenge, we have two potential solutions:

- ✓ Firstly, we can develop a post-processing procedure that can be seamlessly integrated as a final step to the semantic segmentation model outputs. This procedure would extract each contrail instance from a binary mask, enabling individual identification and analysis.
- ✓ Alternatively, we can directly use an **instance segmentation** model, which is directly fed with polygons that outline each contrail instance in the image.

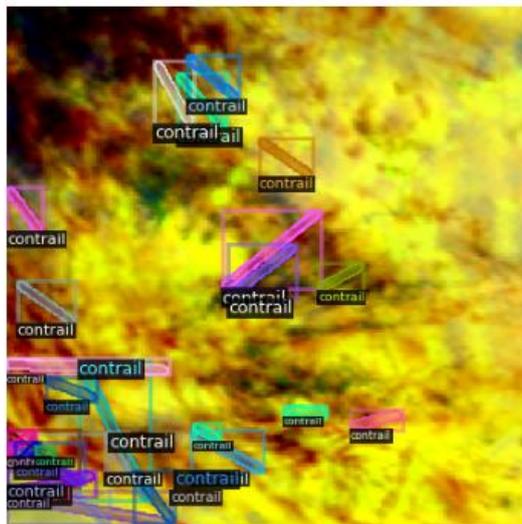


Figure 18. Contrail Instance Segmentation Polygons

However, we realized that for the second approach, we would also need to convert the masks into contrail polygons for training purposes. This is necessary because the labels provided by the dataset we intend to use, the OpenContrails Dataset, are structured to train a semantic segmentation model.

To address this requirement, we developed a methodology specifically tailored for this transformation, which will be detailed in the next subsection. As a result, we trained both instance segmentation and semantic segmentation models and compared their performances.

3.1.1 Methodology for Instance Retrieval

Given the substantial number of masks and the diverse contrail intersection scenarios in them (See Figure 19) our goal is to devise a methodology that effectively isolates each contrail.

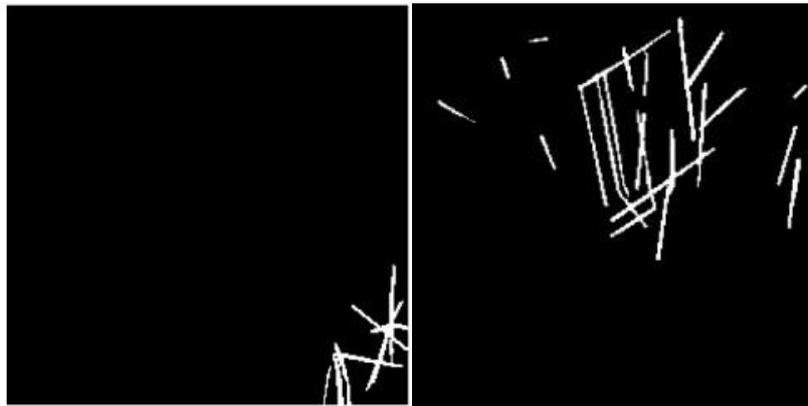


Figure 19. Binary ground truth masks in low (left) and high (right) contrail coverage scenario.

Considering the linear nature and inherent width of each contrail feature, we conceptualize the most fitting ellipsoid to approximate it (See Figure 20). Then, we retrieve the original contour of each instance and designate it as its label. Initially, we pinpoint the longest axis of the ellipsoid, representing the advected aircraft trajectory, and subsequently retrieve the complete shape.

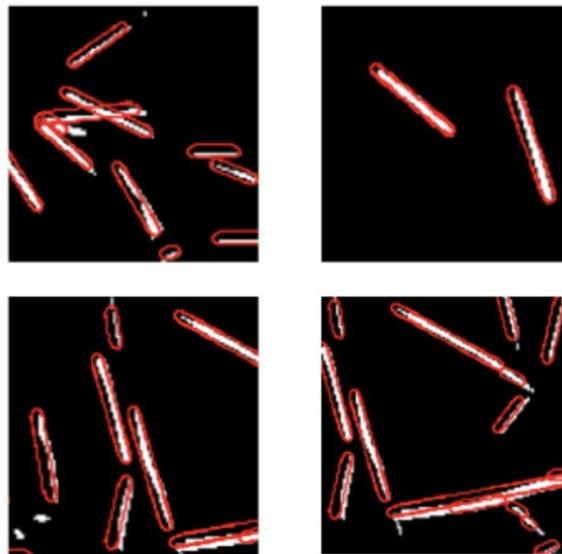


Figure 20. Examples of the ellipsoids aiming to approximate the proposed methodology.

We observe an implicit assumption of linearity in contrail shapes. In instances with a slight curvature (See the right plot in Figure 21), the contrail width allows the identification of the linear segment that represents the aircraft trajectory. Even when there is more pronounced curvature (See the left plot in Figure 21), our approach proficiently identifies the contrail as a combination of distinguishable subsegments, supporting the justification for this simplifying assumption.

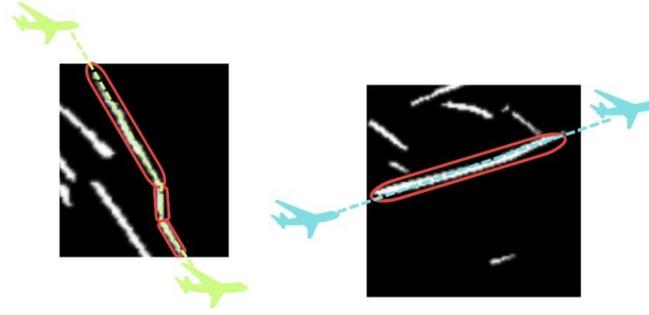


Figure 21. A highly curved advected contrail shape (left) and a slightly curved linear contrail.

One main reason why for this study we prioritize contrail isolation over precise contour accuracy is due to persistent discrepancies among human labellers in annotations from [5], despite verification from sources like ADSB and weather data. Nevertheless, refining this methodology for enhanced pixel-level accuracy and broader applicability may be necessary.

The following section briefly introduces certain background concepts to ensure consistent notation throughout the explanation.

Preliminary concepts and notation

Let J be an open finite subset of R^2 , defining the spatial extent of a 2-dimensional image of size. We introduce the following definitions:

A finite set of ordered pairs $G = \{(i, j) \mid 1 \leq i \leq n; 1 \leq j \leq m; n, m \in N\}$ representing the positions of the image elements within the spatial extent J is referred to as image grid.

A collection of uniformly shaped, uniformly spaced and equally distant samples $P = \{P_{(i,j)} \in J \mid (i, j) \in G\}$ satisfying

- a) $J = \cup_{(i,j) \in G} P_{(i,j)}$
- b) $P_{(i,j)} \cap P_{(v,w)} = \emptyset \quad \forall (i, j), (v, w) \in G \quad (i, j) \neq (v, w)$
- c) $Area(P_{(i,j)}) = Area(P_{(v,w)}) \quad \forall (i, j), (v, w) \in G$

are referred to as image pixels and represent the smallest image elements.

Given a fixed number of spectral channels C , the image function is defined as $D: J \rightarrow R^C$ defined as $D(P_{(i,j)}) = l_{(i,j)}$ where $l_{(i,j)} = (l_{(i,j)}^1, \dots, l_{(i,j)}^C)$ is the vector representing the light intensity values of $P_{(i,j)}$ at each spectral channel. The digital image is the matrix $I \in M_{m \times n}(R^C)$ defined by function D .

With the provided definition of an image grid and a given set of objects of interest $F = \{O_r \mid r = 1, \dots, R\}$ verifying

- I. $F = \{P_{(i,j)} \mid (i, j) \in \cup_{r=1}^R G_r\}$ where $\cup_{r=1}^R G_r \subset G$
- II. $O_r = \{P_{(i,j)} \mid (i, j) \in G_r \subset G\}$ for each $r = 1, \dots, R$
- III. O_r is a connected subset of R^2 with the usual topology for all $r = 1, \dots, R$

we can define a function that classifies image pixels, generating a binary semantic segmentation mask. We only define the binary case because for this study we can assume that all objects of interest in F either share the same class or do not necessitate differentiation between classes.

Let G be an image grid and $S^F: G \rightarrow \{0,1\}$ be a function that for each point in the grid $(i,j) \in G$ takes the value $S^F(i,j) = 0$ if $P_{(i,j)} \notin F$ and $S^F(i,j) = 1$ if $P_{(i,j)} \in F$ then, the binary semantic segmentation mask of the image is the matrix $M_{S^F} \in M_{m \times n}(\{0,1\})$ of the function S^F . The set F is often referred as set of foreground pixels, while $B = P - F$ is called set of background pixels.

Description of the instance retrieval methodology

Given F and S^F , in accordance with the preceding definitions, and considering the corresponding binary semantic segmentation mask M_{S^F} , wherein K linear contrails are segmented, the goal is to find the set $X^* = \{X_k^* \mid k = 1, \dots, K\}$ where $X_k^* = \{(x_0^{*k}, y_0^{*k}), \dots, (x_{t_k}^{*k}, y_{t_k}^{*k})\}$ is the set of coordinates that best describes the contour of each individual contrail k for each $k = 1, \dots, K$.

For the sake of simplicity, if the mask $M_{S^F} \in M_{m \times n}(\{0,1\})$ has $R \leq K$ connected components, we decompose it into R binary masks $\{M_{S^F}^r\}_{r=1}^R$, each containing a single connected component. The decomposition is performed using the algorithm outlined in [17]. Then, since $X^* = \bigcup_{r=1}^R X^{r*}$, we independently identify X^{r*} within each $M_{S^F}^r$ following the steps detailed next. To lighten the notation, we will omit the superscript r . However, it's important to note that these steps should be executed within each connected component mask instead of within each original mask to enhance performance and speed.

- **Binary Mask Parametrization:** Before accurately identifying the segments that represent the longest axis of the ellipsoid that best characterizes each contrail within a given connected component mask, our initial task is to locate **all existing line segments within that mask**. This is why we want to find each set L of the form $L_{ab} = \{(x,y) \in G \mid S_F(x,y) = 1 \exists a,b \in R : y = ax + b\}$.

Consider $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ as the angle of the normal line to $y = ax + b$ with respect to the origin and let $\rho \in R$ be its algebraic distance from the origin. The line equation can be expressed in its polar form as $\rho = x \cos(\theta) + y \sin(\theta)$ and each L_{xy} can be rewritten as $L_{\theta\rho} = \{(x,y) \in G \mid S_F(x,y) = 1; \exists \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \exists \rho \in R : \rho = x \cos(\theta) + y \sin(\theta)\}$.

Notably, this new polar equation represents a sinusoidal curve in the $\rho - \theta$ parameter space. Therefore, every point (x,y) in image space uniquely corresponds to a sinusoidal curve in parameter space. This duality between both spaces also indicates that **there exists a correspondence between a set of co-linear points in the original image space and a set of concurrent curves in the parameter space** [18]. Given this perspective, we can approach the challenge of line identification in image space as the task of identifying points of intersection among multiple curves in parameter space.

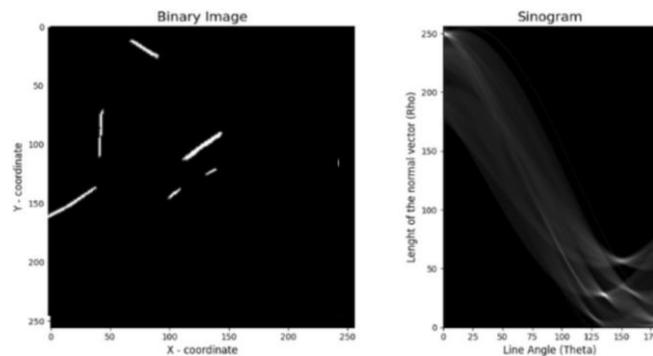


Figure 22. Transformation of the binary mask to the $\rho - \theta$ space

- **Thresholding based on Hough Accumulator:** This conversion of the binary mask into a two-dimensional parameter space is achievable through either the Radon or Hough Transform [19]. The resultant graph, referred to as the Radon or Hough sinogram, illustrates the intensity values of sinusoidal curves. Although the Radon transform achieves a task akin to the Hough Transform, it operates in a continuous parameter space instead of the discretized cells used in the Hough Transform. In this study, we derive the sinogram by employing the Hough Accumulator, defined as the function

$$H(\rho, \theta) = \sum_{(x,y) \in G} S_F(x, y) \delta(\rho - x \cos(\theta) - y \sin(\theta))$$

where $\delta(\cdot)$ is the Dirac Delta function. We will only consider as parameters defining a line in the original image the (ρ, θ) satisfying $H(\rho, \theta) \geq c$ for a selected threshold c , that will represent the minimum number of concurrent sinusoids that must intersect at (ρ, θ) to qualify these parameters as indicative of a line within the image. The **threshold** c can also be interpreted as the **minimum length we want to establish for the detected lines**. In the context of our masks, it has been observed that this parameter must be set to ≈ 12 pixels, that is approximately the medium width of the contrails we are targeting to isolate. This is because we want to discard from this line detections that have a length equal to or less than the smallest axis of the approximated contrail ellipsoids.

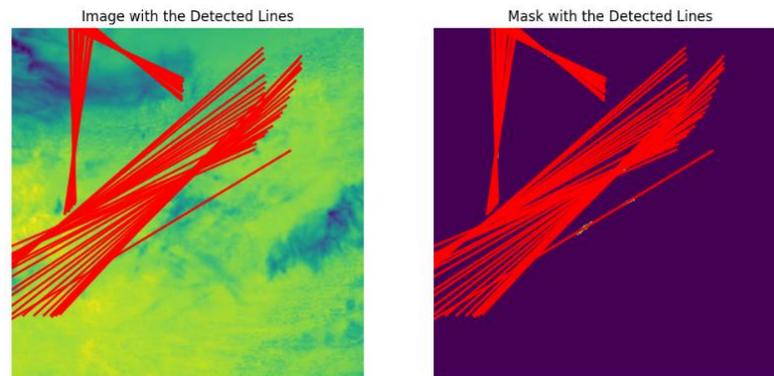


Figure 23. Lines detected with the Hough transform.

- **Clustering:** After successfully identifying the set of targeted line parameters $\chi_c = \{(\rho, \theta) \mid H(\rho, \theta) \geq c\}$ we proceed to cluster the points within this set. Utilizing a fixed size preselected through black box optimization, **we determine the centroid for each cluster**. The resultant set of centroids, denoted as χ'_c and computed as the median values within each cluster, becomes the line parameters for our original image. The corresponding image lines are defined as: $L_{\chi'_c} = \{L_{\rho\theta} \subset G \mid (\rho, \theta) \in \chi'_c\}$

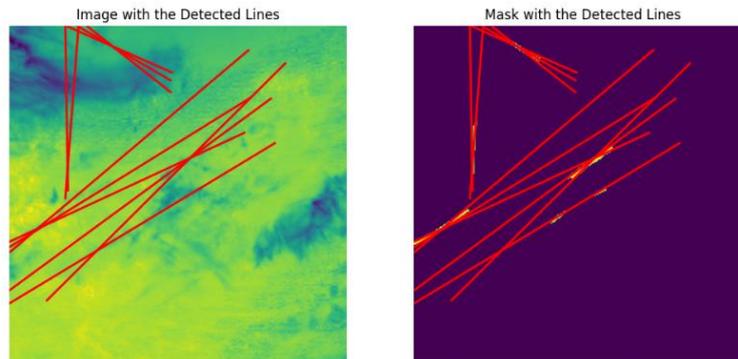


Figure 24. Lines detected with Hough transform after removing similar segments.

- **Refinement:** The final segments would be the *intersection of the detected lines with the foreground pixels* in the mask as shown in Figure 25. To ensure no contrails have been lost during the clustering process, we take the following steps:
 - Intersect the lines with the foreground pixels.
 - Create a new mask with the foreground pixels that does not belong to the intersection.
 - Create a separate mask for each connected component of the new mask.
 - Apply steps I-III independently to each of these sub-connected component masks. The centroids identified in this process for each of this sub masks are added to the set $L_{\chi'_c}$.
 - Iterate this process until there are no more foreground pixels left.

Once all foreground pixels have been processed, we *address the possibility of duplicates* that may have emerged during the iteration. To accomplish this, we merge sets of coordinates that coincide, are very close to each other, or where one set is a subset of the other.

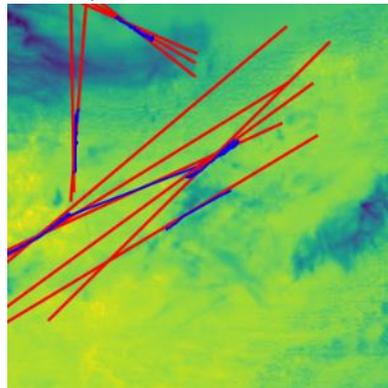


Figure 25. Subsegments retrieved from the lines detected.

- **Contouring:** The final set $L_{\chi'_c}$ contains the segments that represent the longest axis of the contrails in the image. To now *retrieve the ellipsoid* that approximates the contour of each contrail, and then retrieve the actual contour shape, we will do the following to each segment separately:
 - Create a black mask and plot all the detected segments.
 - Apply the *morphological operation of dilation* to the mask in 3 iterations to thicken the segment.

- Intersect separately each connected component with the original mask *and find the contour of the foreground pixels using the Canny edge detector* [20].
- Add the coordinates of the contours to the set X^*

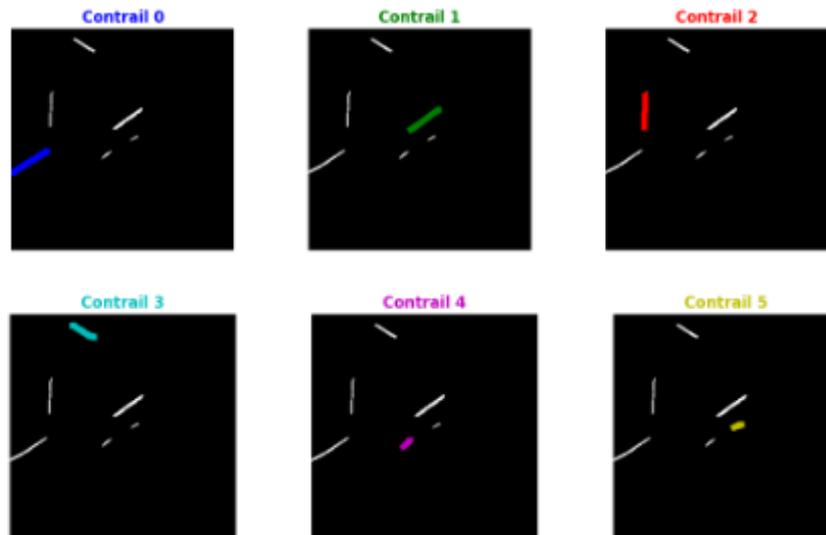


Figure 26. Independent contrail instances retrieved from the original binary mask.

3.1.2 U-Net Model

The U-Net model, introduced in a [21] in 2015, is renowned for its effectiveness in semantic segmentation tasks. Its success has inspired numerous modifications and adaptations over the years. The U-Net is built on an encoder-decoder architecture, which can be formally described as follows:

Given a feature space ϕ and a lower-dimensional space ψ known as the latent space, the **encoder** is defined as a mapping $C: \phi \rightarrow \psi$ that transforms each input data point $x \in \phi$ into a lower-dimensional representation $z = C(W_c, b_c; x) \in \psi$ and it is uniquely defined by the parameters $\{W_c, b_c\}$. The representation z is referred to as the latent code.

Similarly, given a feature space ϕ and a latent space ψ the **decoder** is defined as a mapping $D: \psi \rightarrow \phi$ that transforms the latent code $z \in \psi$ into a reconstruction of the original data point $r = D(W_d, b_d; z) \in \phi$ where $\{W_d, b_d\}$ are the model parameters.

With these definitions, given an encoder $C: \phi \rightarrow \psi$ and a decoder $D: \psi \rightarrow \phi$, the **encoder-decoder** network, also known as autoencoder, can be defined as an application $A: \phi \rightarrow \psi \rightarrow \phi$ such that $A(x) = r$ for each $x \in \phi$, where $r = D(W_d, b_d; z)$ and $z = C(W_c, b_c; x)$.

In the U-Net model, each layer in both the encoder and the decoder performs discrete 2-dimensional **convolution** operations. This convolution operation $*$ is given by

$$(h * W + b)(i, j) = \sum_{m=-k}^k \sum_{n=-k}^k h(i-m, j-n)W(m, n) + b$$

where h is the input feature map, K is the convolution kernel, b is the bias term, (i, j) are the spatial coordinates of the output and k is the half-width of the kernel of size $(2k + 1) \times (2k + 1)$.

The output of each layer $i \in \{1, \dots, n_1\}$ of the encoder is given by $h_i = f_i(K_i * h_{i-1} + b_i)$, where f_i in our case denotes the activation function σ_i of the i -th layer followed by a normalization operation N and a pooling operation.

The Rectified Linear Unit (ReLU) function given by $\text{ReLU}(x) = \max(0, x)$, along with its variants such as Leaky ReLU or Gaussian Error Linear Unit (GeLU), is commonly employed as the activation function for intermediate layers in neural networks due to its simplicity and effectiveness in promoting nonlinear behaviour within the network. For the output layer, we utilize in stead the Sigmoid activation function defined as $\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$. This function is specifically designed to map the values of each pixel to probabilities. As for the pooling operation, various alternatives exist to downsize the feature maps of each layer. Popular methods include

- a) **Max-pooling:** Involves creating new feature maps by selecting each element of the pooling layer as the maximum value within its neighbourhood region. For each spatial location (i,j) , the corresponding value in the pooling layer would be $p_{ij} = x_{p'q'k'}$ such that $x_{p'q'k'} = \max(x_{pqk}) \in R_{ij}$, where R_{ij} denotes the region formed by the set of neighbours of p_{ij} , and each x_{pqk} represents the element at position (p, q) of R_{ij} in feature map k .
- b) **Average-pooling:** Computes each element of the pooling layer as the arithmetic mean of its neighbours in the input layer. For each spatial location (i,j) , the corresponding value in the pooling layer would be $p_{ij} = \frac{1}{|R_{ij}|} \sum_{x_{pqk} \in R_{ij}} x_{pqk}$.

In the U-Net model, the layers in the decoder perform **transposed convolutions** and make use of upsampling operations that are the inverse pooling operations that retrieve feature maps with higher dimensionality.

Additionally, the U-Net introduces the concept of **skip connections** in the decoder. Therefore, the output of the i – th layer is given by $\widehat{h}_i = f(W_i^T * [\widehat{h}_{i-1}, h_{n_l-i}] + b_i)$ where $[\widehat{h}_{i-1}, h_{n_l-i}]$ denotes the concatenation of feature maps from the decoder and the corresponding encoder layer. By directly connecting early and late-stage layers, skip connections help mitigate information loss during downsampling and facilitate the flow of gradients during training. This allows the model to better preserve spatial information, capture fine details, and produce more accurate predictions, leading to improved performance.

The U-shape of this network is primarily due to the incorporation of these skip connections (see Figure 27).

