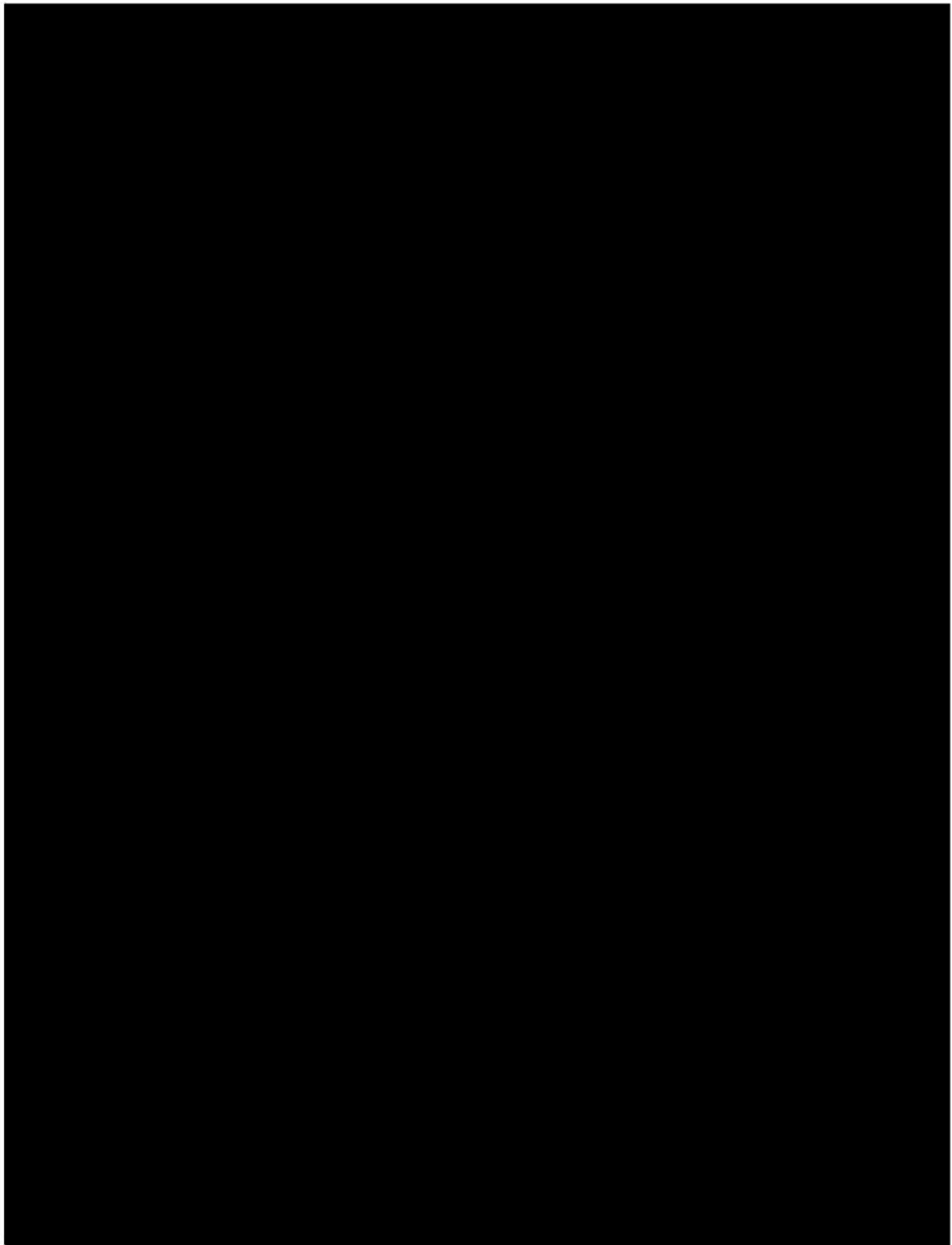