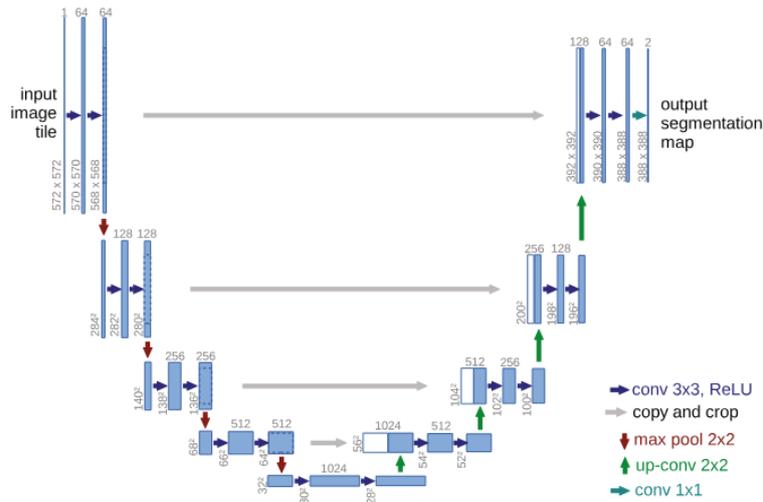


Figure 27. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations [21].

In recent years, Transformer networks, known for their prowess in processing sequential data in Natural language Processing (NLP), have been adapted for image processing, forming **Vision Transformers** (ViTs). These models leverage attention layers to capture global context and long-range dependencies within images efficiently. Self-attention enables each spatial location in the input feature map to compute attention weights based on its relationship with all others, facilitating parallel computation and making ViTs well-suited for tasks with large feature maps. Considering our problem, incorporating these layers could notably improve the model's performance, especially given the substantial input size (256 x 256 x 3) and the complex interrelations among image elements.

For ViTs, the inputs are not full images but rather a series of vectors called image tokens. These **tokens** are computed as following these steps:

- Patching: The input image of size (H, W, C) is divided into N_p non-overlapping patches of size (P, P, C) , denoted as $\{p_1, \dots, p_{N_p}\}$ where $N_p = \frac{W}{P} \times \frac{H}{P}$ is the total number of patches
- Flattening: Each patch p_i is resized into a vector v_i of size $D = P \times P \times C$
- Tokenization: Each final token t_i is obtained by $t_i = v_i + PE_i$ where $PE_i = (pe_1, \dots, pe_D)$ stands for the positional embedding vector. Here $pe_d = \sin\left(\frac{i}{10000^{\frac{d}{D}}}\right)$ if d is even and

$$pe_d = \cos\left(\frac{i}{10000^{\frac{d}{D}}}\right) \text{ if } d \text{ is odd for } d = 1, \dots, D$$

As a general formulation, in a **self-attention layer** we transform the input feature map h into three sets of vectors: queries (Q), keys (K), and values (V). This transformation is accomplished by applying learnable weight matrices W_Q, W_K, W_V . Thus, for each spatial location, we compute the following vectors

$$q_{i,j} = h_{i,j} \cdot W_q, \quad k_{i,j} = h_{i,j} \cdot W_k, \quad v_{i,j} = h_{i,j} \cdot W_v$$

Then, the attention weights are computed as $\alpha_{i,j} = \text{softmax}\left(\frac{q_{i,j} \cdot k_{i,j}^T}{\sqrt{d_k}}\right)$, where d_k represents the dimensionality of the key vectors $k_{i,j}$. Finally, the output feature map of the layer would be given by

$$h_{i,j} = \sum_{m=1}^H \sum_{n=1}^W \alpha_{i,j,m,n} \cdot v_{m,n}$$

Research has led to many attention layer adaptations like multi-head attention, scaled-dot product attention, conv-attention, multi-axis attention, or efficient multi-head self-attention, among others. These, combined with diverse integration strategies involving convolutional layers, activations, and normalizations, have given rise to **numerous state-of-the-art architectures**. These architectures are tailored to specific tasks and domains, offering distinct advantages based on requirements and data characteristics.

When applied as backbones for models like U-net, they significantly enhance performance, particularly in complex tasks such as segmenting complex contrail patterns in satellite imagery. Incorporating these **advanced backbones** will ensure superior feature extraction and segmentation precision for our application.

Drawing insights from the Google Research competition leaderboard, our strategy entailed choosing the most effective backbone architectures and investigating their limitations. This method enabled us to uncover potential solutions to enhance overall performance. We deemed this approach optimal for leveraging the discoveries made by fellow researchers, thereby facilitating our own learning process and advancing our understanding of model design and optimization.

The **Next Generation Vision Transformer (NextViT)** [22] has emerged as a high-performing model in several leaderboard solutions of this competition. For this model, in its 2022 paper, the authors conducted a performance comparison on the widely used ImageNet dataset, which serves as a benchmark for evaluating model performance. The comparison results, as depicted in Figure 28 clearly showcase NextViT's superiority in image classification tasks on this specific dataset. Therefore, recognizing the efficacy of NextViT, we made the deliberate choice to incorporate this backbone architecture as one of the primary options for our U-Net model.

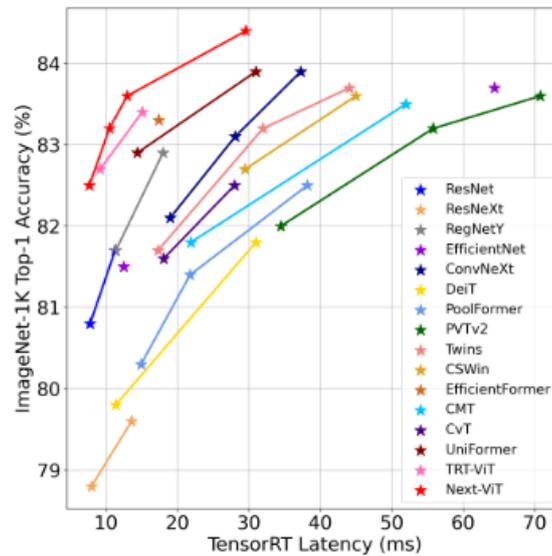


Figure 28. Comparison among Next-ViT and efficient Networks, in terms of accuracy-latency trade-off on the ImageNet-1K for classification on TensorRT [22].

For similar reasons, we have also decided to test as backbone alternatives for our model the following architectures: **Co-Scale Conv-Attentional Image Transformers (CoaT)** [23] and **Multi-Axis Vision Transformer (MaxViT)** [24].

Some of the main characteristics of each of them are the following:

1. **NextViT**: The architecture of this new generation of transformers is illustrated in Figure 29. The left column displays the overall hierarchical structure of Next-ViT. The middle column details the components: the Next Convolution Block (NCB) and the Next Transformer Block (NTB). The right column provides a detailed visualization of the multi-head convolutional attention (MHCA), efficient multi-head self-attention (E-MHSA), and the optimized MLP modules. This enhanced hybrid architecture combines the strengths of CNNs and transformers, resulting in a powerful and efficiently deployable CNN-Transformer model.

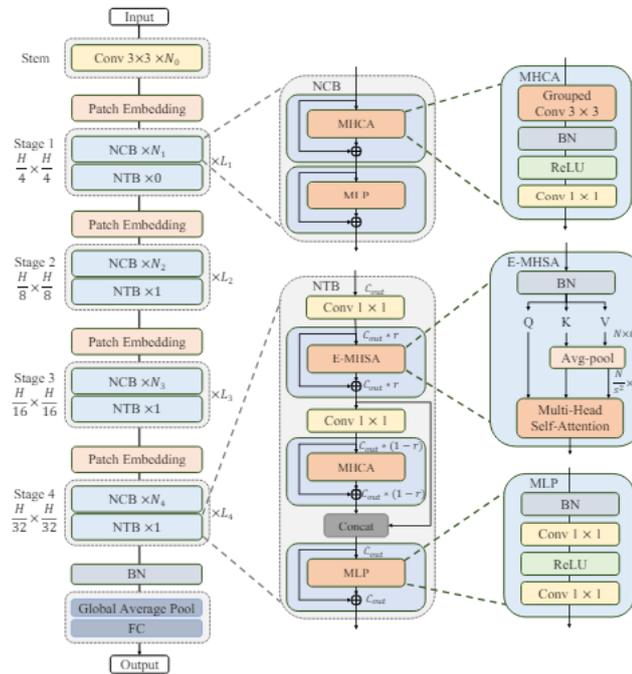


Figure 29. Overall architecture of the NextViT model

- CoaT:** The main contributions of this network are the introduction of efficient conv-attention operations and co-scale mechanisms. In the conv-attention module a new factorized attention module defined as is introduced. The overall structure of this block is shown in Figure 30. In it a convolutional position encoding is applied to the image tokens from the input. The resulting features are fed into a factorized attention with a convolutional relative position encoding.

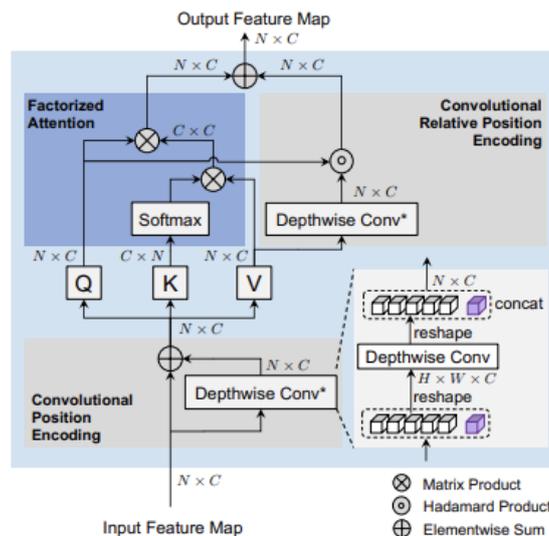


Figure 30. . Illustration of the conv-attentional technique [23].

The co-scale mechanism, on the other hand, is designed to integrate fine-to-coarse, coarse-to-fine, and cross-scale information into image transformers. Two types of blocks are proposed: serial and

parallel blocks. Within the parallel blocks, two co-scale variants are introduced: direct cross-layer attention and attention with feature interpolation (see Figure 31).

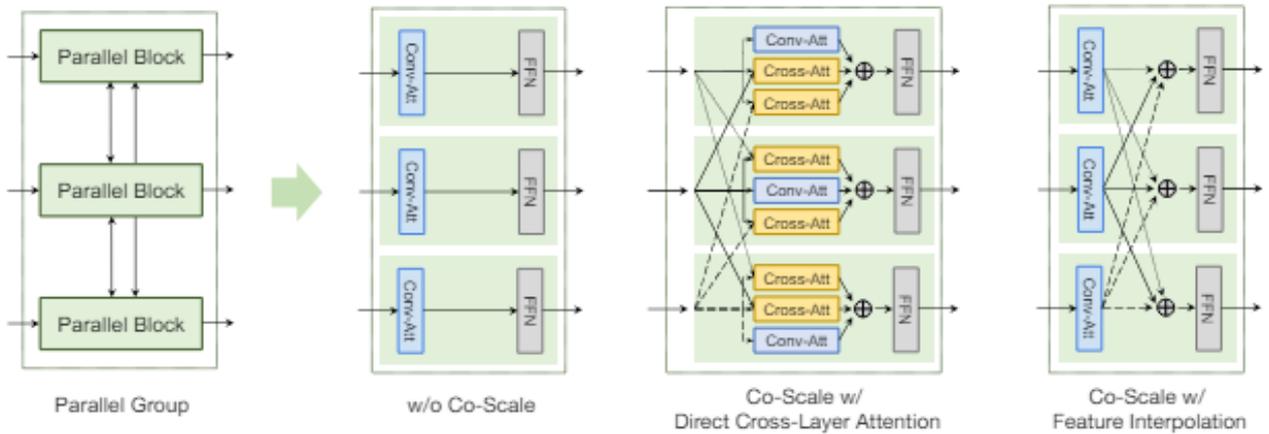


Figure 31. Schematic illustration of the parallel group in CoaT [23].

The overall architecture is shown in Figure 32. On the left it shown CoaT-Lite architecture, that consists of serial blocks only, where image features are down-sampled and processed in a sequential order. On the right it is shown the CoaT architecture, that consists of serial blocks and parallel blocks. Both blocks enable the co-scale mechanism.

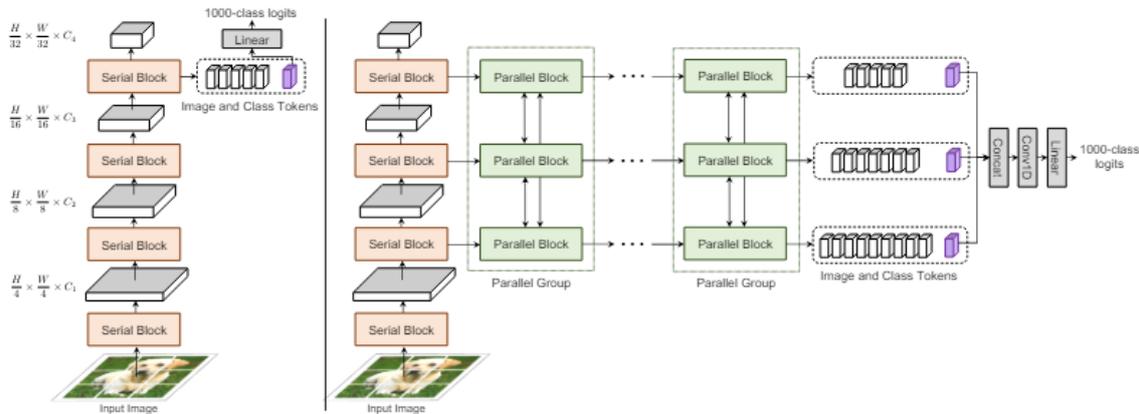


Figure 32. CoaT Model Architecture

- MaxViT:** This work introduces a unified design that combines the efficiency of convolutional operations with the flexibility of sparse attention, enabling it to scale effectively to very large datasets. The proposed model, called multi-axis attention, features two key components: blocked local attention and dilated global attention.

These design choices facilitate global-local spatial interactions for inputs of any resolution, maintaining only linear complexity. The architecture, as shown in Figure 33, follows a hierarchical design like ConvNets and ResNet but introduces a new type of basic building block that integrates Mobile Inverted Bottleneck Convolution (MBConv), block attention, and grid attention layers. For simplicity, normalization and activation layers are omitted from the description.

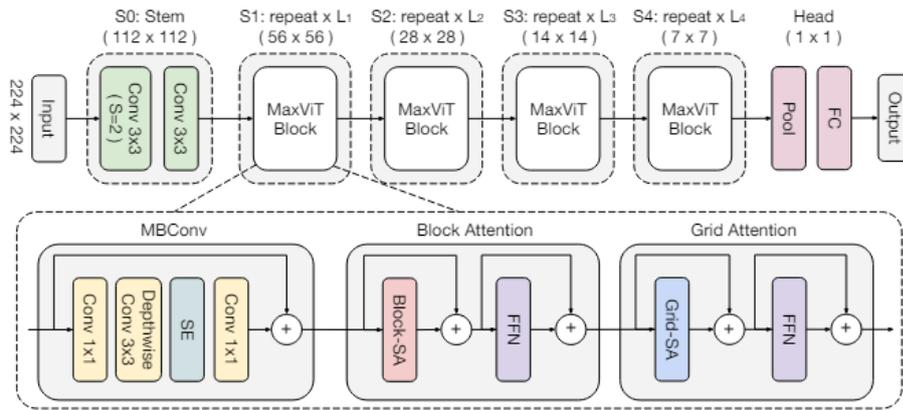


Figure 33. Overall architecture of the MaxViT model

The results for the MaxViT model are not included in this document due to the inability to achieve satisfactory performance. A further exploration is needed to find the issues that might have caused these unsatisfactory results.

3.1.3 YOLO Model

The YOLO (You Only Look Once) model is a convolutional neural network (CNN) proposed by Joseph Redmon in 2015. Its popularity stems from its high accuracy despite its compact size and the ability to be trained on a single GPU [25].

In its early versions, YOLO was implemented in C code within the custom deep learning framework called "Darknet." Later, Ultralytics followed with the YOLOv3 repository in PyTorch, which evolved into YOLOv5 in 2020. Thanks to its open-source nature, various developers have created several YOLO derivatives in recent years, such as Scaled-YOLOv4, YOLOR, and YOLOv7, by making small modifications to Ultralytics' YOLOv5 model. Meanwhile, Ultralytics has been working on releasing the latest and most advanced version of YOLO, called YOLOv8, on January 10, 2023. This version builds on the success of previous iterations and introduces new features and improvements to enhance performance, flexibility, and efficiency. Figure 34 illustrates the temporal evolution of this model.

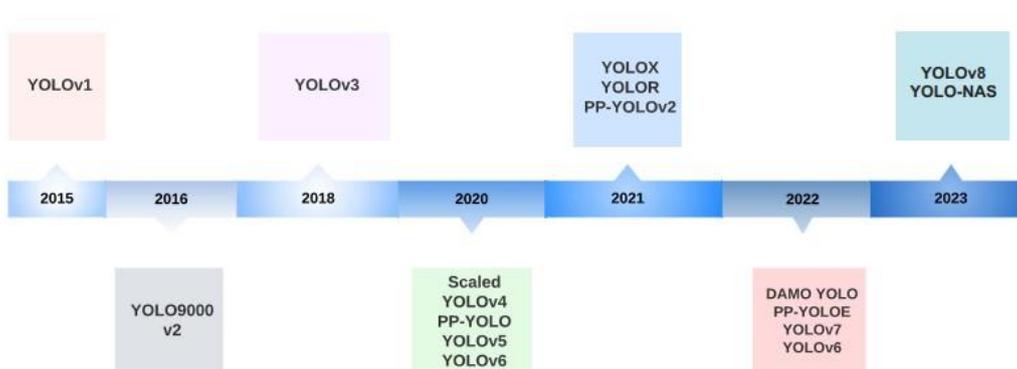


Figure 34. Temporal evolution of the distinct YOLO versions

The architecture of the YOLOv8 convolutional network features a pyramidal-shaped backbone (encoder) with 5 blocks, allowing for object detection at different scales by upsampling and concatenating the outputs across different layers.

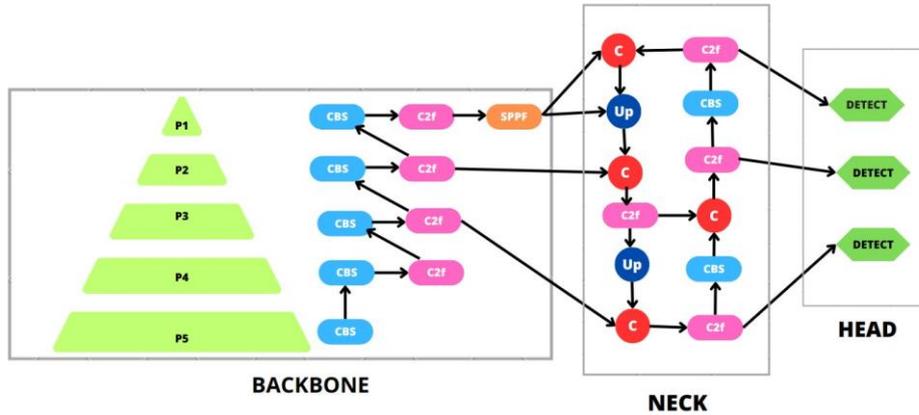


Figure 35. Overall Architecture of the YOLOv8 model.

In Figure 35, the Up and C blocks stand for Upsampling and Concatenation (across the channel dimension), respectively. The CBS module stands for Convolution - Batch Normalization - SiLU activation. This last two operations are defined as follows:

- Batch Normalization: Given a batch of size B and layer outputs of size (H, W, C) , first the mean and the standard deviation across the batch for each channel c are computed as

$$\mu_c = \frac{1}{B \cdot H \cdot W} \sum_{i=1}^B \sum_{j=1}^H \sum_{k=1}^W h_{i,j,c,k} \quad \sigma_c^2 = \frac{1}{B \cdot H \cdot W} \sum_{i=1}^B \sum_{j=1}^H \sum_{k=1}^W (h_{i,j,c,k} - \mu_c)^2$$

Then, the output of each layer is normalized as $h'_{i,j,c,k} = \frac{h_{i,j,c,k} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$, using ϵ to avoid division

by zero. Finally, the normalized output is shifted and scaled as

$$y_{i,j,c,k} = \gamma_c \cdot h'_{i,j,c,k} + \beta_c$$

Where γ_c and β_c are parameters learned by the network that ensure the network can adjust the normalized features to any desired mean and variance, instead of fixing the mean and variance to zero and one values that can lead to vanishing or exploding gradients.

- Sigmoid Linear Unit (SiLU) activation: It is a non-linear activation providing a smooth, non-saturating response, which helps mitigate the vanishing gradient problem. It can be expressed as $SiLU(x) = x \cdot \sigma(x)$ where $\sigma(x) = \frac{1}{1 + \exp(-x)}$.

The main block within the YOLOv8 framework, however, is the C2f block, that combines the ELAN (Efficient Layer Aggregation Networks) concept from YOLOv7 [26] with the C3 module from YOLOv5. This new block ensures that the gradient flow is used effectively while maintaining the network's lightweight nature, as shown in Figure 36.

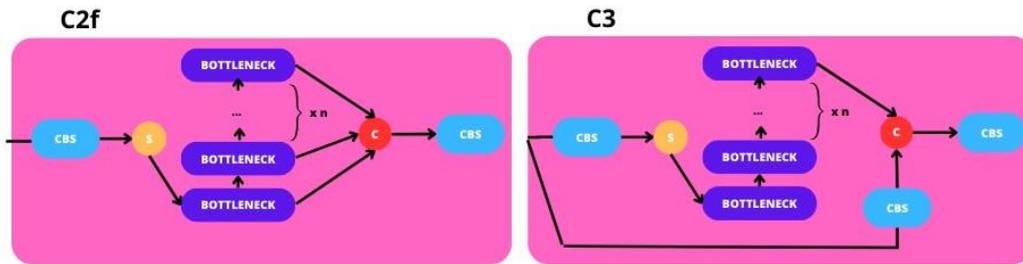


Figure 36. A comparison between the C2f Block of YOLOv8 (left) and C3 Block of YOLOv5 (right).

In the previous figure, S stands for the split operation and the bottleneck is composed of sequential CBS blocks with a residual connection or shortcut as shown in Figure 37.

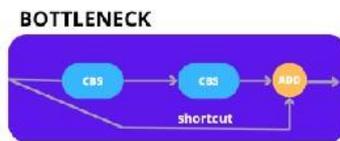


Figure 37. Bottleneck Block in YOLOv8

Finally, the SPPF (Spatial Pyramid Pooling Fusion) module in YOLO is used to capture information at different spatial scales and merge it to obtain a global representation. Its structure is depicted in Figure 38.

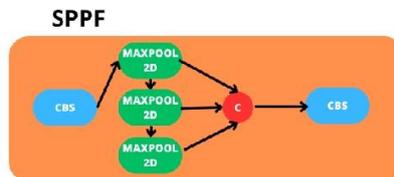


Figure 38. Spatial Pyramid Pooling Fusion Block in YOLOv8

The output Detect block computes in parallel two sequential CBS blocks plus a 2D convolution, followed in the first case by a Bounding Box Regression Loss and in the second by a Classification Loss, as shown in Figure 39. The model has a decoupled head, which means that it can be adapted to perform different tasks:

- ❖ YOLOv8: performs object detection with bounding boxes
- ❖ YOLOv8-seg: performs object detection with bounding + instance segmentation
- ❖ YOLOv8-pose: performs object detection with bounding + keypoint detection

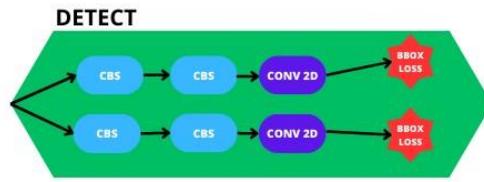


Figure 39. Detect Block in YOLOv8

- Bounding Box Regression Loss: Utilizes the Complete Intersection over Union (CIoU) + + Distribution Focal Loss (DFL) function.

The CIoU function is based on the similarity between the area of the detected object B and the area of the ground truth object B_{gt} , also considering the distance between the bounding boxes. It is given by the equation

$$\mathcal{L}_{CIoU} = 1 - IoU + \mathcal{R}_{DIoU} + \alpha v$$

Having

- The Intersection over Union defined as

$$IoU = \frac{|B \cap B_g|}{|B \cup B_g|}$$

- The Distance IoU penalty term given as

$$\mathcal{R}_{DIoU} = \frac{d^2(b, b_{gt})}{c^2}$$

where b and b_{gt} are the centre locations of B and B_{gt} respectively, d is the Euclidean distance and c is the diagonal of the smallest box covering B and B_{gt} .

- v known as the trade-off parameter and given by

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w_{gt}}{h_{gt}} - \arctan \frac{w}{h} \right)^2$$

where w and h and the width and height of B , respectively, and w_{gt} , h_{gt} the width and height of B_{gt} .

- α called compensation parameter and defined as

$$\alpha = \frac{v}{1 - IoU + v}$$

On the other hand, the DFL function focuses on the distribution existing in the environments close to the object. Thus, given a label $y \in [y_i, y_{i+1}]$ and given h_i and h_{i+1} the outputs of layer i and $i + 1$, the DFL function is defined as

$$DFL(h_i, h_{i+1}) = -(y_{i+1} - y) \log(h_i) + (y - y_i) \log(h_{i+1})$$

- Classification Loss: The VariFocal Loss (VFL) is based on the Binary Cross Entropy (BCE) loss function. BCE is commonly used for classification tasks and measures the difference between two probability distributions for a given random variable or set of events. Given the true distribution p and the estimated distribution q , the BCE is given by:

$$H(p, q) = - \sum_{x \in \chi} p(x) \log(q(x)) + (1 - p(x)) \log(1 - q(x))$$

To improve this function and scale the losses the VFL is given as

$$VFL(p, q) = -q(q \log(p) + (1 - q) \log(1 - p)) \quad \text{if } q \geq 0$$

and

$$VFL(p, q) = -\alpha p^\lambda \log(1 - p) \quad \text{if } q = 0$$

Here, p^λ is a scaling factor for the loss, and α is a regularization term.

3.1.4 Mask-RCNN Model

The Mask Region-based Convolutional Neural Network (Mask R-CNN) [27] It extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression.

In contrast to the YOLO model, Mask R-CNN is considered a two-stage detector. Each prediction goes through two distinct processing stages: first, the Region Proposal Network (RPN) identifies candidate object regions (Rois), and then a fully connected head processes these regions to refine the predictions for classification, bounding box regression, and mask generation. This overall architecture is shown in Figure 40.

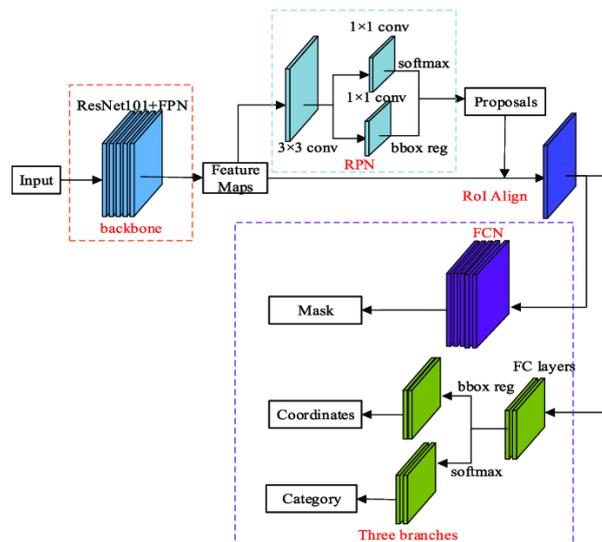


Figure 40. Mask R-CNN overall architecture

Backbone

There exist different options as backbone for Mask R-CNN, in our case the best performance was found for Residual Network (ResNet) 50, where its specific structure is shown in Figure 41. ResNet-50 has 50 layers, including convolutional layers, batch normalization layers, activation functions (ReLU), and fully connected layers. The key feature in ResNet-50 is the introduction of residual blocks, which allow the network to learn residual functions with reference to the layer inputs, instead of learning unreferenced functions. This helps in training much deeper networks by making it easier to optimize. If a regular layer of the network is defined as $h_i = f_i(K_i * h_{i-1} + b_i) = F(h_{i-1})$, then the residual layer would be defined as $h_i = F(h_{i-1}) + h_{i-1}$.

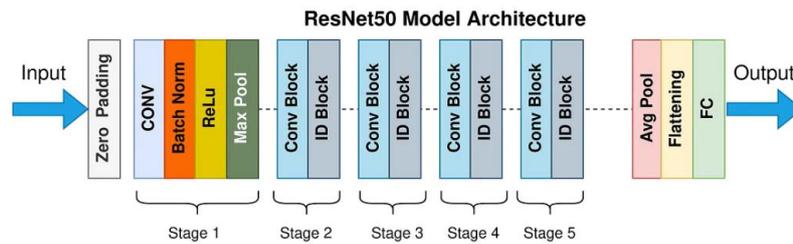


Figure 41. Architecture of ResNet-50

Region Proposal Network (RPN)

The RPN functions as a fully convolutional network that takes a feature map derived from the input image and outputs a set of rectangular object proposals, each accompanied by an objectness score.

- ❖ Initially, a 3x3 convolutional layer is applied to the input feature map, reducing its dimensionality while preserving spatial information. Subsequently, for each location in the resulting feature map, a predefined set of anchor boxes, spanning various sizes and aspect ratios, is generated.
- ❖ Following this, a 1x1 convolutional layer computes an objectness score for each anchor, indicating the likelihood of containing an object.
- ❖ Additionally, another 1x1 convolutional layer refines the coordinates of the anchor boxes to better align with the objects in the image.

This multi-step process enables the RPN to propose potential object regions with associated confidence scores, serving as an initial step in the object detection pipeline. The RPN is trained with a multi-task loss function that combines the classification loss and the regression loss given as

$$L_{RPN} = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

where p_i is the predicted probability of anchor i being an object, p_i^* is its ground-truth label. The classification loss is a Binary Cross Entropy loss

$$L_{cls}(p_i, p_i^*) = -p_i^* \log(p_i) - (1 - p_i^*) \log(1 - p_i)$$

And the regression loss is a smooth L1 loss

$$L_{\text{reg}}(t_i, t_i^*) = \sum_{j \in \{x, y, w, h\}} S_{L_1}(t_{ij} - t_{ij}^*)$$

Where $S_{L_1}(x) = \frac{1}{2}x^2$ if $|x| < 1$ and $S_{L_1}(x) = |x| - \frac{1}{2}$ otherwise. Also t_{ij} and t_{ij}^* are the predicted and the ground truth bounding box adjustment for the j -th coordinate of anchor i .

RoI Align and RoI Pooling

In Mask R-CNN, RoI Pooling and RoI Align are techniques used to extract fixed-size feature maps from RoIs. RoI Pooling divides an RoI into a grid of bins and applies max pooling within each bin to generate a fixed-size output, but this can cause misalignment due to quantization. RoI Align, on the other hand, addresses this issue by using bilinear interpolation to compute the exact values of the input features at four sampling points in each bin, preserving spatial alignment. This is particularly important for tasks requiring precise localization, such as mask prediction, as it maintains the accurate correspondence between the features and the RoIs, leading to better segmentation quality.

Fully Convolutional Network (FCN) and Fully Connected (FC) Layers

After RoI Align, the resulting feature maps are flattened into a 1-dimensional vector and passed through a series of fully connected layers. This process simultaneously computes the classification scores and bounding box coordinates.

The **classification loss** is calculated using cross-entropy, which measures the discrepancy between the predicted class probabilities p_i and the actual labels y_i . It is given as

$$L_{\text{cls}} = - \sum_i y_i \log(p_i)$$

The **bounding box regression loss** typically employs a smooth L1 loss, comparing the predicted t_i and true bounding box coordinates t_i^*

$$L_{\text{bbox}} = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i - t_i^*)$$

Additionally, the feature maps from RoI Align are fed into a series of convolutional layers within a Fully Convolutional Network (FCN), which are then upsampled to produce masks that match the original image size, preserving spatial information.

The **mask loss** is a pixel-wise binary cross-entropy loss that measures the difference between the predicted masks p and the ground truth masks y .

$$L_{\text{mask}} = - \sum_{i,j} (y_{ij} \log(p_{ij}) + (1 - y_{ij}) \log(1 - p_{ij}))$$

The **final loss** is the sum of all these losses

$$L = L_{\text{cls}} + L_{\text{bbox}} + L_{\text{mask}}$$

3.1.5 Contributions in terms of Optimization of the U-Net models

For both the Mask R-CNN and YOLOv8 models, we utilized their official loss functions because these functions are specifically designed to suit the models' architectures and achieve optimal performance. Exploring alternative loss functions could potentially disrupt this balance and degrade performance, so we have not pursued such alternatives at this time. However, we plan to further explore this topic to determine if we can find better alternatives for this specific application.

For U-Net models, the preferred loss functions to achieve more accurate results, as demonstrated in the Google Research competition, include the following or a combination of them: the Binary Cross-Entropy and Focal Losses (previously explained in the context of the YOLO model), the Dice Loss (defined as one minus the Dice score), and the **Lovász Hinge Dice Loss**.

This last loss function is a modification of the Lovász Hinge Loss, that is a loss function designed to directly optimize a submodular loss function, often the Intersection-over-Union (IoU) score. It combines the principles of hinge loss with the Lovász extension.

- ❖ **The Hinge loss** penalizes predictions that are not only incorrect but also those that are correct but with low confidence. It ensures that correct predictions with a margin greater than 1 incur no loss, while incorrect or low-confidence correct predictions incur a linear loss. For binary classification is given as:

$$l(y, f(x)) = \max(0, F(x))$$

Where $F(x) = 1 - y \cdot f(x)$, $y \in \{-1, 1\}$ is the ground truth label, and $f(x)$ is the predicted score

- ❖ **The Lovász extension** is a mathematical technique used to extend submodular set functions to real-valued functions. For instance, directly optimizing IoU is challenging because it is a discrete, non-differentiable function. To address this, we use surrogate loss functions that are differentiable and can be optimized using gradient-based methods. The Lovász extension helps in constructing such surrogate loss functions.

Formal definition:

Given a submodular loss function (such as IoU or Dice) $F: 2^V \rightarrow \mathbb{R}$ satisfying the diminishing return property (i. e., for every $A \subseteq B \subseteq V$ and $s \in V \setminus B: F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$, meaning the marginal gain decreases as the set grows larger) the Lovasz extension $\hat{F}: [0,1]^n \rightarrow \mathbb{R}$ extends this function from the discrete domain of the power set 2^V to a continuous domain $[0,1]^n$ where $n = |V|$ (in this case $|V|$ is the number of pixels in the image). Now, for each $x \in [0,1]^n$ given the permutation σ (sorting the elements of x in descending order) the Lovász extension is given as

$$\hat{F}(x) = \sum_{i=1}^n x_{\sigma(i)} \left(F(S_{\sigma(i)}) - F(S_{\sigma(i-1)}) \right)$$

where $S_{\sigma(i)} = \{\sigma(1), \dots, \sigma(i)\}$ and $S_{\sigma(0)} = \emptyset$.

The Lovász Hinge Loss: for each prediction i , compute the margin m_i as the difference between the ground truth value and the predicted score $m_i = y_i - p_i$ where y_i is the ground truth value and p_i is the predicted score. Then, the loss is given as

$$\mathcal{L}_{Lvs} = \frac{1}{N} \sum_{i=1}^N \hat{F}(m^{(i)})$$

where N is the number of classes.

The **Dice Lovasz Hinge Loss** is a specific case of this definition in which we choose the submodular set function F as the Dice loss given by

$$F=1 - F_D = 1 - \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Our main contribution has to do with the realization that this choice for F often leads to a los of false positive predictions, as shown in.

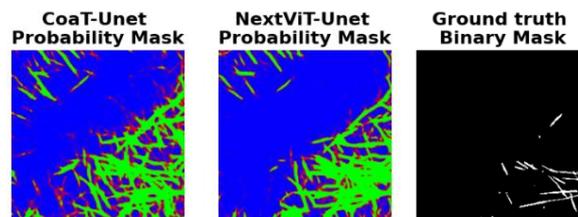


Figure 42. A comparison of the predicted probability masks by the U-Net models when using Dice Lovasz Hinge Loss and the ground truth mask.

Therefore, we chose to optimize the F-Beta Lovasz Hinge Loss with a very low beta parameter in order to yield more importance to avoid the generation of false positives.

The **F-Beta Lovasz Hinge Loss** is a specific case of this loss function in which the submodular function would be the F-Beta loss given as

$$F = 1 - F_{\beta} = 1 - \frac{(1 + \beta^2) \cdot TP}{(1 + \beta^2) \cdot TP + \beta^2 \cdot FP + FN}$$

3.1.6 Transfer Learning

Using **pretrained backbones**, particularly those trained on extensive datasets, is a highly effective practice in deep learning. A model backbone consists of layers that extract features from input data. It benefits greatly from pretraining as it enables the network to learn a wide array of general features, which extent from basic edges to complex object structures, leading to improved performance even with limited data. Pretrained models also **reduce training time and computational requirements** since they **start with a solid foundation of visual understanding instead of being initialized with random weights and biases**. This approach ensures consistency, reliability, and enhanced generalization in model performance.



Figure 43. Example of randomly selected image samples from ImageNet

ImageNet [28], with its dataset of over 14 million labeled images across thousands of categories (see Figure 43), is an ideal option for pretraining due to its diversity, large-scale benchmarks, and strong community support, making it a cornerstone in advancing computer vision. For this work, **we have used pretrained weights on ImageNet to initialize the backbones of all the models that are presented here.**

3.1.7 Model Ensembling

Model ensembling is a powerful technique in machine learning that combines the predictions of multiple models to improve overall performance. The key reasons why ensembling is beneficial include:

- ✓ **Reduction of Variance:** Individual models may be sensitive to the specific data they are trained on, which can lead to high variance in predictions. By ensembling the predictions of multiple models, the overall variance can be reduced.
- ✓ **Reduction of Bias:** Different models may capture different aspects of the data. By combining models, the ensemble can potentially reduce the overall bias compared to a single model.
- ✓ **Improved Generalization:** Ensembles tend to generalize better to unseen data compared to individual models. This is because they aggregate the strengths of various models, which can lead to better performance on new data.
- ✓ **Handling Overfitting:** While a single model may overfit the training data, an ensemble of models is less likely to overfit because it combines predictions from models that may have different overfitting behaviours.

In the context of our work, ensembling is particularly advantageous because we have been training multiple models, each with distinct capabilities. By combining these diverse models, we can leverage their unique strengths, leading to improved overall performance. Many alternatives exist to create an ensemble on top of different trained models, but we have explored mainly two of them:

Weighted Ensemble

We aim to find the optimal weights that produce the most effective combination of probability masks generated by multiple models. The objective is to maximize a specific evaluation metric, such as the Dice Score or the F1-Score, through black box optimization techniques.

Learning Ensemble

This approach involves training multiple diverse models, known as base learners, on the same training dataset. The predictions from these base learners are then used as input features for a higher-level meta-model, which makes the final prediction. The meta-model is designed to optimally combine the predictions from the base learners.

*This method shows great promise, as it has the potential to significantly improve the performance compared to using individual models or simple weight-averaging. However, we are still in the process of deploying this strategy for our application and **have not yet completed its implementation**. Therefore, this remains an area for future development.*

3.1.8 Post-processing Techniques

When applying the models to detect features in images outside the training dataset, several significant issues have been identified in the models' errors:

- *Lack of temporal consistency*
- *Misidentification of dark cloud edges as contrails*
- *Mistaken detection of some linear structures in the background as contrails*
- *Failure to detect thin linear clouds*

To mitigate these errors, certain postprocessing techniques have proven effective and will be elaborated upon below.

Temporal Smoothing

To ensure temporal consistency along the detections made in an image sequence, since each frame is processed separately by the models we are using at the time, we have decided to make use of optical flow techniques to find cloud motion vectors. The concept of optical flow refers to the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (camera) and the scene. It provides a way to track the movement of pixels between consecutive frames of a video or image sequence. The concept relies on several key mathematical principles and assumptions.

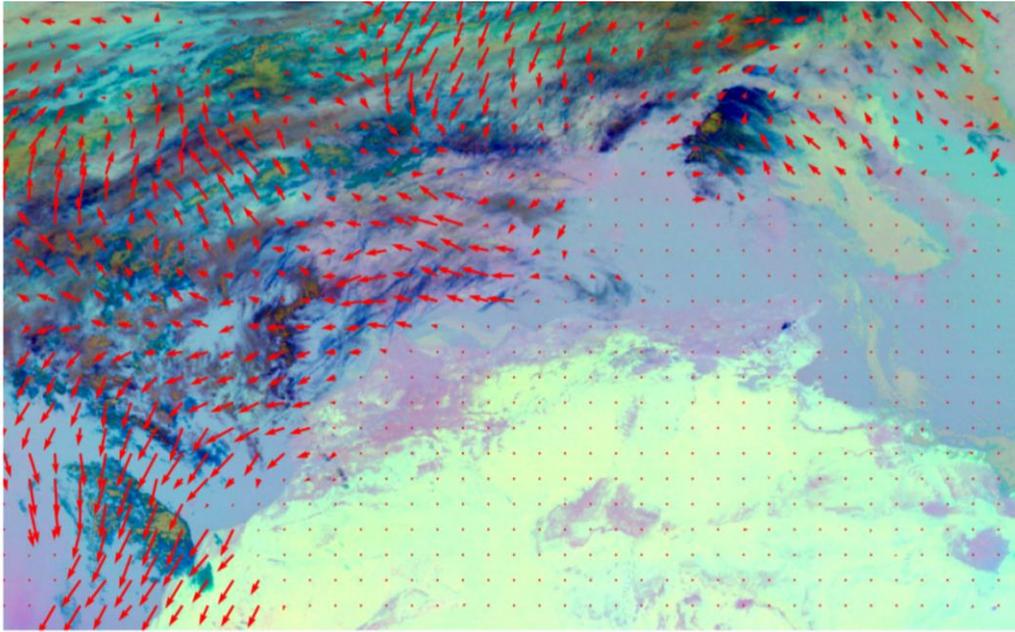


Figure 44. The optical flow field computed between two consecutive MSG/SEVIRI images

The optical flow equation given an image I is formulated as

$$u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0$$

Where $u = \frac{\partial x}{\partial t}$ and $v = \frac{\partial y}{\partial t}$. To get to this equation, some assumptions need to be made:

- ❖ **Brightness constancy assumption:** This assumption states that the intensity of a particular point in the image remains constant between consecutive frames. Thus, for a pixel at location (x, y) and time t , denoted as $I(x, y, t)$, its intensity at the subsequent time step, $t + \delta t$, remains the same at the location $(x + u, y + v)$, where u and v represent the optical flow vectors. Hence,

$$I(x, y, t) = I(x + u, y + v, t + \delta t)$$

- ❖ **Ignoring higher order terms in Taylor expansion:** Using Taylor expansion we get that

$$I(x + u, y + v, t + \delta t) = I(x, y, t) + u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} + O(u^2, v^2, \delta t^2, uv, u\delta t, v\delta t)$$

By ignoring higher order terms and substituting the brightness constancy assumption we get the optical flow formula.

When dealing with small patches of an image, the **aperture problem** can lead to inaccurate or incomplete motion estimates because the motion detected within each small patch (aperture) is ambiguous. To overcome the aperture problem, additional constraints and assumptions are necessary to obtain a unique solution for the motion vector. The **Lucas-Kanade method** [29] is a widely used approach that addresses this problem. Instead of solving the above equation for each pixel, we consider a **small window of n pixels** around the point of interest. This gives us a system of equations for all pixels within the window:

$$\begin{bmatrix} I_x(x_1, y_1) & I_x(x_1, y_n) \\ \cdots & \cdots \\ I_x(x_n, y_1) & I_x(x_n, y_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(x_1, y_1) \\ \cdots \\ I_t(x_n, y_n) \end{bmatrix}$$

That, in compact form, can be written as $Av = b$ and be approximated using the **least squares solution** $v = (A^T A)^{-1} A^T b$.

Once we calculate optical flow vectors for the entire image using methods like the Lucas-Kanade method, we can anticipate the movement of objects or features in subsequent frames by applying these vectors to the current detections or masks. This **propagation process** allows us to predict where objects or features will be in the following frames based on the direction and magnitude of the flow vectors. Integrating this propagated information with the detections or masks obtained from the current frame ensures consistency and accuracy in object tracking or feature detection across multiple frames of the video, enhancing the overall reliability of the system.

Reduction of False Positive detections

First, we aim to filter out regions that have been mistakenly detected as contrails but are actually **too large to be contrails**. These could be features such as islands or other elements that sometimes extend hundreds of kilometres, achieving sizes sometimes as big as an entire country. Here, in Figure 45, we demonstrate the sizes of detected contrails for a given MSG/SEVIRI scene with the aid of a histogram.

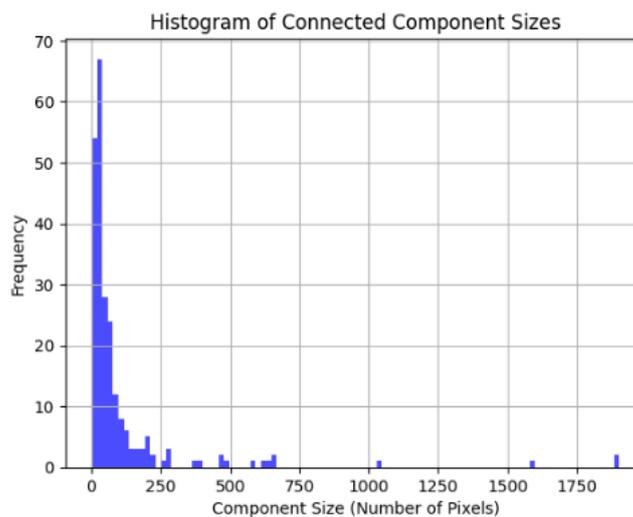


Figure 45. Histogram of the sizes of contrails detected by a model

In order to filter out these large features we threshold those detections that deviate too much from the mean of the detection sizes.

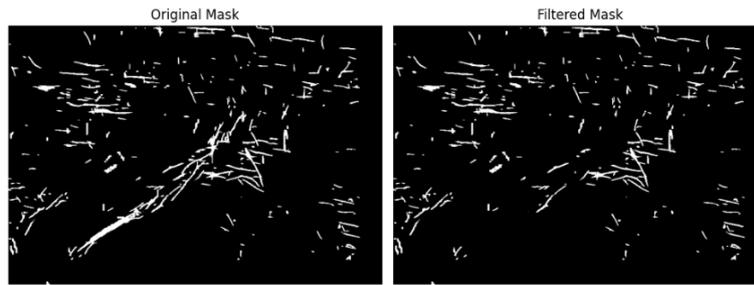


Figure 46. Example of filtering of large detected features

In the next filtering step, we process the image by separating it into Hue-Saturation-Value (HSV) channels (see Figure 47). This colour space helps us better distinguish the features of interest.

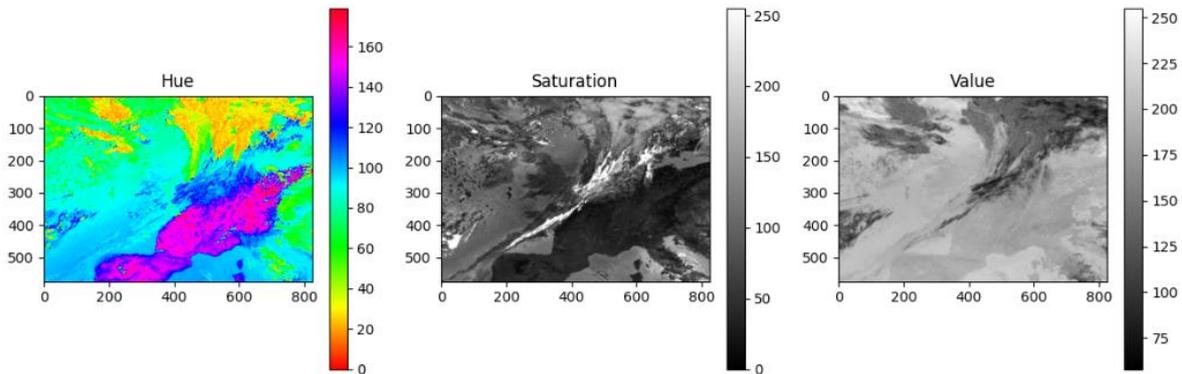


Figure 47. Decomposition of the Ash-RGB product into H,S,V channels.

Since we know that in this false-colour RGB representation, contrails always appear darker than the background, we can effectively **filter out some linear artifacts**. By applying a threshold to the Value channel, we can exclude regions that are too light, as they are unlikely to represent pixels belonging to a contrail. This helps in eliminating linear shaped structures detected on the ground or at the sea.

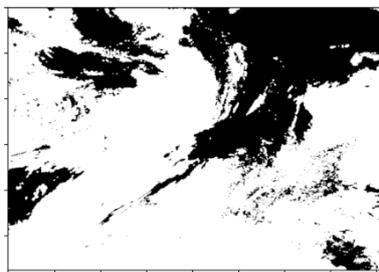


Figure 48. Example mask for filtering out lighter value pixels

The result of applying these two filters to the model detections would look something like shown in Figure 49. Although with this method we achieve a good reduction in the false positive detections by filtering out non-cloud pixels and non-realistic contrail sized features, ***we still need to further explore alternative techniques to achieve cleaner and finer final predicted masks.***

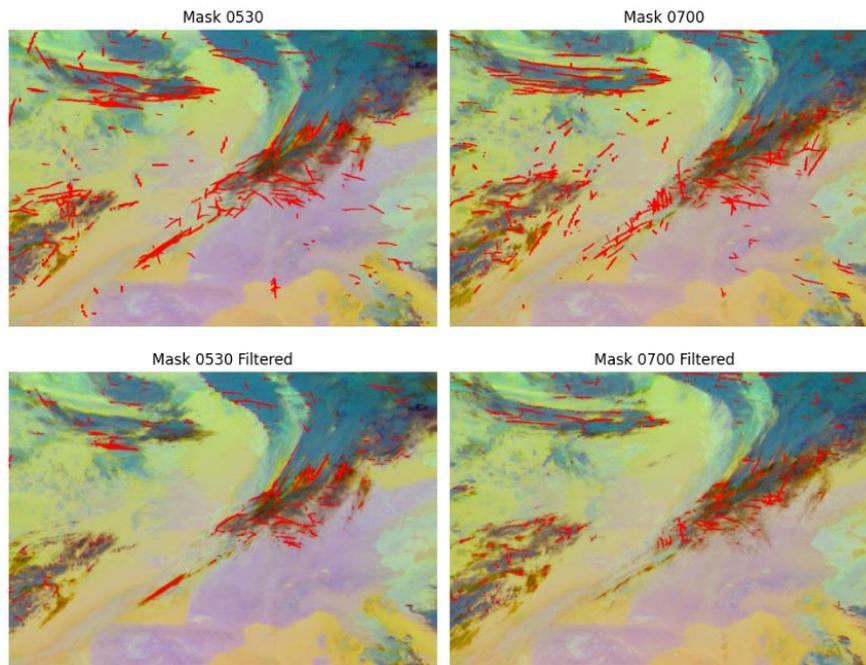


Figure 49. The result of applying the two filters to the model detections on two different MSG/SEVIRI scenes from the same day.

One preliminary idea for filtering out cloud edges, which has not yet been tested, involves using spatial image gradients. This can be achieved by applying the Sobel operator (for example) to compute the gradient magnitudes and directions, highlighting significant changes in intensity (see Figure 50). By analysing the pixels along the contours of detected features, we can determine edge characteristics: if a detected linear structure has high gradient magnitudes on one side of its contour but low magnitudes on the other, it is likely an edge of a cloud rather than a contrail.

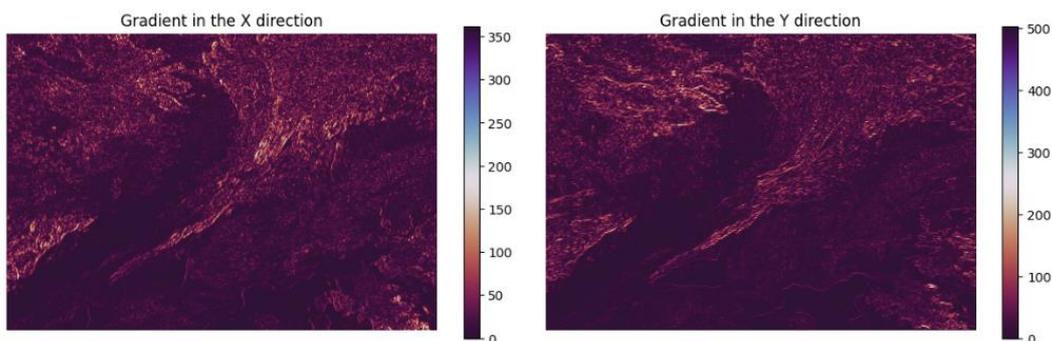


Figure 50. Computation of the image gradients using Sobel operator

Reduction of False Negative detections

The model's difficulty in detecting contrails that are too thin or lack sufficient contrast with the background is primarily due to the low resolution of the MSG/SEVIRI images used for testing, where each pixel corresponds to a 3km x 3km area. Consequently, fine details are lost, making it challenging to identify narrow or low-contrast contrails. However, ***this issue is expected to be mitigated at least partially with the availability of MTG imagery***, which offers higher resolution and thus should enhance the model's ability to detect these contrails more accurately.

3.2 Physics-Driven Contrail Simulation Model

This section provides details into the physics-driven simulation process for contrails.

3.2.1 Schmidt-Appleman criteria

Conditions for the formation of aircraft contrails are relatively well understood, primarily governed by atmospheric thermodynamics as per the Schmidt-Appleman criterion [30] [31] [32]. Among various factors contributing to contrail formation, the predominant cause is a thermodynamic phenomenon known as "heterogeneous nucleation". This occurs when water vapour emitted by engines condenses into supercooled liquid droplets, which subsequently freeze at a lower supersaturation level due to the addition of cloud-forming particles (soot) or cloud condensation nuclei. Contrails typically develop at high altitudes where the atmosphere is frost-saturated, meaning it is supersaturated with respect to ice but subsaturated with respect to water. Persistent contrails occur when aircraft traverse regions of airspace where ambient temperature and relative humidity, concerning both water and ice, meet specific criteria, expressed through the following inequalities:

$$\left\{ \begin{array}{l} RH_w > r_{cr.} \\ T < T_{cr.} \\ RH_i > 100\% \\ RH_w < 100\% \end{array} \right\} \quad (1)$$

where, RH_w denotes the ambient relative humidity with respect to water, and T is the ambient temperature. The relative humidity with respect to ice, RH_i can be derived from the following formula:

$$RH_w = RH_i \frac{6.0612 \exp(18.102 T / (249.15 + T))}{6.1162 \exp(22.5777 T / (273.78 + T))}$$

In addition, $T_{cr.}$ and $r_{cr.}$ are critical temperature and relative humidity respectively:

$$T_{cr.} = -46.46 + 9.43 \ln(G - 0.053) + 0.72 \ln^2(G - 0.053) + 273.15$$

$$r_{cr.} = \frac{G(T - T_{cr.}) + e_{sat}^{liq}(T_{cr.})}{e_{sat}^{liq}(T)}$$

In above, the temperature in Celsius. Moreover, we have:

$$e_{sat}^{liq}(T) = 6.0612 \exp\left(\frac{18.102 T}{249.52 + T}\right)$$

$$G = \frac{EI_{H_2O} C_p P}{\epsilon Q (1 - \beta)}$$

In above, $EI_{H_2O} = 1.25$ is water vapour emission index, $Q = 46 \times 10^6 J/kg$ is combustion heat per unit mass of jet fuel. Moreover, $C_p = 1.004 \times 10^3 J/kg.K$ is heat capacity of air. $\epsilon = \frac{M_{H_2O}}{M_{air}}$ is the ratio of molecular masses and $\beta = 0.3$ is the average propulsion efficiency of the jet engine.

3.2.2 Contrail Cloud Microphysics Model

Contrail cloud ice particles expand by absorbing excess water vapour from the nearby atmosphere when temperatures are below 0°C. The process responsible for this growth through the diffusion of water vapour is termed deposition, a widely studied dynamic phenomenon in cloud microphysics [33]. This section provides a comprehensive overview of ice particle diffusional growth. It is important to note that the current model assumes spherical shapes for all ice particles across the entire contrail's lifespan [34].

Suppose that r (meter) is the single ice particle radius and D_v (m^2/s) is the diffusivity of water vapour at temperature between -40 and 40°C, defined as:

$$D_v = 0.211 \left(\frac{T}{T_0}\right)^{1.94} \left(\frac{P_0}{P}\right) 10^{-4}$$

In above, T (in Kelvin) denotes the reference temperature, where $T_0=273.15$ K represents the standard temperature, P (in Pascals) signifies the reference pressure, and $P_0=101325$ Pa denotes the standard pressure. Further let λ (in meters) represent the mean free path of air molecules, defined as:

$$\lambda = \frac{2\mu}{P \sqrt{\frac{8m_{air}}{\pi RT}}}$$

In the above definition, $\mu = 1.83 \times 10^{-5}$ is viscosity of air, $m_{air} = 0.018$ kg/mol represents the molecular mass of air and $R = 8.214$ J/K.mol is the universal ideal gas constant.

Next, the adjusted diffusivity of water vapour to account for kinetic correction is determined as follows:

$$D'_v = \frac{D_v}{\frac{r(t)}{r(t) + \Delta v} + \frac{D_v}{r(t)\alpha} \sqrt{\frac{2\pi M_w}{RT}}}$$

where $\alpha=0.7$ is deposition coefficient, $\Delta v= 1.3\lambda$ and $M_w = 0.018$ kg/mol is ratio of molecular mass of water and dry air.

We define the adjusted thermal conductivity of water vapour for kinetic correction as:

$$k'_a = \frac{k_a}{\frac{r(t)}{r(t) + \Delta_T} + \frac{k_a}{r(t)\alpha\rho_{air}C_p} \sqrt{\frac{2\pi M_w}{RT}}}$$

In above, $k_a = 0.025$ J.m/s is thermal conductivity of water vapor, $\Delta_T = 2.16 \times 10^{-7}$ m is thermal accommodation coefficient, ρ_{air} denotes air density, and $C_p = 1.004 \times 10^3$ stands for the specific heat of dry air.

We denote the saturation vapour pressure with respect to ice by:

$$e_{sat,i} = \exp \left(9.55 - \frac{5723.3}{T} + 3.53 \log(T) - 0.0073 T \right)$$

Moreover, the saturation vapour pressure with respect to water is expressed as:

$$e_{sat,w} = 611.2 \exp \left(\frac{17.62 (T - 273.15)}{243.12 + (T - 273.15)} \right)$$

Specific latent heat of sublimation L_s is expressed as

$$L_s = 10^3 (2834.1 - 0.29 T - 0.004 T^2)$$

Given the above quantities, we can now compute mass diffusional growth rate of a single ice particle by:

$$\dot{m}(t) = \frac{4\pi C S_{v,i}}{\frac{RT\rho_i}{e_{sat,i} D_v' M_w} + \frac{L_s \rho_i}{k_a' T} \left(\frac{L_s M_w}{RT} - 1 \right)} = f(r, P, T)$$

where $C = 1/4\pi$ is capacitance factor of a spherical ice particle, and saturation ratio of water vapour with respect to ice is $S_{v,i} = \frac{e_{sat,w}}{e_{sat,i}} - 1$.

Since the radius of a single spherical ice particle ($r(t)$) appears in the formula for $\dot{m}(t)$, we need to couple the computation of $r(t)$ to that of $\dot{m}(t)$. The formula for the radius growth of a single ice particle reads:

$$\dot{r}(t) = \frac{\dot{m}(t)}{4\pi r^2(t)}$$

3.2.3 Particle-Transport Model:

In this section, we present our methodology to track the persistent contrails over time. To this end, we introduce an Eulerian approach which accounts for a general advection-diffusion equation. It is assumed that (upon satisfaction of the persistent-contrail criteria), ice particles start to grow right after they are injected into the atmosphere. It is noteworthy that a more precise simulation requires that one also track the evolution of water vapour temperature. More specifically, the water vapour being injected from the aircraft engine is at around 600 °C; hence, the cooling-down process typically occurs at some certain distance from the engine, and this by itself mandates the solution of energy-transport equation. Nevertheless, since this distance is quite short (typically below a few hundreds of meters) the previous assumption regarding the growth of ice particle right after the injection seems reasonable.

The behaviour of ice-particles can be defined through the following coupling equations:

$$u = u_{jet} + \bar{u}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0$$

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u u) = -\nabla p + \nabla \cdot \tau + \rho g$$

$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = \nabla^2 (K\varphi) + S$$

In above, u represents the velocity of the injecting jet flow from the aircraft engine which is a summation of u_{jet} (representing the velocity of the injecting jet flow with respect to wind/background speed) and \bar{u} (the velocity of wind with respect to a ground observer). Moreover, ρ represents the density of air, t represents time, τ denotes viscous stress tensor, φ represents the concentration of ice-particles, K is the diffusivity coefficient and S is to account for any possible source terms. It is noteworthy that it is also assumed that u is the same as the velocity of resisting air.

From the above equations, it is evident that the equation representing the concentration of ice particles is coupled to the jet equations through u_{jet} . Specifically, u_{jet} is significant near the engine, i.e., $u_{jet} \gg \bar{u}$ and soon after, u_{jet} decays such that the transport of ice particles is dictated through \bar{u} . In other words, there is a distance after the engine at which $\bar{u} \gg u_{jet}$. Therefore, it seems reasonable to focus on the long-term propagation of ice particles through wind. Nonetheless, understanding the initial concentration of ice particles necessitates a careful resolution of the jet phenomenon, leading to a better understanding of the downwash process.

As pointed out earlier, the downwash process dominates over only one minute after the injection whilst, our target is to model the propagation of ice particles over a wide area, extending to hours. Therefore, our current model does not account for the solution of NS equations. It is also noteworthy that beside a careful solution to the above coupled system of equations is computationally expensive, it entails many computational and theoretical challenges, and it is, by itself, an active area of research.

Therefore, the focus of the present model is on the resolution of ice-particles concentration through the advection-diffusion equation where the transport phenomenon is dictated by wind. However, in order to (naively) account for the jet velocity, we assume (through the available analytical solution for the incompressible radial jet flow) that its decaying velocity is proportional to the inverse of the distance from the engine location.

Having considered the above discussion, we express our model by expanding the advection-diffusion equation as:

$$\frac{\partial \varphi}{\partial t} + (w_x + u_{x,jet}) \frac{\partial \varphi}{\partial x} + (w_y + u_{y,jet}) \frac{\partial \varphi}{\partial y} + (w_z + u_{z,jet} + u_s) \frac{\partial \varphi}{\partial z} = k_x \frac{\partial^2 \varphi}{\partial x^2} + k_y \frac{\partial^2 \varphi}{\partial y^2} + k_z \frac{\partial^2 \varphi}{\partial z^2} + S$$

In above, w_x, w_y and $w_z \ll 1$ are wind components. We assume that persistent contrails are formed during the cruise phase of the flight. Therefore, $u_{z,jet} = 0$. Moreover, the induced jet velocity is computed by:

$$u_{x,jet} = \frac{V_{exit} \cos(\theta)}{r}, \quad u_{y,jet} = \frac{V_{exit} \sin(\theta)}{r}, \quad r = \|X - X_{aircraft}\|.$$

where $X:=(x,y,z)$.

Moreover, u_s is the settling velocity, defined as [35]:

$$u_s = f_s \frac{(\rho_i - \rho_{air}) d_p^2}{18\mu} g \sin\left(\text{atan}\left(\frac{w_y}{w_x}\right)\right).$$

where d_p represents the diameter of ice particles.

The order of magnitude for $u_s \ll 1$, makes it possible to ignore the advection in the 'z' direction. Moreover, for the sake of the computational speed, we further disregarded the diffusion term in 'z' direction to construct a 2D model. It is noteworthy that the propagation of ice-particles in 'z' direction may be tracked separately either through a Lagrangian approach or empirical estimations.

The main term in the above advection-diffusion equation is the source term (S). More specifically, this term includes ice-particle growth (which is the main term contributing to persistent contrails), as well as some slip mechanisms such as inertia, magnus and thermophoresis effects [36]. Therefore, we can write:

$$S = S_i + S_t + S_m + S_{growth}$$

In above, $S_i \propto \pi d_p \mu ||u||$ is the inertia due to the Stokes friction, $S_m \propto \nabla u$ is the slip mechanism due to the magnus effect, $S_t = D_t \nabla \varphi \frac{\nabla T}{T}$ is the slip mechanism due to thermophoresis (soret) effect, and $S_{growth} = \frac{f(r,P,T)N_p}{\Delta x \Delta y (1)}$ (N_p being the number of ice particles after the downwash process [34]) is the main source term accounting for the growth of ice-particles leading to persistent/growing particles.

Comment 1: upon performing an order of magnitude analysis [36], it becomes apparent that the slip mechanism due to inertia is negligible for $d_p < 10^{-6}m$. This term can become significant in long terms. Our simulations show that upon setting the initial ice-particle diameter as $d_{p,i} = 10^{-6}m$ ([34]), a persistent growth after about 2 hours of exposure to $T=-15^{\circ}C$ (which gives the maximum growth rate) leads to a final ice-particle diameter of about $d_{p,f} = 5 \times 10^{-6}m$. In other words, the source term due to the inertia is negligible for many scenarios.

Comment 2: Magnus and thermophoresis effects can contribute to the propagation of ice particles. Nevertheless, at this stage, it appears that the experimental data determining the diffusion rates due to these terms is not available.

3.2.4 Numerical Solution to Particle-Transport Model:

Since the considered advection-diffusion equation is time-dependent encompassing source terms and variable velocity fields, deriving an analytic solution seems beyond the state of the art on this topic. Therefore, at this point, numerical solutions are more appropriate.

We solve the advection-diffusion equation employing Alternating Directional Implicit (ADI) which is proved to be unconditionally-stable (not relying on CFL condition) for linear transport equations. Furthermore, ADI approach provides an accurate temporal resolution.

Here, we present an overview of the ADI approach.

First Fractional Step:

- Treating the x-direction implicitly and the y-direction explicitly:

$$\frac{\varphi_{i,j}^{n+1/2} - \varphi_{i,j}^n}{\Delta t/2} + (w_x + u_{x,jet}) \frac{\partial \varphi_{i,j}^{n+1/2}}{\partial x} + (w_y + u_{y,jet}) \frac{\partial \varphi_{i,j}^n}{\partial y} = k_x \frac{\partial^2 \varphi_{i,j}^{n+1/2}}{\partial x^2} + k_y \frac{\partial^2 \varphi_{i,j}^n}{\partial y^2} + S_{i,j}^n$$

Second Fractional Step:

- Treating the y-direction implicitly and the x-direction explicitly:

$$\bullet \quad \frac{\varphi_{i,j}^{n+1} - \varphi_{i,j}^{n+1/2}}{\Delta t/2} + (w_x + u_{x,jet}) \frac{\partial \varphi_{i,j}^{n+1/2}}{\partial x} + (w_y + u_{y,jet}) \frac{\partial \varphi_{i,j}^{n+1}}{\partial y} = k_x \frac{\partial^2 \varphi_{i,j}^{n+1/2}}{\partial x^2} + k_y \frac{\partial^2 \varphi_{i,j}^{n+1}}{\partial y^2} + S_{i,j}^{n+1/2}$$

We use a central difference scheme for the second derivatives and forward (upwind) schemes for the first-order derivatives and the discrete problem is reduced to a 2-step triangular system which is solved using Tridiagonal matrix algorithm (Thomas/TDMA) [37].

Estimating Wind, Temperature, and Relative Humidity from Reanalysis Data:

The available data from Reanalysis does not have adequate resolution for particle-tracking algorithm. Therefore, before incorporation, wind field, together with temperature and relative humidity should be reconstructed for the considered domain.

To this end, we adopt a neural network model to reconstruct the above quantities. Each quantity is modelled by a simple 2-layer fully-connected network as follows:

$$q(X) = W_{q,2} f(W_{q,1} X + b_q)$$

In above, q denotes the modelled quantity, $X := (x, y, z, t)$ is the input vector, $W_{q,1} \in n \times 4$, $b_q \in n \times 1$ are weights and biases of the neural network and $W_{q,2} \in 1 \times n$ is the scaling vector. Moreover we choose the following activation function which is more suited for neural approximations:

$$f(p) = \begin{cases} p^2 & p > 0 \\ -p^2 & p < 0 \end{cases}$$

Comment 3: In order to model wind consistent manner, one needs to ensure that the wind components adhere to NS equations. This is because wind, by itself, can be considered as an incompressible flow. Such simulations involving some equations (to be satisfied) are typically referred to as Physics-Informed Neural Networks (PINN). However, due to the computational cost of a complete PINN simulation, the current model only accounts for the continuity equation. Therefore, through backpropagation, we first compute the derivatives of wind components and extend the loss function (which is an Euclidian norm, measuring the difference between the available data and neural output) by a term representing minimization of $|\frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y}|$ over the entire domain.

Aircraft Repositioning

Since aircraft is cruising constantly, its position needs to be updated at each time step through the Kinematic flight dynamics:

$$\frac{dx_{aircraft}}{dt} = V_{aircraft} \cos \theta + w_x(x, y, z, t)$$

$$\frac{dy_{aircraft}}{dt} = V_{aircraft} \sin \theta + w_y(x, y, z, t)$$

The above equations are solved using R-K 4th -5th order.

4 Results

4.1 Data-Driven Contrail Detection Models

In this section we will introduce the metrics used for evaluating the performance of the different models and the quantitative results obtained based on this metrics. We also provide some qualitative results in which the contrails detected by the images are examined and a discussion of the qualitative performance and in which situations the models perform well

4.1.1 Metrics used for evaluation

Segmentation Metrics

To evaluate the performance of the models in terms of semantic segmentation, specifically for the task of classifying each pixel as either a contrail or not a contrail. These metrics help us understand how well the models are performing the segmentation task. The most relevant metrics we focus on are precision and recall. Additionally, we provide the F1 score and Dice score, which evaluate the balance between precision and recall. Here is a more detailed explanation of each of them:

- **Precision:** It is defined as the ratio of true positive (TP) predictions to the total number of positive predictions (true positives and false positives, FP). The formula is given as

$$\text{Precision} = \frac{TP}{TP + FP}$$

In the context of our problem, it quantifies the proportion of pixels predicted by the model to be part of a contrail that actually belong to a true contrail.

- **Recall:** Also known as sensitivity or true positive rate, measures the ability of the model to identify all the existing pixels belonging to instances. It is defined as the ratio of true positive predictions to the total number of actual positives (true positives and false negatives, FN). The formula is given by

$$\text{Recall} = \frac{TP}{TP + FN}$$

In the context of our problem, it represents the proportion of contrail pixels that have been correctly detected by the model out of all the pixels that actually belong to a true contrail.

- **F1-Score:** It is the harmonic mean of precision and recall. It provides a single metric that balances the trade-off between precision and recall, especially when the two values are not equal. A high F1 score indicates that both the precision and recall are reasonably high. It can be formulated as

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Dice Score:** It is defined as twice the area of overlap between the predicted and true regions divided by the total number of pixels in both the predicted and true regions. The Dice score ranges from 0 to 1, where 1 indicates perfect overlap. The formula is

$$\text{Dice Score} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

The Dice score *is often interpreted in terms of spatial overlap between segments*, making it more intuitive for tasks where the goal is to measure how well regions in images match. The F1 score, while mathematically similar, is interpreted more broadly in terms of balancing precision and recall.

Keypoint Detection Metrics

To assess the effectiveness of the Keypoint Detection models, our evaluation will primarily focus on precision, recall, and F1 scores, which are described similarly to the segmentation metrics. However, in this case, we measure the agreement between the predicted keypoints and the ground truth keypoints, rather than considering all the pixels defining each contrail.

In this context, the most indicative metric is likely the mean Average Precision (mAP). This metric quantifies the precision-recall trade-off across all thresholds and provides a comprehensive measure of model performance. Specifically, mAP calculates the average precision across all classes, providing a single value that summarizes the model's ability to accurately detect keypoints across the dataset. This metric is formulated as follows:

- **Mean Average Precision:** The mAP for keypoint detection is computed as the mean of the Average Precision (AP) scores for each keypoint. It is given as

$$\text{mAP} = \frac{1}{K} \sum_{k=1}^K \text{AP}_k$$

where K is the total number of keypoints, and AP_k is the Average Precision for the k -th keypoint. To calculate AP_k , we integrate the precision-recall curve for each keypoint:

$$\text{AP}_k = \int_0^1 P_k(R) dR$$

4.1.2 Model Qualitative Results

Here we are going to present some visual results obtained when making predictions on images of the validation set with the different models to get a better understanding of their performance.

Keypoint Detection Results

In this section, we present the visual results obtained using the best-performing keypoint detector, specifically the YOLOv8-pose model with 2-keypoint labels. The results, both those shown here and those examined separately, indicate that the model demonstrates high precision, meaning that the **detected objects are very likely to be contrails**. However, the recall is comparatively lower, indicating that the **model fails to detect many contrails**.

Here in Figure 51 and Figure 52 we can see a comparison between the ground truth labels of 4 different frames within the validation set, along with the predicted labels and the confidence score of each of them.

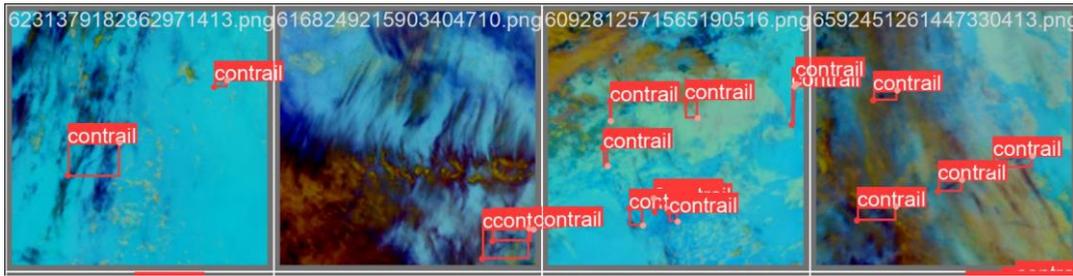


Figure 51. . Ground truth contrail labels with 2 keypoints

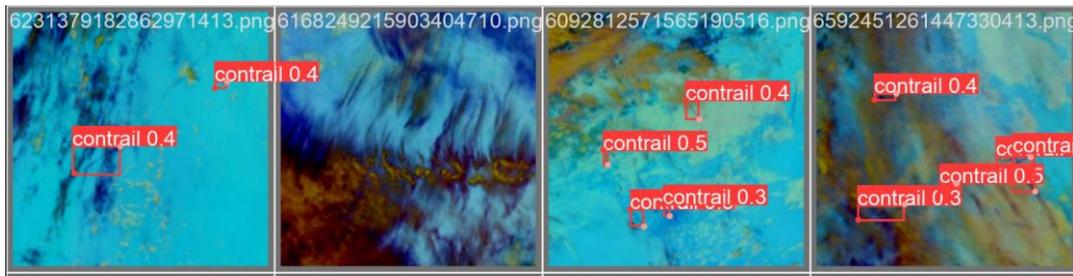


Figure 52. Contrails detected with YOLOv8-pose

Segmentation Results:

Here, we present several examples of the predictions made by each model and compare them with the ground truth results. Although the actual output of each model is a probability map, we have thresholded these for clearer visual interpretation.

Overall, *U-Net models produce the most accurate predicted masks*. For instance, see the following figures:

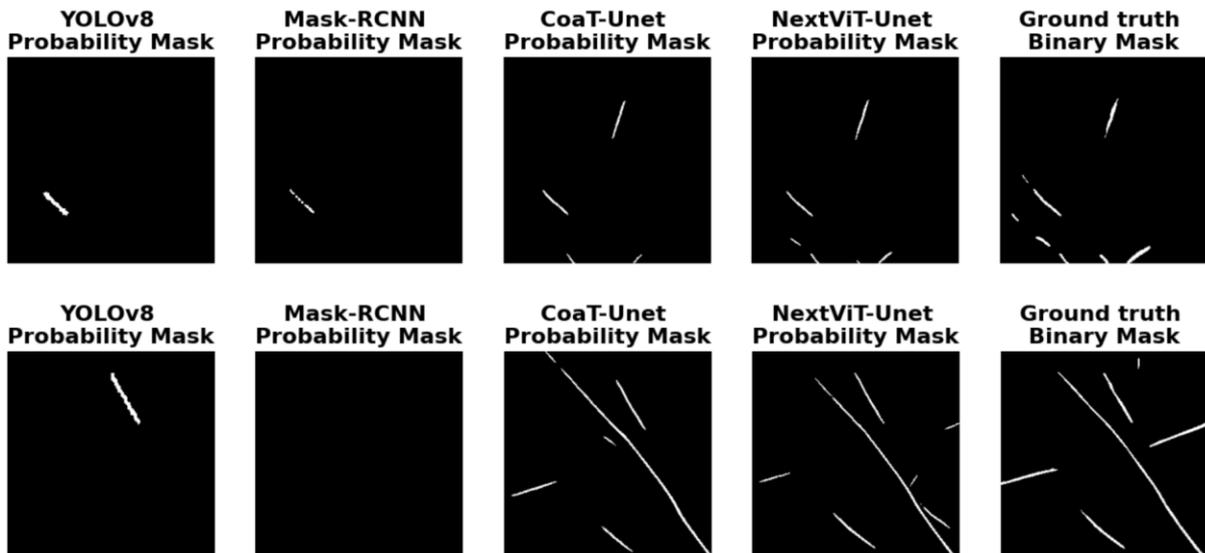


Figure 53. Masks generated by the different models and compared against the ground truth mask. Here are shown examples in which U-Net models outperform the rest.

Although they often perform better in many situations, it appears that when the contrails are clearly visible in the image, *all models are able to segment them accurately*, as shown in the following figures.

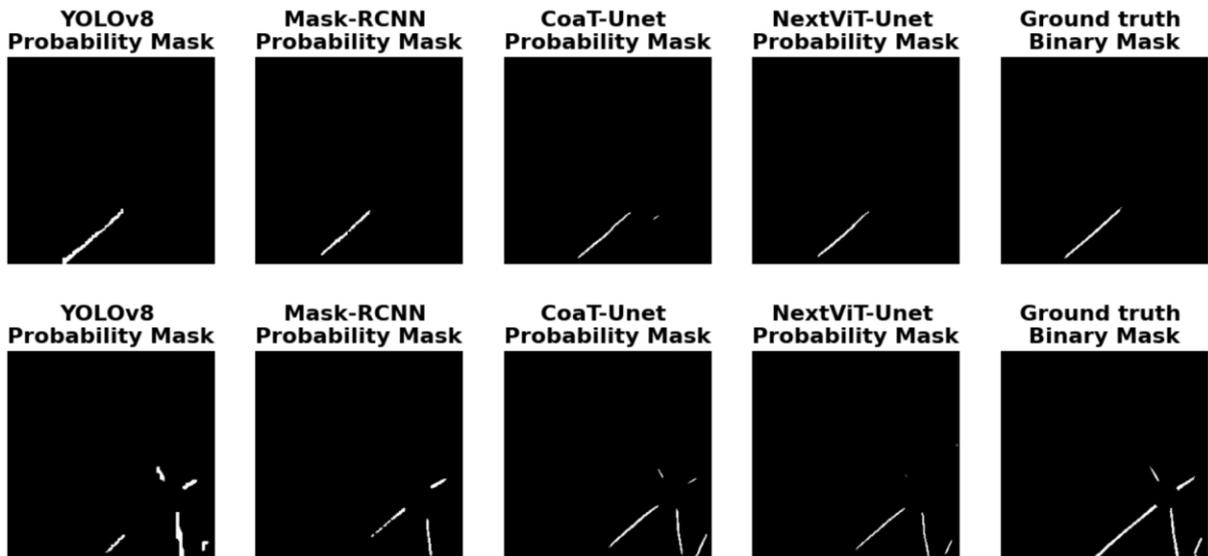


Figure 54. Masks generated by the different models and compared against the ground truth mask. Here are shown examples in which all of the models have a good performance.

However, we cannot conclude that only U-Net models should be considered for predictions. Although the other models generally don't perform as good, *there are scenes in which Mask R-CNN or YOLO outperform the U-Net models*, as shown in the following figures.

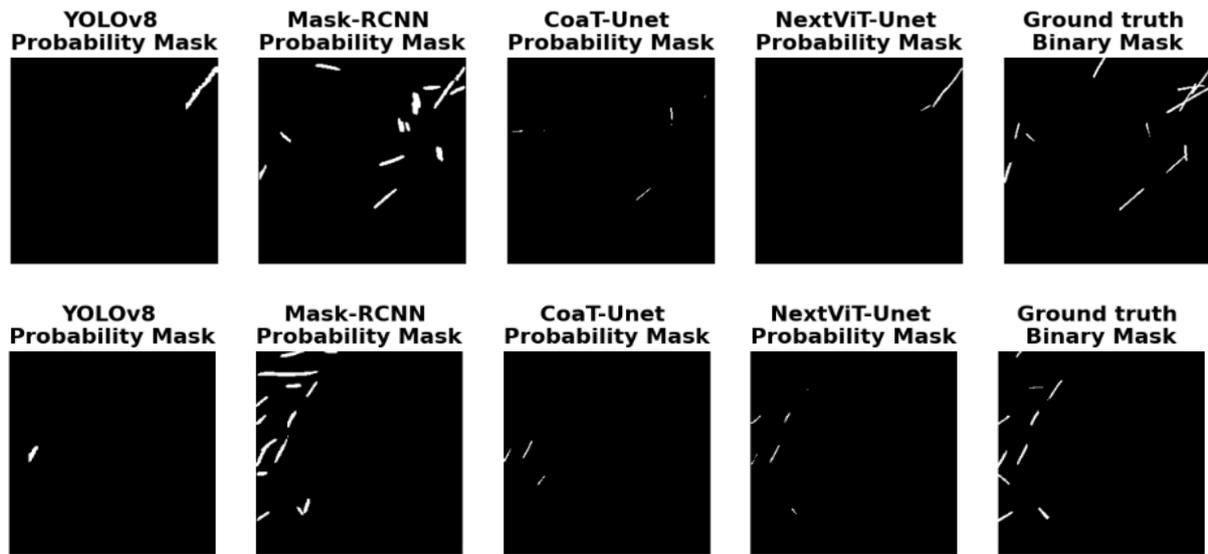


Figure 55: Masks generated by the different models and compared against the ground truth mask. Here are shown examples in which Mask-RCNN outperform the rest of the models

This indicates the potential advantage of combining or ensembling the outputs of all models to achieve the best possible results.

4.1.3 Model Performance Results

KeyPoint Detection Results:

We trained *YOLOv8-pose* for keypoint detection on our dataset. The keypoint detection labels were generated using a methodology that converts binary masks into polygons, but excludes the final contouring steps, retaining only the detected segments.

We experimented with various numbers of keypoints to define each contrail segment. According to the metrics, the results appear suboptimal. However, visual inspection suggests the performance may not be as poor as indicated by the metrics, suggesting that an alternative evaluation method might be necessary.

The current results are as follows:

A) Contrails defined by two keypoints: Here we expose the results obtained when training the model with 2-keypoint labels as shown in Figure 56.

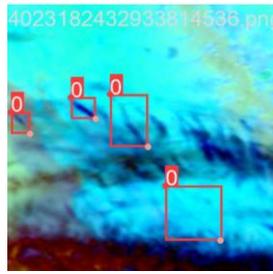


Figure 56: 2-keypoint contrail labels.

The precision-recall curve illustrates the trade-off between these two metrics. At a confidence threshold of 0.5, the model achieves its highest **mAP of 49.9%**. This indicates that, at this threshold, the model balances precision and recall optimally, capturing a significant portion of true positives while maintaining a reasonable level of accuracy in its predictions. The **F1-score** achieves its best performance of **53%** by setting the confidence threshold to a value of 0.186.

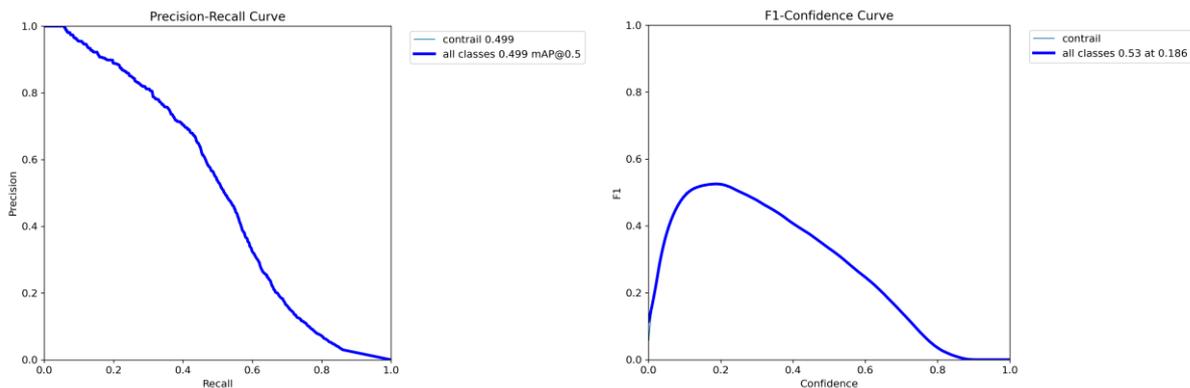


Figure 57: Precision-Recall Curve (left) and F1-score Curve (right) of YOLOv8-pose with 2 keypoint labels

The evolution of the loss functions and various metrics across epochs is depicted in. During training, the loss functions decrease monotonically, indicating that the model is effectively learning from the data. However, during validation, the loss functions exhibit a monotonic decrease only in the initial stages, suggesting that *the model's learning effectiveness diminishes over time*. The performance metrics, while generally increasing monotonically during training, tend to plateau and fluctuate around certain values, indicating that the model has reached its capacity to learn further and, in some cases, may even begin to overfit, leading to a decline in performance.

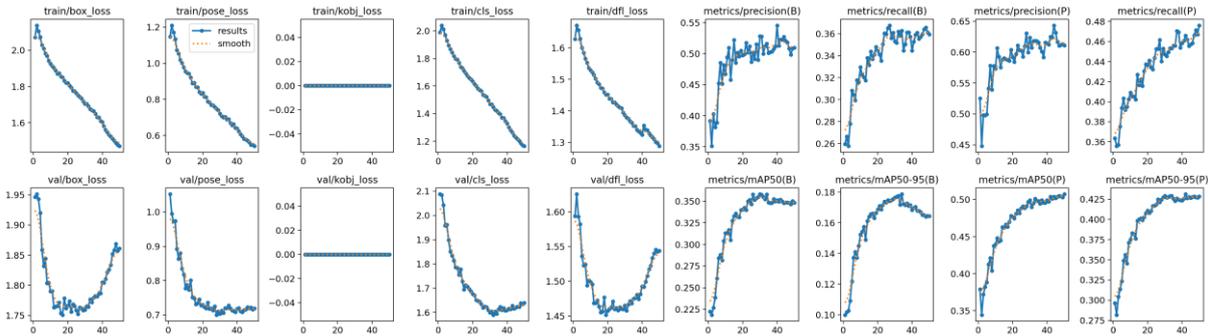


Figure 58: Evolution of the loss functions and the different metrics across epoch both in the training and validation set for YOLOv8-pose with 2 keypoints. Here B stands for Bounding Box results and P for Pose or Keypoint Detection results.

B) Contrails defined by three keypoints: Here we expose the results obtained when training the model with 3-keypoint labels as shown in Figure 59.



Figure 59: 3-keypoint contrail labels

The precision-recall curve indicates that at a confidence threshold of 0.5, the model achieves its highest **mAP of 46.6%**, which is 3.3% lower than the previously observed value. Additionally, *the F1-score* reaches its peak performance at **50%** when the confidence threshold is set to 0.174. This decline in metrics suggests that the model's performance is hindered by the increased complexity of predicting a greater number of keypoints, resulting in **lower overall accuracy**.

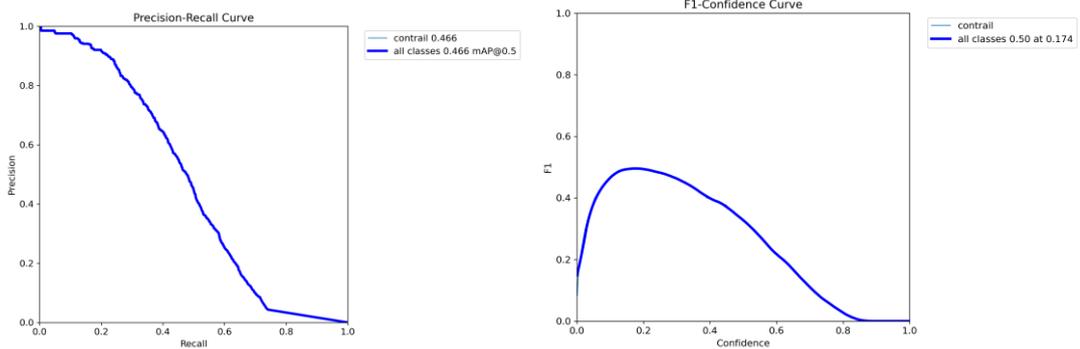


Figure 60: Precision-Recall Curve (left) and F1-score Curve (right) of YOLOv8-pose with 3 keypoint labels

During training, the loss functions decrease consistently, indicating effective learning. However, in validation, they only decrease initially, suggesting reduced learning effectiveness over time. Performance metrics fluctuate, generally increase, but eventually plateau and sometimes decline, indicating potential overfitting and the model's *limited generalization to validation data*.

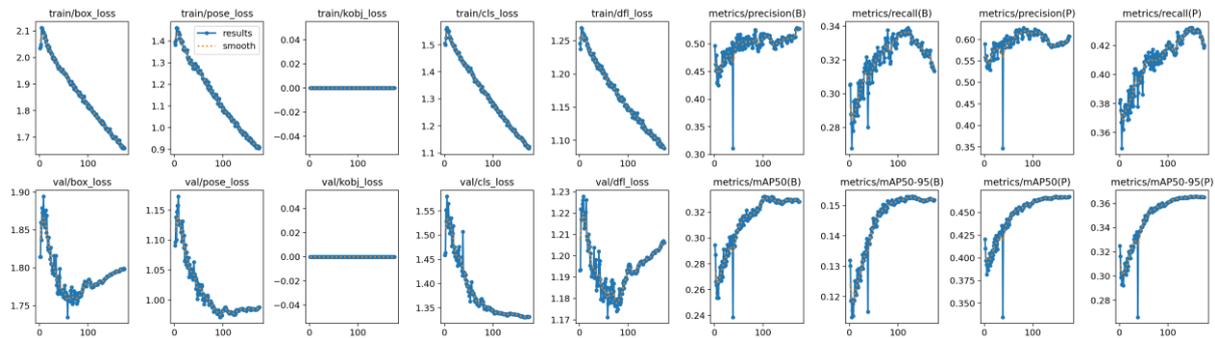


Figure 61: Evolution of the loss functions and the different metrics across epoch both in the training and validation set for YOLOv8-pose with 3 keypoints. Here B stands for Bounding Box results and P for Pose or Keypoint Detection results.

C) **Contrails defined by 20 keypoints:** Here we expose the results obtained when training the model with 20-keypoint labels as shown in Figure 62.



Figure 62: 20-keypoint contrail labels.

The precision-recall curve indicates that at a confidence threshold of 0.5, the model achieves its highest **mAP of 35.4%**, which is 11.2% lower than the previously observed value. Additionally, **the F1-score** reaches its peak performance at **44%** when the confidence threshold is set to 0.286. This demonstrates that increasing the number of keypoints leads to a decline in overall performance. **A potential reason for this phenomenon could be related to the methodology used to retrieve the segments that best**

describe each contrail. Given that there are multiple ways to select these segments, the model might produce different segment alternatives. As a result, when computing the metrics, the variability in segment predictions leads to significantly lower performance scores.

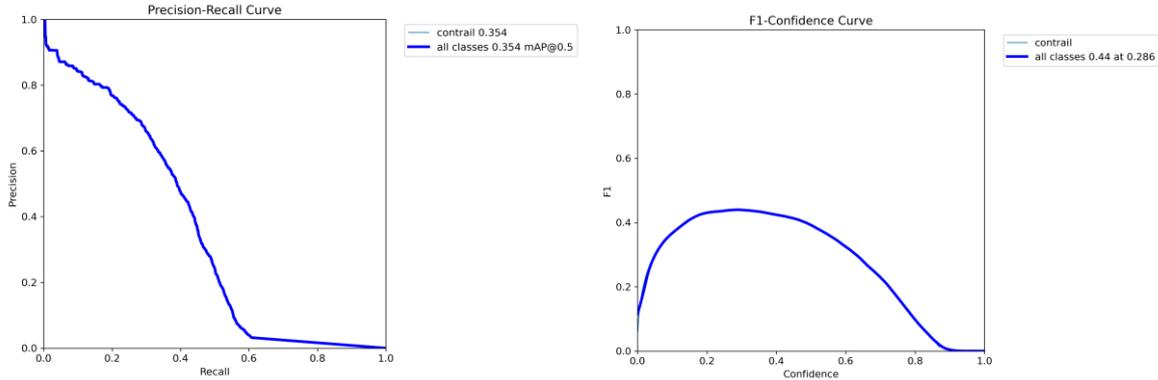


Figure 63: Precision-Recall Curve (left) and F1-score Curve (right) of YOLOv8-pose with 20 keypoint labels

The loss and metric functions generally follow similar patterns as observed in previous cases. However, we note **that the model ceased training after 40 epochs due to the early stopping criterion**, which was set at 20 epochs. This means the training halted because the model's performance did not improve during these 20 consecutive epochs and instead showed only fluctuations.

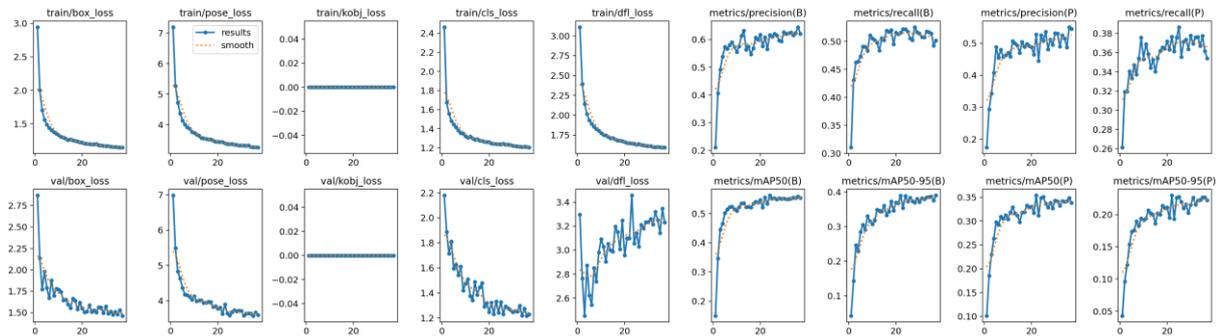


Figure 64. Evolution of the loss functions and the different metrics across epoch both in the training and validation set for YOLOv8-pose with 20 keypoints. Here B stands for Bounding Box results and P for Pose or Keypoint Detection results.

Segmentation Results:

Before **evaluating the performance of the YOLOv8 and Mask R-CNN models in terms of semantic segmentation**, it's important to note that they are not encoder-decoder networks. Unlike encoder-decoder networks that output a matrix with the same dimensions as the input image, this model uses a series of fully connected layers at the end. As a result, it outputs vectors representing the contours of detected contrail polygons along with the probability of each feature being a contrail. To evaluate segmentation performance, we need to create a contrail probability map from these contrail polygons. The way we do this is by following these steps:

1. Probability mapping: For each detected contrail polygon, we use the associated probability values to create a mask.

2. Instance mask retrieval: We plot each segmented contrail on a zero-valued mask, filling the entire segmented area with the corresponding probability values.
3. Combine Masks: We repeat this process for all detected contrail polygons and add all the individual masks together.
4. Clip Values: Finally, we clip the resulting added values to the 0-1 interval to ensure they represent valid probabilities.

This process allows us to generate a comprehensive contrail probability map from the model's outputs, facilitating the evaluation of its performance in semantic segmentation and allowing the comparison with other semantic segmentation models.

A) Performance of the CoaT U-Net model:

The results of the U-Net model trained with a CoaT backbone are shown in Figure 65.

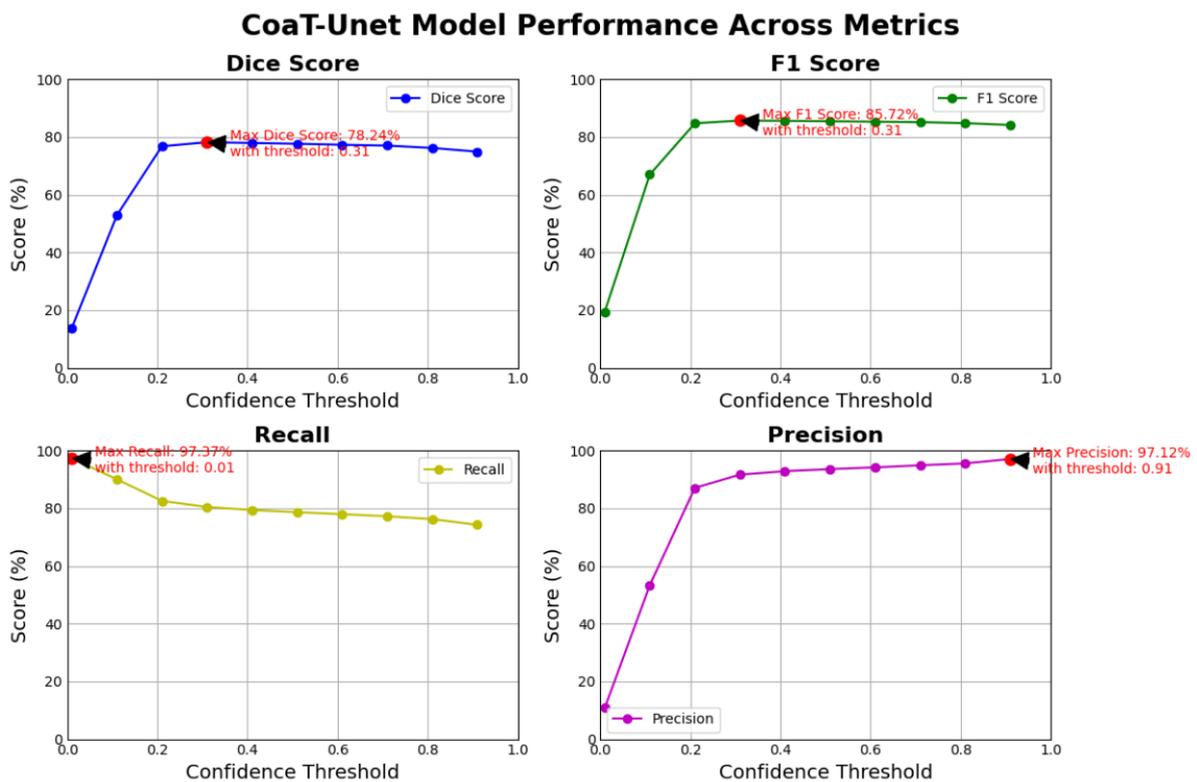


Figure 65. Metrics obtained with the CoaT-Unet model

The plots indicate that this model can achieve *near-perfect recall* by setting a low confidence threshold, which results in very low precision. This means the model can potentially detect all contrails in the scene but will also produce a significant number of false positives. However, by setting the confidence threshold to 0.31, a good balance between precision and recall is achieved, resulting in an **F1 Score of 85.72%** and a **Dice Score of 78.24%**. These dice score for a model trained on this data represents a strong performance, *surpassing state-of-the-art performance results*.

The **computational time** for training **20 epochs** was approximately **24 hours**, running on a single NVIDIA RTX 4090 GPU with 12GB of memory.

B) Performance of the NextViT U-Net model

The results obtained after training a U-Net model with NextViT backbone outperform the previous results obtained with the CoaT backbone network, as can be seen in Figure 66.

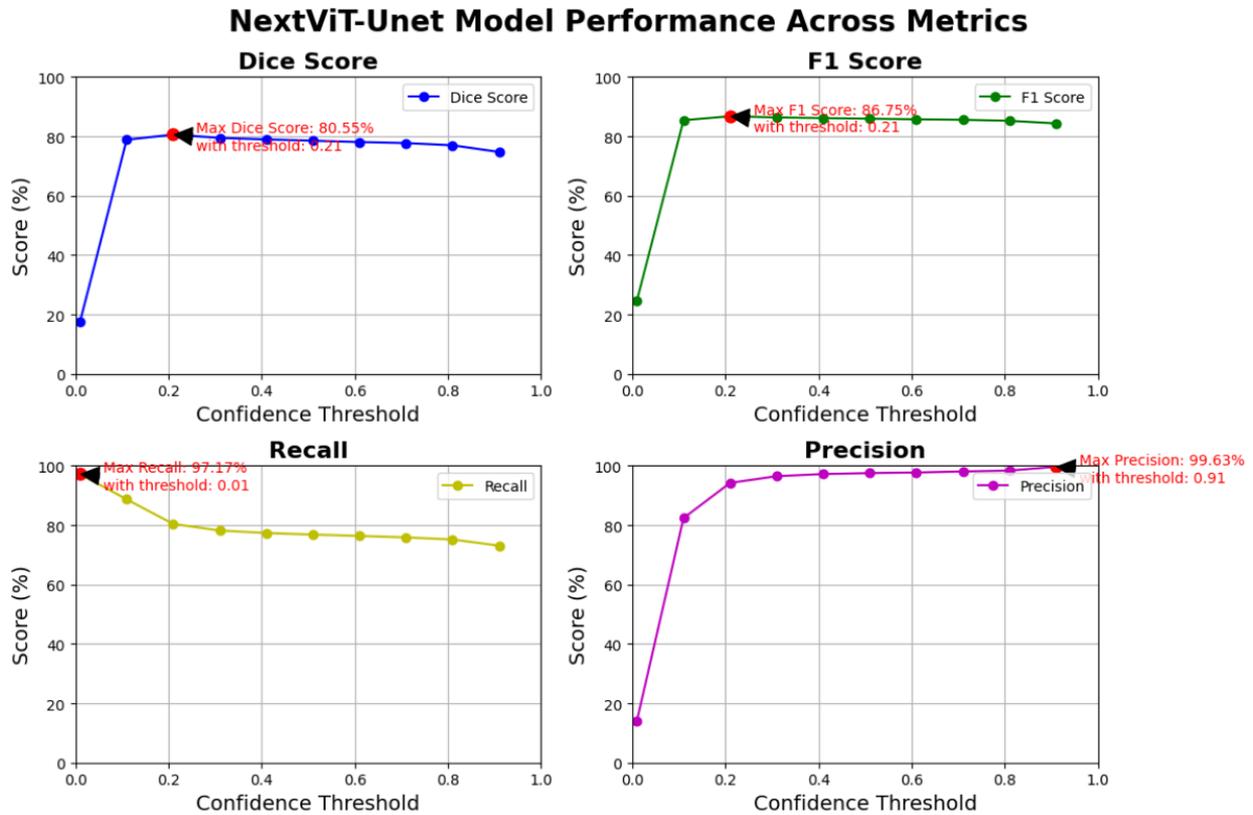


Figure 66. Metrics obtained with the NextViT U-Net model

The interpretation of the results is similar to the previous case, with the notable improvement that the model now achieves **an F1-score of 86.75%** and **a Dice Score of 80.55%**. These enhanced metrics indicate an even **better balance between precision and recall**, and the high Dice Score suggests excellent spatial overlap between the predicted and ground truth masks. **This performance in terms of the Dice Score is considered state-of-the-art**, reflecting significant advancements in the model's ability to accurately detect and delineate contrails.

The **computational time** for training **20 epochs** was approximately **24 hours**, running on a single NVIDIA RTX 4090 GPU with 12GB of memory.

C) Performance of the Yolov8 model

The best metrics obtained with this model when we evaluate it over the validation set of the OpenContrails Dataset are shown in Figure 67.

The results indicate that the model's recall remains consistently high across varying confidence thresholds, suggesting its **robustness in capturing positive instances** regardless of confidence levels.

Conversely, precision improves notably with higher confidence thresholds, indicating the model's increased selectivity in true positive predictions.

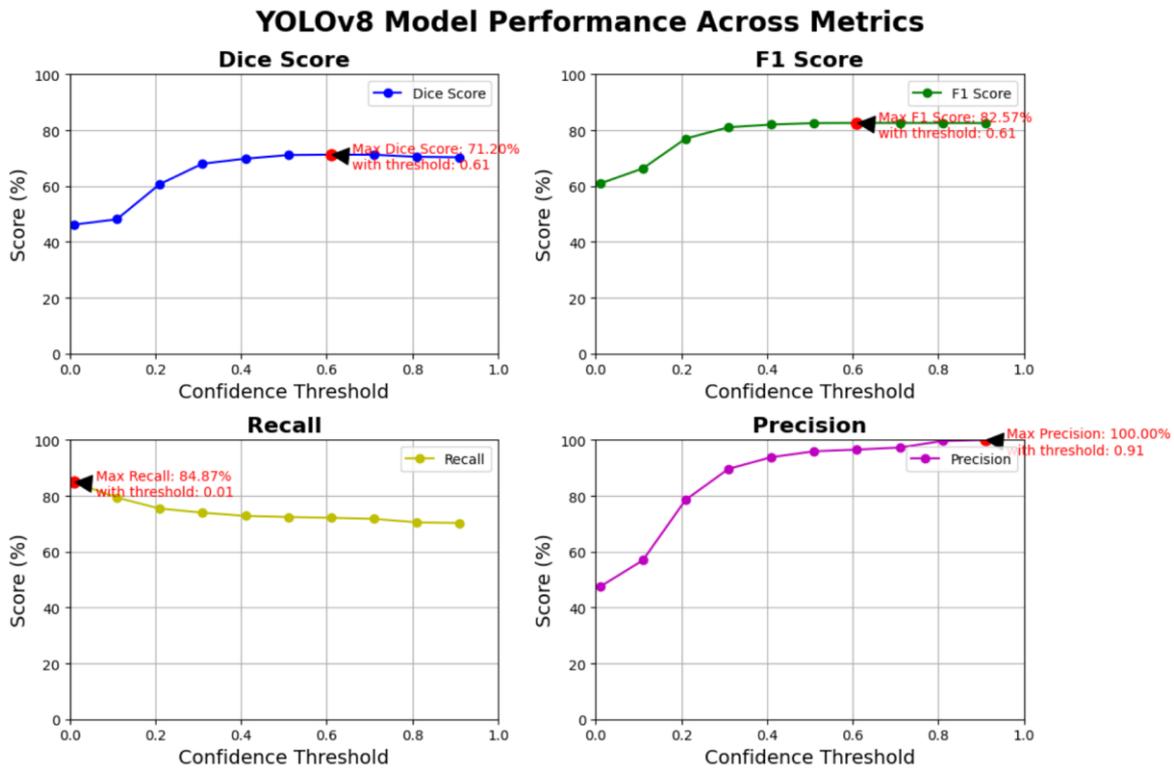


Figure 67. Metrics obtained with the YOLOv8 model

The Dice score demonstrates good overlap between predicted and ground truth masks at a relatively high confidence threshold (0.61).

However, the notably higher F1 score compared to the Dice score at the same confidence level suggests that *the model excels more in accurately classifying pixels as belonging to a contrail or not than in precisely delineating the shapes of segmented objects*, emphasizing its strength in pixel-wise classification accuracy.

The **computational time** for training **120 epochs** was approximately **40 hours**, running on a single NVIDIA RTX 4090 GPU with 12GB of memory.

D) Performance of the Mask-RCNN model

The metrics obtained with the Mask-RCNN model when evaluating it over the validation set of OpenContrails Dataset are shown in Figure 68.

The evaluation of this model reveals a noteworthy pattern: while its recall consistently hovers around 71% across all confidence thresholds, indicating its *capability to capture positive instances*, its *precision score has a strong dependence on the confidence threshold selected*.

This suggests that the model tends to generate numerous false detections at low confidence levels, while it's capable to maintain balance at high threshold. In essence, has a good performance in both Dice Score and F1 Score, however, *in comparison with the previous models it has a worse performance*.

The **computational time** for training **2,500 epochs** was approximately **20 minutes**, running on a single NVIDIA RTX 4090 GPU with 12GB of memory.

Mask R-CNN Model Performance Across Metrics

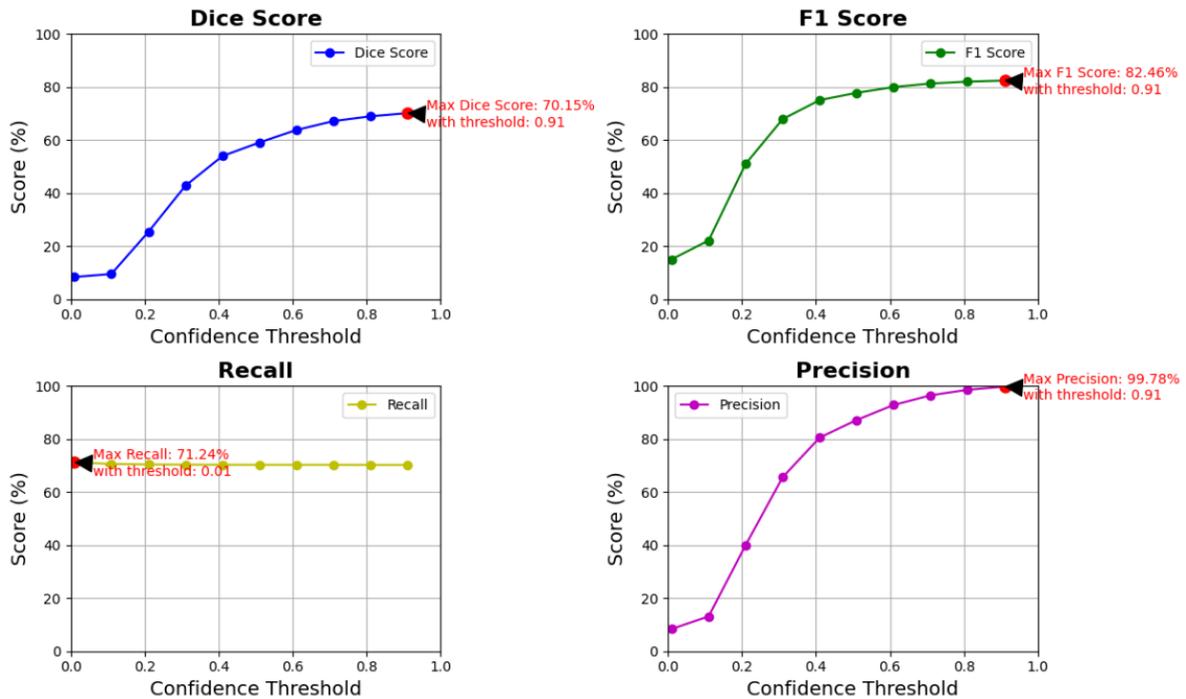


Figure 68. Metrics obtained with the Mask R-CNN model

E) **Performance of the Weighted Ensemble Model:** We utilized black box optimization to determine the optimal weights for maximizing the Dice Score. The optimal weights identified were 0.8 for the NextViT model, 0.1 for CoaT, and 0.05 each for Mask R-CNN and YOLOv8. The results are shown in Figure 69

However, *this combination only improved the Dice Score by 0.01% compared to the best-performing single model*. Consequently, the computational cost of using all four models outweighs the minimal performance gain. *Further exploration of ensembling techniques is necessary to achieve a substantial performance improvement.*

Weighted Ensemble Model Performance Across Metrics

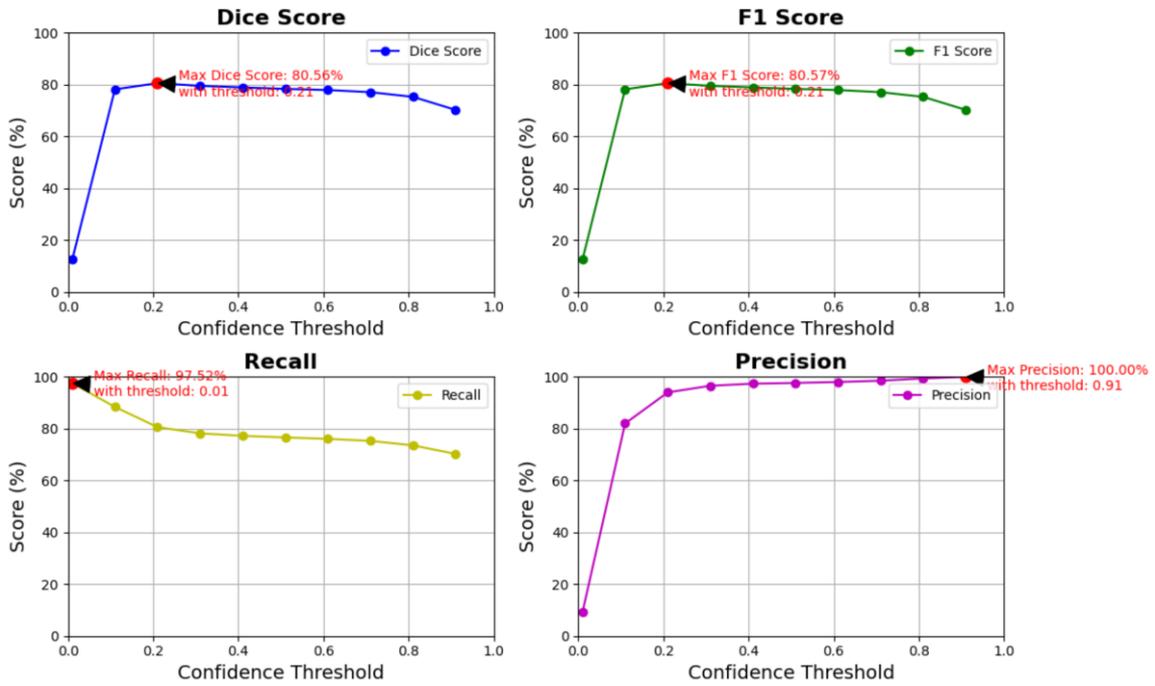


Figure 69. Metrics obtained with the Weighted Ensemble model

4.1.4 Comparison across all trained models

For a more comprehensive understanding of how the various models compare in terms of their performance according to the dice score functions, we have made a unique plot shown in Figure 70.

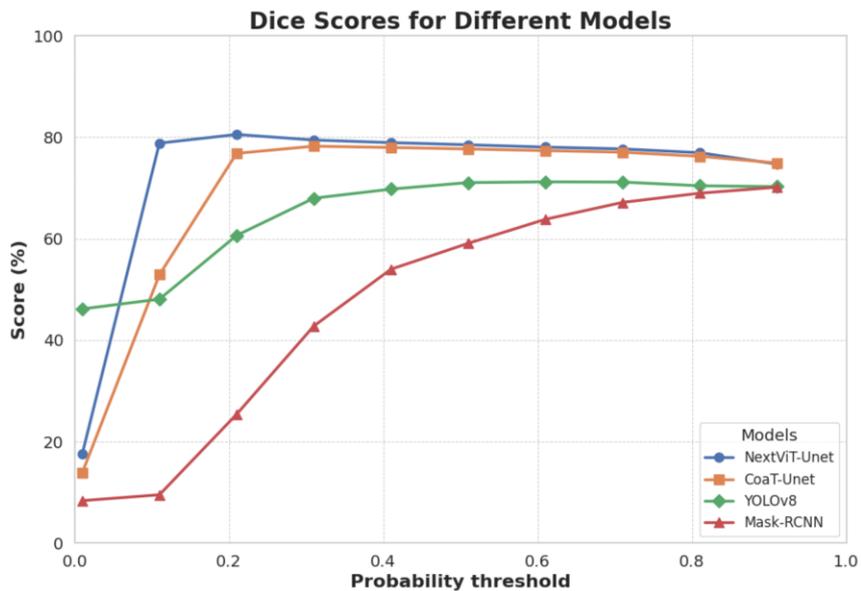


Figure 70. Comparison between of the Dice Score as a function of the probability threshold across models

By selecting the threshold that optimizes the dice score for each model, we obtain the values displayed in Figure 71.

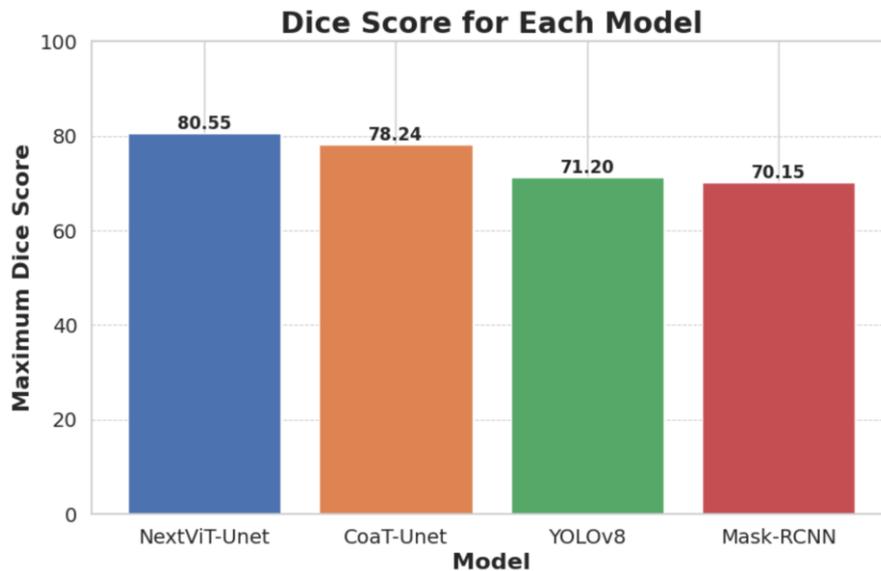


Figure 71. Best Dice Score for each model for a selected threshold

4.1.5 Comparison with Previous Results

The results of the top-performing model within the Google Research Kaggle competition [38] are shown in Figure 72. We notice that the highest score, reaching 71.6%, is achieved by a single model. Interestingly, *our YOLOv8 model nearly matches this score, while two other models (CoaT and NextViT U-Nets) actually exceed it.*

Scores				
		CV	Public	Private
	input size			
Single time	512	0.697	0.707	0.712
Single time	1024	0.703	0.719	0.716
4-panel	1024	0.704	0.719	0.722
Ensemble		0.706	0.725	0.724

Figure 72. Dice Score of MaxViT U-Net 1st place solution of the Google Research competition

Our implementation of the Coat U-Net and NextVit U-Net models was based on the approach used by the team that achieved the 2nd position on the leaderboard. These results are displayed in Figure 73, along with the results obtained with several other models.

Model	CV (single fold)	CV	Public LB	Private LB
EfficientNet v2-s		0.699	0.704	0.699
CoaT_U (single image)	0.6887±0.0014	0.6983	0.70694	0.70073
ExCoaT_U (single image)	0.6952±0.0008	0.7002	0.71648	0.70858
NeXtViT_U (single image)	0.6848±0.0013	0.6917	0.71038	0.69617
ExNeXtViT_U (single image)	0.6904±0.0006	0.6941	0.71399	0.70492
CoaT_ULSTM	0.6960±0.0003	0.7039	0.71243	0.71790
Ex CoaT_ULSTM	0.7027±0.0002	0.7064	0.72136	0.71705
CoaT_UT	0.6950±0.0013	0.7055	0.72257	0.71284
Ex CoaT_UT	0.6997±0.0008	0.7040	0.71630	0.71125
NeXtViT_ULSTM	0.6951±0.0004	0.7014	0.71308	0.70831
Ex NeXtViT_ULSTM	0.6988±0.0007	0.7025	0.71925	0.71432
SAM (single image)	0.694±0.0008	0.694	0.698	0.704
SAM (frame 2,3,4)	0.696±0.0008	0.696	0.698	0.706

Figure 73. Dice Score of the models presented in the 2nd place solution of the Google Research competition

We surpassed the performance of the two top-performing models *by reconsidering our optimization strategy*. Initially, we minimized the Dice Lovasz Loss to strike a balance between precision and recall. However, upon observing the results, we noticed that the models tended to prioritize higher recall.

To address this, we revamped the loss function into a *Beta Lovasz Hinge function*, assigning a significantly lower weight to recall maximization and emphasizing precision maximization. We set the *Beta parameter to a value of 0.1*, as smaller values caused gradient vanishing while higher values degraded performance. This adjustment led to improved results.

4.1.6 Preliminary experiments with European data

To evaluate the ability of our models to detect contrails across various data sources covering the European geographical extent, we conducted preliminary experiments using MSG/SEVIRI data. Originally, we planned to use MTG data for these experiments because it offers better resolution and same properties as the GOES-16 images on which the models have been trained. However, *the availability of MTG data has been delayed due to ongoing corrections needed to address artifacts found in the initial test images*.

Experiment conducted on the 25th of September 2023

The initial experiment conducted with data covering European regions focused on areas including the northern Spain, Italy, France, and several other northern countries within specific latitude and longitude coordinates. This experiment took place on September 25, 2023, between 8:30 am and 10:30 am.

The selection of this date, time, and location was primarily motivated by the **visibility of a contrail observed in Toulouse through ground camera observation**. This contrail was observed from around 9 am to 10 am, providing a ground truth observation against which we could compare the data obtained from MSG/SEVIRI images. In Figure 74, the region of coverage is depicted along with the precise location of the camera and the detected contrail, providing visual context for the experiment.

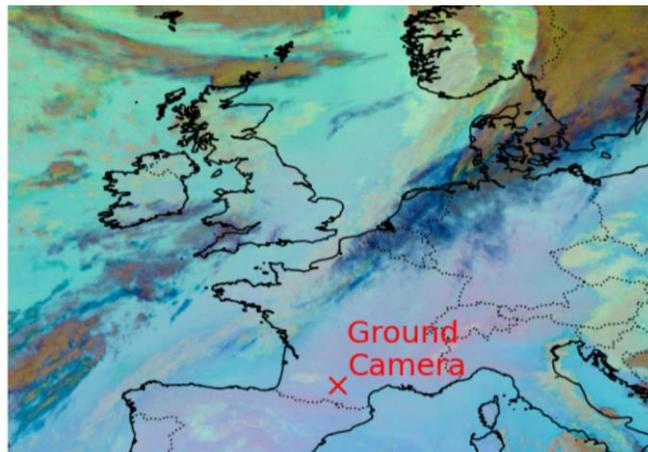


Figure 74. Area of coverage of the 1st experiment conducted in Europe along with the location of the ground camera contrail observation. The scene is an Ash-RGB MSG/SEVIRI image.

In this phase of the study, we utilized various models trained on GOES-16 images to identify and analyse contrails present in the imagery. These models were employed to detect existing contrails and compare them with labelled contrails within the images, as well as with the contrail observed in Toulouse through ground camera footage.

The outcomes of these detection efforts are depicted in Figure Y. The key observations derived from these results are as follows:

- ❖ The contrail observed in the ground camera image does not appear visible in the MSG/SEVIRI image (see Figure 75) and consequently remains undetected by the model. **This discrepancy is likely due to the size of the contrail.** It's important to note that each pixel in the false colour RGB image corresponds to an area of 3km x 3km, which means that if the contrail's width is smaller than this size, it becomes challenging for the model to detect it.

❖



Figure 75. MSG/SEVIRI Ash-RGB image on the 25/09/2023 with a red circle covering Toulouse.

- ❖ Although the model identifies several contrails, it also generates numerous false positives. To address this issue, the post-processing techniques outlined earlier have been applied to the images.

Some visual examples of the contrails detected are shown in Figure 76.

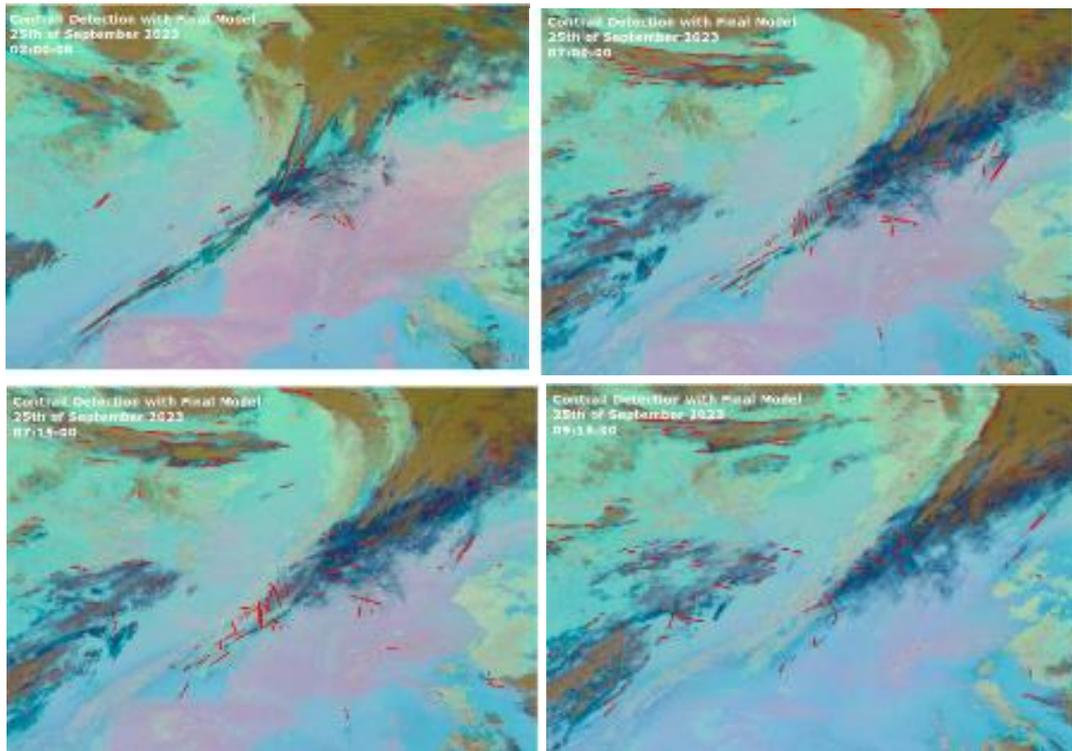


Figure 76. Contrails detected by the models on the 25/09/2023 in MSG/SEVIRI Ash-RGB images.

Experiment Conducted on the 26th of January 2024

In the context of this second scenario, we have selected it because *several contrails were detected by visual inspection* (see Figure 77) on the images and therefore, we wanted to contrast our observations with the capabilities of the model.

The experiment was conducted on 26th of January of 2024 from 7:30am to 11:30am, in the Spanish region mainly, as well as north Africa and other countries in middle-Europe.

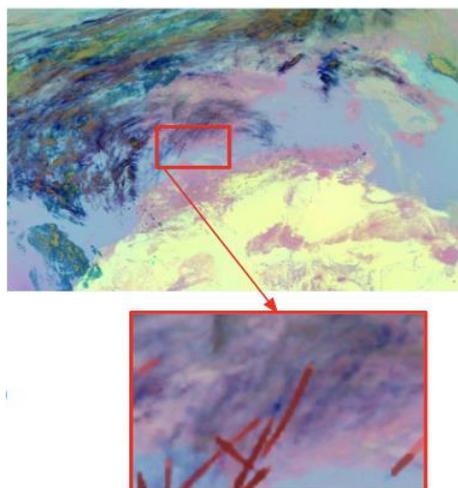


Figure 77. An example of contrails labelled on an MSG/SEVIRI Ash-RGB image on the 26/01/2024 by visual inspection.

The contrail detection results are displayed in the figure. Despite our postprocessing efforts to eliminate false detections, further refinement is necessary to achieve the most reliable predictions.

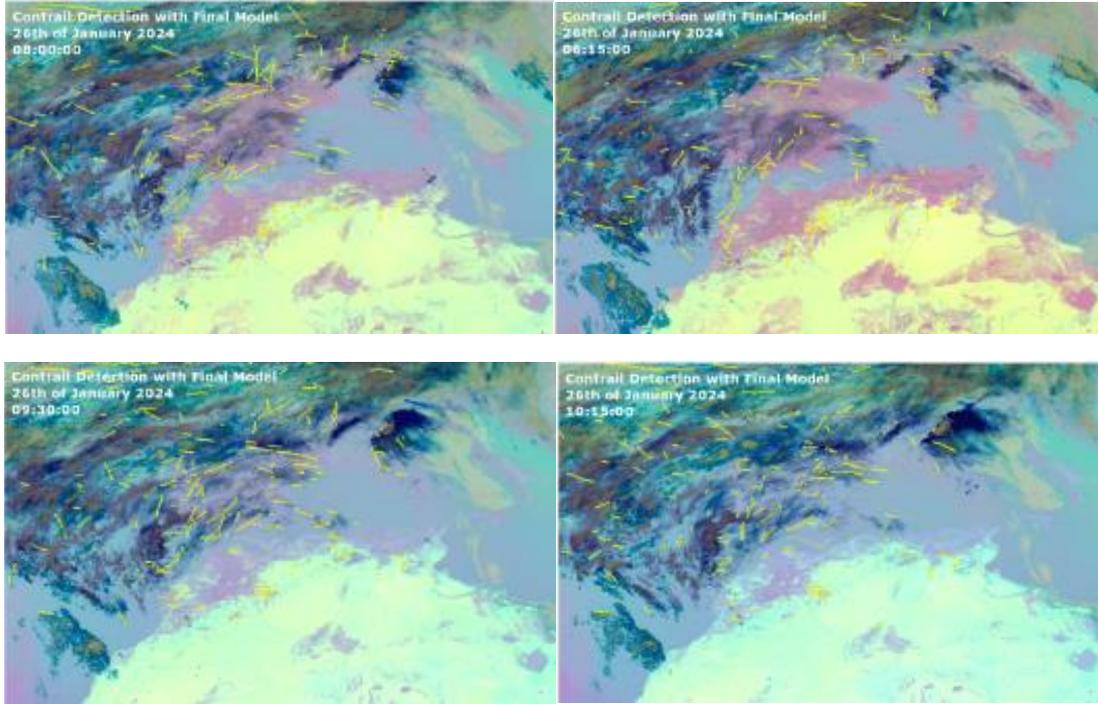


Figure 78. Contrails detected by the models on the 26/01/2024 in MSG Ash-RGB images.

Figure 79 demonstrates how incorporating optical flow to propagate masks from previous steps and merging them with detections at each time step can ensure consistent model detections across a sequence of images.

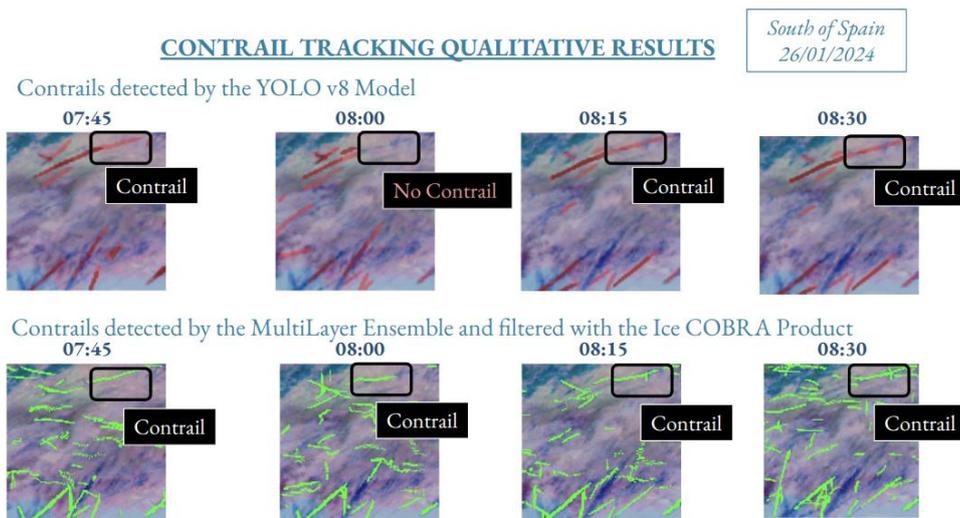


Figure 79. Ensuring temporal consistency throughout an image sequence making optical flow corrections on the MSG Ash-RGB images from the 26/01/2024.

4.2 Physics-Driven Contrail Simulation Model

In this section, we present some simulation results for non-persistent and persistent contrails and the latter is (roughly) compared against that of CoCip model.

4.2.1 non-persistent contrails

Non-persistent contrail simulation refers to the contrails that dissipate soon after they are formed. This is because the Schmidt-Appleman criteria is not fully satisfied. This situation can be simulated by deactivating the source term representing the growth of ice particles.

Figure 80 shows the evolution of a one single contrail for a hypothetical flight with a wind field randomized for about 30% about the available wind data. Figure 81 and Figure 82 exhibit the results of a simulation at 10km level flight with standard atmospheric quantities and real wind data above Madrid (30-30 squared kilometres). Specifically, these figures depict the evolution of non-persistent overlapping contrails due to the flight of 4 aircraft.

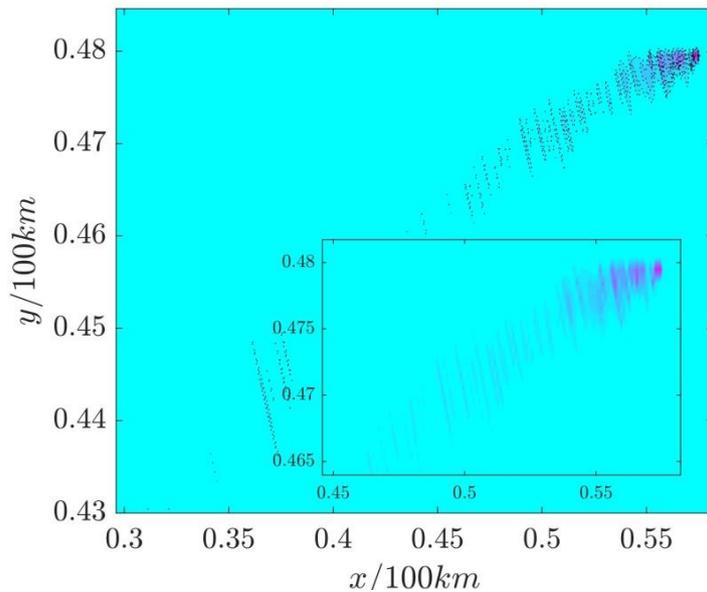


Figure 80: Single non-persistent contrail

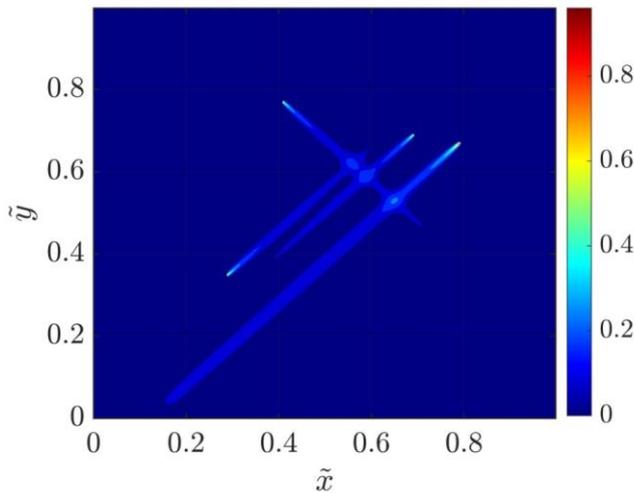


Figure 81: non-persistent contrails from 4 aircraft after 90 seconds

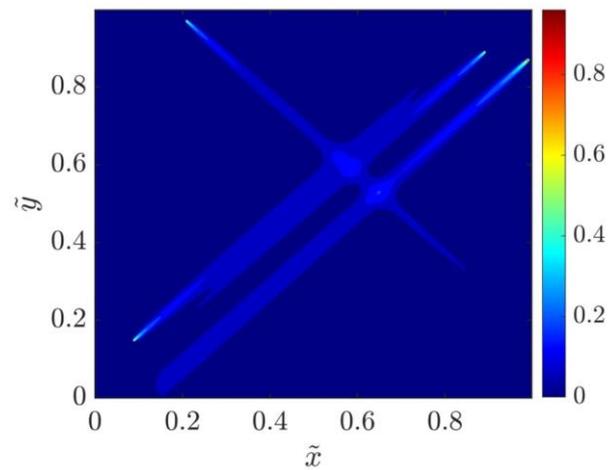


Figure 82: non-persistent contrails for 4 aircraft after 150 seconds

4.2.2 Persistent contrails

As previously mentioned, persistent contrails are ice particles released in areas with conditions conducive to their growth. presents a simulation for a hypothetical flight from Gran Canaria to Madrid at 8 a.m. on February 12, 2023. This figure shows that after one hour of flight, the aircraft is just beginning to encounter an area prone to the formation of persistent contrails. Figure 81 extends this simulation by another hour. Figure 82 displays a similar simulation (for the same flight and time) using CoCiP. Although the exact time span used in the CoCiP algorithm is unclear, a comparison between Figure 81 and Figure 82 already reveals a fair level of correspondence.

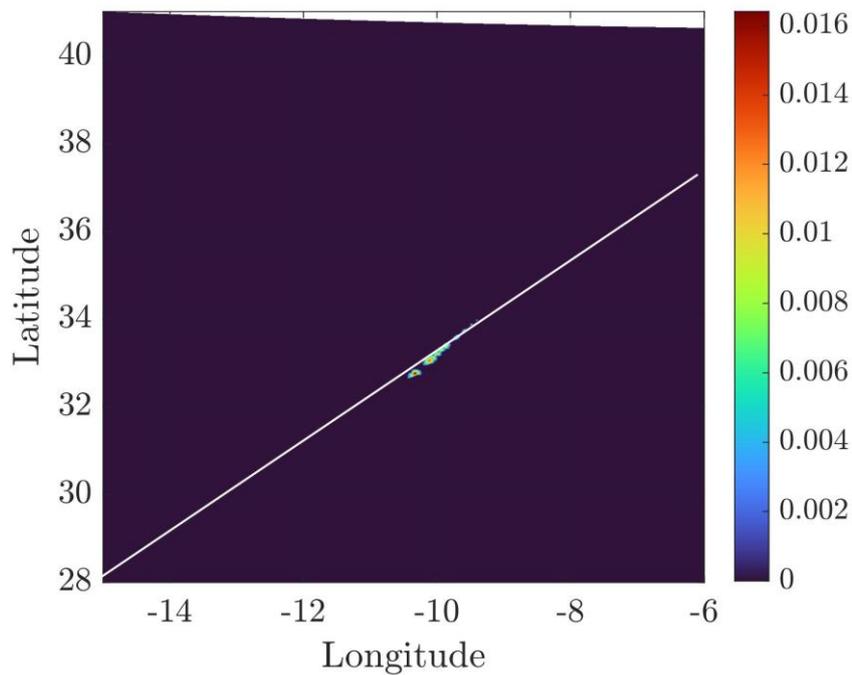


Figure 81: persistent contrails for a single flight after about 1 hour (colourbar: kg/m^3)

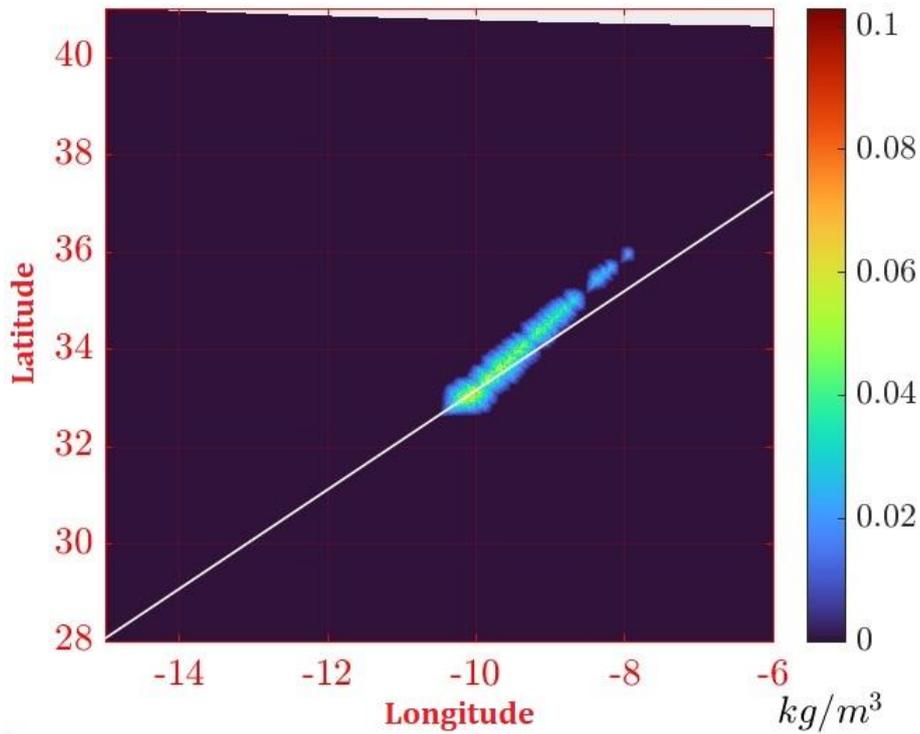


Figure 82: persistent contrail for a single flight after about 2 hours

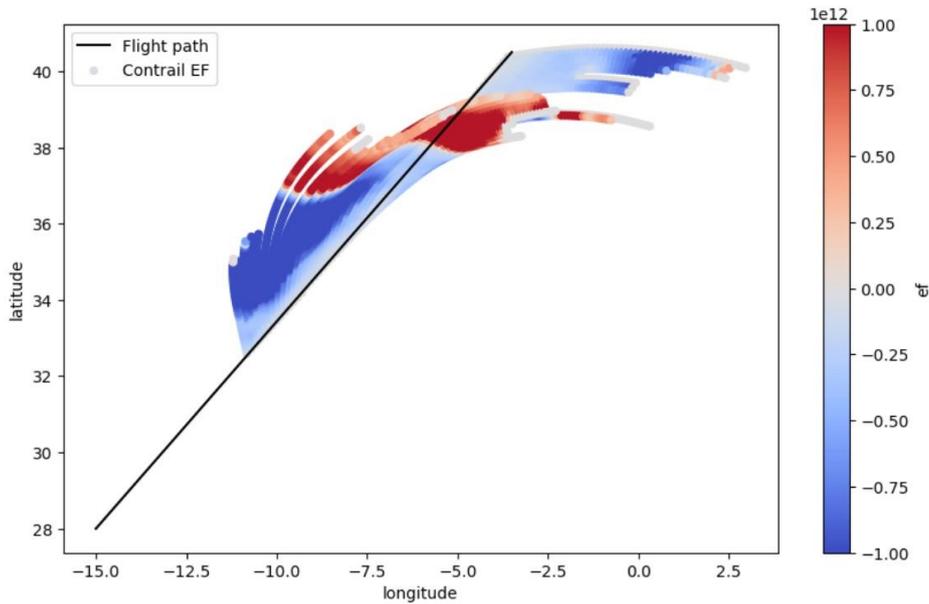


Figure 83: contrail's energy forcing from CoCiP model

4.2.3 Random Traffic:

A simulation was carried out for 200 flights over an area with 1000 km-by –1000 km size and Madrid at its centre on 24th, Dec., 2023 at 10 km altitude. The simulation highlights the spotted regions of growing/persistent contrails after half an hour flight. Specifically, Figure 84 shows the accumulation of

all generated contrails after half an hour flight, while Figure 85 shows the spotted regions where some of those contrails have met the specific conditions to grow.

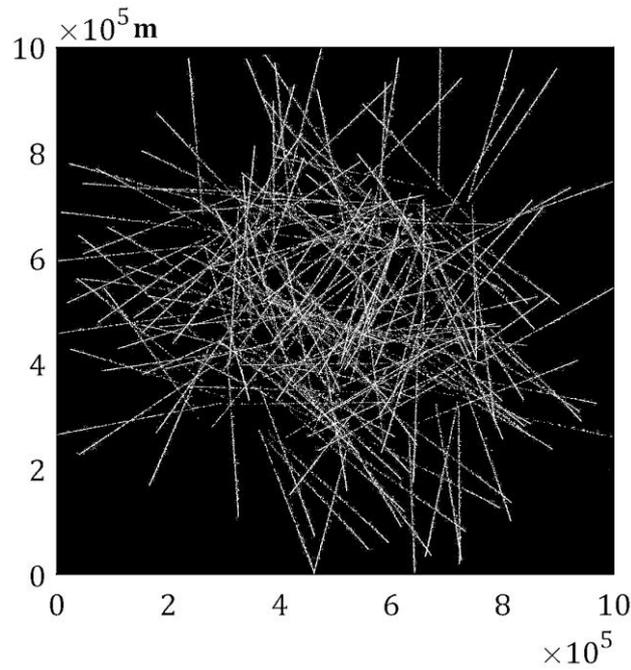


Figure 84: Contrails generated by a traffic after about 30 minutes flights

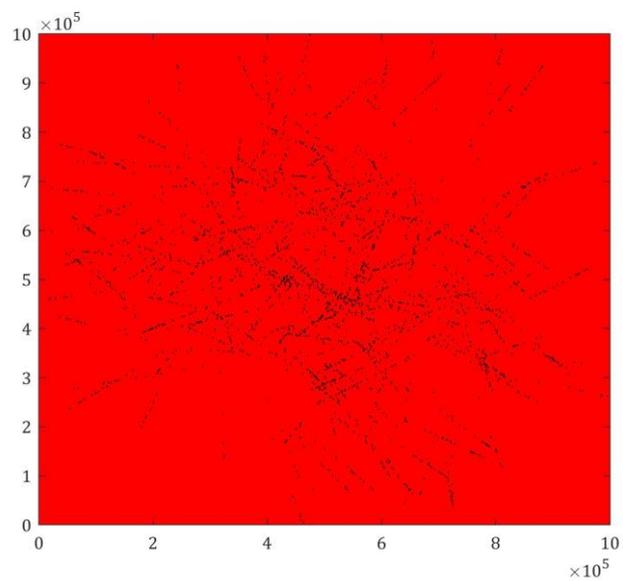


Figure 85: Persistent contrail spots after about 30 minutes (axes are in meters)

5 Conclusions

The work presented in this deliverable has successfully met the primary goal of developing a model capable of detecting contrails in satellite imagery and another model for characterizing the evolution of contrails into aviation-induced cloudiness over time.

We have achieved nearly a 10% improvement in terms of the dice score on the contrail detection models. We have also outlined all the required pre- and post-processing steps necessary to attain optimal performance.

Additionally, we have demonstrated that these models can be directly applied to European data, though quantitative validation of detections using MSG/SEVIRI images is still pending. This labelling effort will be undertaken and included in the next deliverable. We anticipate that the results may not be as strong as those from GOES-16 scenes due to the lower resolution of MSG/SEVIRI scenes. However, once MTG data becomes available, the performance is likely to match that of the GOES-16 data.

Regarding the physics-driven contrail simulation model, we have implemented a novel transport equation aimed at addressing existing slip mechanisms, thereby simulating the advection-diffusion of ice particles. It is noteworthy that the propagation of persistent contrails is governed by a comprehensive advection-diffusion equation. A thorough comprehension of this equation is imperative for more precise simulations of persistent contrails and, consequently, radiative forcing. Although the presented model still necessitates refinement and extension at an algorithmic level, we have observed a reasonable level of consistency compared to other similar methods such as CoCiP. The subsequent steps involve enhancing the developed code in the following aspects:

- 1- Physics representation: This implies updating the model to incorporate additional physical processes.
- 2- Large-scale simulation: Currently, the computational time required for tracking contrails over expansive domains, such as the entire Europe, is substantial. Efforts are underway to devise strategies for enhancing algorithm efficiency in this regard.

6 References

- [1] D. Lee, D. Fahey, A. Skowron, M. Allen, U. Burkhardt, Q. Chen, S. F. S.J. Doherty, P. Forster, J. Fuglestedt y R. A. Gettelman, «The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018,» *Atmospheric Environment*, vol. 244, p. 117834, 2021.
- [2] G. a. S. F. Myhre, «On the tradeoff of the solar and thermal infrared radiative impact of contrails,» *Geophysical Research Letters*, vol. 28, nº 16, pp. 3119-3122, 2001.
- [3] R. M. Hermann Mannstein y P. Wendling, «Operational detection of contrails from NOAA-AVHRR-data,» *International Journal of Remote Sensing*, vol. 20, nº 8, pp. 1641-1660, 1999.
- [4] G. Zhang, J. Zhang y J. Shang, «Contrail Recognition with Convolutional Neural Network and Contrail Parameterizations Evaluation,» *Sola*, vol. 14, pp. 132-137, 2018.
- [5] J. Y.-H. Ng, K. McCloskey, J. Cui, V. R. Meijer, E. Brand, A. Sarna, N. Goyal, C. V. Arsdale y S. Geraedts, «OpenContrails: Benchmarking Contrail Detection on GOES-16 ABI,» *arXiv*, 2023.
- [6] J.-F. Gayet, G. Febvre, G. Brogniez, H. Chepfer, W. Renger y P. Wendling, «Microphysical and optical properties of cirrus and contrails: cloud field study on 13 october 1989,» *Journal of the Atmospheric Sciences*, vol. 53, pp. 126-138, 1996.
- [7] W. K. Pratt, de *Digital Image Processing*, (New York: John Wiley, 1991).
- [8] D. Meinert, « Training a neural network to detect jet contrails in satellite images.,» de *Impact of Emissions from Aircraft and Spacecraft upon the Atmosphere, Proceedings of an International Scientific Colloquium*, KoÈln, Germany, 1994.
- [9] K. J. F. McCloskey, S. D. Geraedts, B. H. J. V. R. Meijer, E. W. Brand, D. K. Fork, J. Platt, C. Elkin y C. H. V. Arsdale, «A human-labeled Landsat contrails dataset,» 2021.
- [10] A. Bhandari, S. Rallabandi, S. Singhal, A. Kasliwal y P. Seth, «Performance evaluation of deep segmentation models for Contrails detection,» *arXiv preprint arXiv:2211.14851*, 2022.
- [11] O. Ronneberger, P. Fischer y T. Brox, «. U-net: Convolutional networks for biomedical image segmentation,» de *International Conference on Medical image computing and computer-assisted intervention*,, 2015.
- [12] H. Zhao, J. Shi, X. Qi, X. Wang y J. Jia, «. Pyramid scene parsing network,» de *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [13] L.-C. Chen, G. Papandreou, F. Schroff y H. Adam, «Rethinking atrous convolution for semantic image segmentation.,» *arXiv preprint arXiv:1706.05587*, 2017.

- [14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff y H. Adam, «Encoder-decoder with atrous separable convolution for semantic image segmentation.,» de *Proceedings of the European conference on computer vision (ECCV)*, 2018.
- [15] «GOES Ash RGB Quick Guide,» 2017. [En línea]. Available: https://rammb.cira.colostate.edu/training/visit/quick_guides/GOES_Ash_RGB.pdf. [Último acceso: 2024 January 5].
- [16] P. Bogacki y L. Shampine, «A 3(2) pair of Runge - Kutta formulas,» *Applied Mathematics Letters*, vol. 2, nº 4, pp. 321-325, 1989.
- [16a] Lewellen, D. C. (2014). Formation and persistence of contrails. *Journal of Atmospheric Sciences*.
- [16b] Kärcher, B., & Lohmann, U. (2002). A parameterization of cirrus cloud formation. *Journal of Geophysical Research: Atmospheres*.
- [16c] Burkhardt, U., & Kärcher, B. (2009). Global radiative forcing from contrail cirrus. *Nature Climate Change*.
- [16d] Schumann, U. (2012). A contrail cirrus prediction model. *Geoscientific Model Development*.
- [17] F. Bolelli, S. Allegretti, L. Baraldi y C. Grana, «Spaghetti Labeling: Directed Acyclic Graphs for Block-Based Connected Components Labeling,» *IEEE Transactions on Image Processing*, vol. 29, pp. 1999-2012, 2020.
- [18] R. O. Duda y P. E. Hart, «Use of the Hough transformation to detect lines and curves in pictures,» *Commun. ACM*, vol. 15, nº 1, p. 11–15, 1992.
- [19] R. Aramini, F. Delbary, M. C. Beltrametti, M. Piana y A. M. Massone, «The Radon transform and the Hough transform: a unifying perspective,» *arXiv*, 2016.
- [20] J. Canny, «A computational approach to edge detection,» *IEEE Transactions on pattern analysis and machine intelligence*, nº 6, pp. 679-698, 1986.
- [21] R. Olaf, F. Philipp y B. Thomas, «U-net: Convolutional networks for biomedical image segmentation,» de *Medical image computing and computer-assisted intervention--MICCAI 2015: 18th international conference*, Munich, German, 2015.
- [22] J. Li, X. Xia, W. Li, H. Li, X. Wang, X. Xiao, R. Wang, M. Zheng y X. Pan, «Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios,» *arXiv preprint arXiv:2207.05501*, 2022.
- [23] W. Xu, Y. Xu, T. Chang y Z. Tu, «Co-scale conv-attentional image transformers,» *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.
- [24] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik y Y. Li, «Maxvit: Multi-axis vision transformer,» de *European conference on computer vision*, 2022.

- [25] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, «You only look once: Unified, real-time object detection,» de *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [26] C.-Y. Wang, A. Bochkovskiy y H.-Y. M. Liao, «YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,» de *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 7464--7475*, 2023.
- [27] K. He, G. D. P. Gkioxari y R. Girshick, «Mask r-cnn,» de *Proceedings of the IEEE international conference on computer vision*, 2017.
- [28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li y L. Fei-Fei, «ImageNet: A large-scale hierarchical image database,» de *IEEE Conference on Computer Vision and Pattern Recognition*, 2009 .
- [29] B. D. Lucas y T. Kanade, «An iterative image registration technique with an application to stereo vision,» de *IJCAI'81: 7th international joint conference on Artificial intelligence,2*, 674--679, 1981.
- [30] U. Schumann, «On conditions for contrail formation from aircraft exhaust,» *Meteorol. Zeitschrift*, vol. 5, pp. 4-23, 1996.
- [31] U. Schumann, P. Konopka, R. Baumann, R. Busen, T. Gerz, T. Schlager, P. Schulte y H. Volkert, «Estimate of diffusion parameters of aircraft exhaust plumes near the tropopause from nitric oxide and turbulence measurements,» *Journal of Geophysical Research*, vol. 100, pp. 147-162, 1995.
- [32] K. M. Gierens, M. Monier y J. F. Gayet, «The deposition coefficient and its role for cirrus clouds,» *Journal of Geophysical Research*, vol. 108, 2003.
- [33] H. Klett, R. Pruppacher y J. D., «Microphysics of clouds and precipitation,» *Norwell, Mass: Kluwer Academic*, 2000.
- [34] J. Li, J.-H. Kim, B. Sridhar y H. K. Ng, «Ames Contrail Simulation Model: Modeling Aviation Induced Contrails and the Computation of Contrail Radiative Forcing Using Air Traffic Data,» *NASA/TM-202300014633*, 2023.
- [35] S. Tu, X. Liu y H. C. , «Effect of gravity on colloidal particle transport in a saturated porous medium: Analytical solutions and experiments,» *Plos One*, vol. 17, nº 10, 2022.
- [36] J. Buongiorno, «Convective Transport in Nanofluids,» *Heat Transfer, Transactions of the ASME*, vol. 128, 2006.
- [37] A. Jafarimoghaddam, «The Linearly Stretching Wall Jet,» *European Journal of Mechanics, B/Fluids*, vol. 80, pp. 52-59, 2021.
- [38] J. Ng, C. Elkin, A. Sarna, W. Reade y M. Demkin, «Google Research - Identify Contrails to Reduce Global Warming,» Kaggle, May 2023. [En línea]. Available: <https://kaggle.com/competitions/google-research-identify-contrails-reduce-global-warming>.

- [39] W. Krebs, H. Mannstein, L. Bugliaro y B. Mayer, «Technical note: A new day- and night-time Meteosat Second Generation Cirrus Detection Algorithm MeCiDA,» *Atmospheric Chemistry and Physics*, vol. 7, nº 24, pp. 6145--6159, 2007.
- [40] M. Vazquez-Navarro, H. Mannstein y S. Kox, «Contrail life cycle and properties from 1 year of MSG/SEVIRI rapid-scan images,» *Atmospheric Chemistry and Physics*, vol. 15, nº 15, pp. 8739--8749, 2015.
- [41] A. Gettelman, C.-C. Chen y C. G. Bardeen, «The climate impact of COVID-19-induced contrail changes,» *Atmospheric Chemistry and Physics*, vol. 21, nº 12, pp. 9405--9416, 2021.
- [42] H. Appleman, «The Formation of Exhaust Condensation Trails by Jet Aircraft,» *Bulletin of the American Meteorological Society*, vol. 34, nº 1, pp. 14 - 20, 1953.
- [43] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo y Be, «The ERA5 global reanalysis,» *Quarterly Journal of the Royal Meteorological Society*, vol. 146, nº 730, pp. 1999-2049, 2020.
- [44] U. Schumann, «On conditions for contrail formation from aircraft exhausts,» *Meteorologische Zeitschrift*, vol. 5, nº 1, pp. 4-23, 1996.
- [45] GISGeography, «Sentinel 2 Bands and Combinations,» 9 March 2024. [En línea]. Available: <https://gisgeography.com/sentinel-2-bands-combinations/>. [Último acceso: 15 February 2024].
- [46] P. Heiselberg y H. Heiselberg, «Aircraft Detection above Clouds by Sentinel-2 MSI Parallax,» *Remote Sensing*, vol. 13, nº 15, 2021.
- [47] H. Mannstein, A. Br\"omser y L. Bugliaro, «Ground-based observations for the validation of contrails and cirrus detection in satellite imagery,» *Atmospheric Measurement Techniques*, vol. 3, nº 3, pp. 655--669, 2010.
- [48] S. R. AB, «Flightradar24,» [En línea]. Available: <https://www.flightradar24.com/40.55,-3.64/6>.
- [49] Y. Borthomieu, «14 - Satellite Lithium-Ion Batteries,» de *Lithium-Ion Batteries*, Amsterdam, Elsevier, 2014, pp. 311-344.
- [50] P. Shen, D. Liu, I. Gultep, H. Lin, N. Cai, S. Cao y Z. Wang, «Boundary Layer Features of One Winter Fog in the Yangtze River Delta, China,» *Pure and Applied Geophysics*, vol. 179, nº 9, pp. 3463--3480, 2022.
- [51] L. Palacios-Peña, R. Lorente-Plazas, J. P. Montávez y P. Jiménez-Guerrero, «Saharan Dust Modeling Over the Mediterranean Basin and Central Europe: Does the Resolution Matter?,» *Frontiers Earth Science*, vol. 7, pp. 2296-6463, 2019.
- [52] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin y H. Hu, «Video swin transformer,» de *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3202--3211, 2022.

- [53] L. Hedegaard y A. Iosifidis, «Continual 3D convolutional neural networks for real-time processing of videos,» de *European Conference on Computer Vision*, 2022.
- [54] S. Hochreiter y J. Schmidhuber, «Long Short-Term Memory,» *Neural Computation*, vol. 9, nº 8, pp. 1735-1780, 1997.

7 List of acronyms

Acronym	Description
ADI	Alternating Directional Implicit
ADS-B	Automatic Dependent Surveillance-Broadcast
AIC	Aviation Induced Cloudiness
AP	Average Precision
AVHRR	Advanced Very-High-Resolution Radiometer
BT	Brightness Temperature
BTD	Brightness Temperature Difference
CDA	Contrail Detection Algorithm
CloU	Complete Intersection over Union
CNN	Convolutional Neural Network
CoaT	Co-Scale Conv-Attentional Image Transformer
CO2	Carbon Dioxide
DFL	Distribution Focal Loss
ECMWF	European Center for Medium-Range Weather Forecasts
ERA5	5 th Generation of ECMWF atmospheric reanalysis
ELAN	Efficient Layer Aggregation Network
EMHSA	Efficient Multi-Head Self-Attention
ERF	Effective Radiative Forcing
ESA	European Space Agency
EUMETSAT	European Organization for the Exploitation of Meteorological Satellites
FC	Fully Connected
FCN	Fully Convolutional Network
FN	False Negatives
FP	False Positives
GeLU	Gaussian Error Linear Unit
GEO	Geosynchronous Equatorial Orbit
GOES-16	Geostationary Operational Environmental Satellites - 16 Series
HSV	Hue-Saturation-Value
IoU	Intersection over Union
IR	Infrared
IRS	Infrared Sensor
ISSR	Ice Super Saturated Regions
LEO	Low Earth Orbit

LSTM	Long Short-Term Memory
LWIR	Long Wave Infrared
mAP	Mean Average Precision
MaxViT	Multi-Axis Vision Transformer
MBConv	Mobile Inverted Bottleneck Convolution
MeCIDA	Meteosat Second Generation Cirrus Detection Algorithm
MHCA	Multi-Head Convolutional Attention
MSG	Meteosat Second Generation
MTG	Meteosat Third Generation
NextViT	Next Generation Vision Transformer
NCB	Next Convolutional Block
NIR	Near Infrared
NOAA	National Oceanic and Atmospheric Administration
NOx	Nitrogen Oxides
NTB	Next Transformer Block
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Units
ResNet	Residual Network
RHI	Relative Humidity over ice
RoI	Region of Interest
RGB	Red, green and blue
RPN	Region Proposal Network
SAC	Schmidt-Appleman Criterion
SEVERI	Spinning Enhanced Visible and Infra-Red Imager
SiLU	Sigmoid Linear Unit
SPPF	Spatial Pyramid Pooling Fusion
SWIR	Short Wave Infrared
TN	True Negatives
TP	True Positives
UT	Universal Time
VIS	Visible
YOLO	You Only Look Once

Table 1: List of acronyms

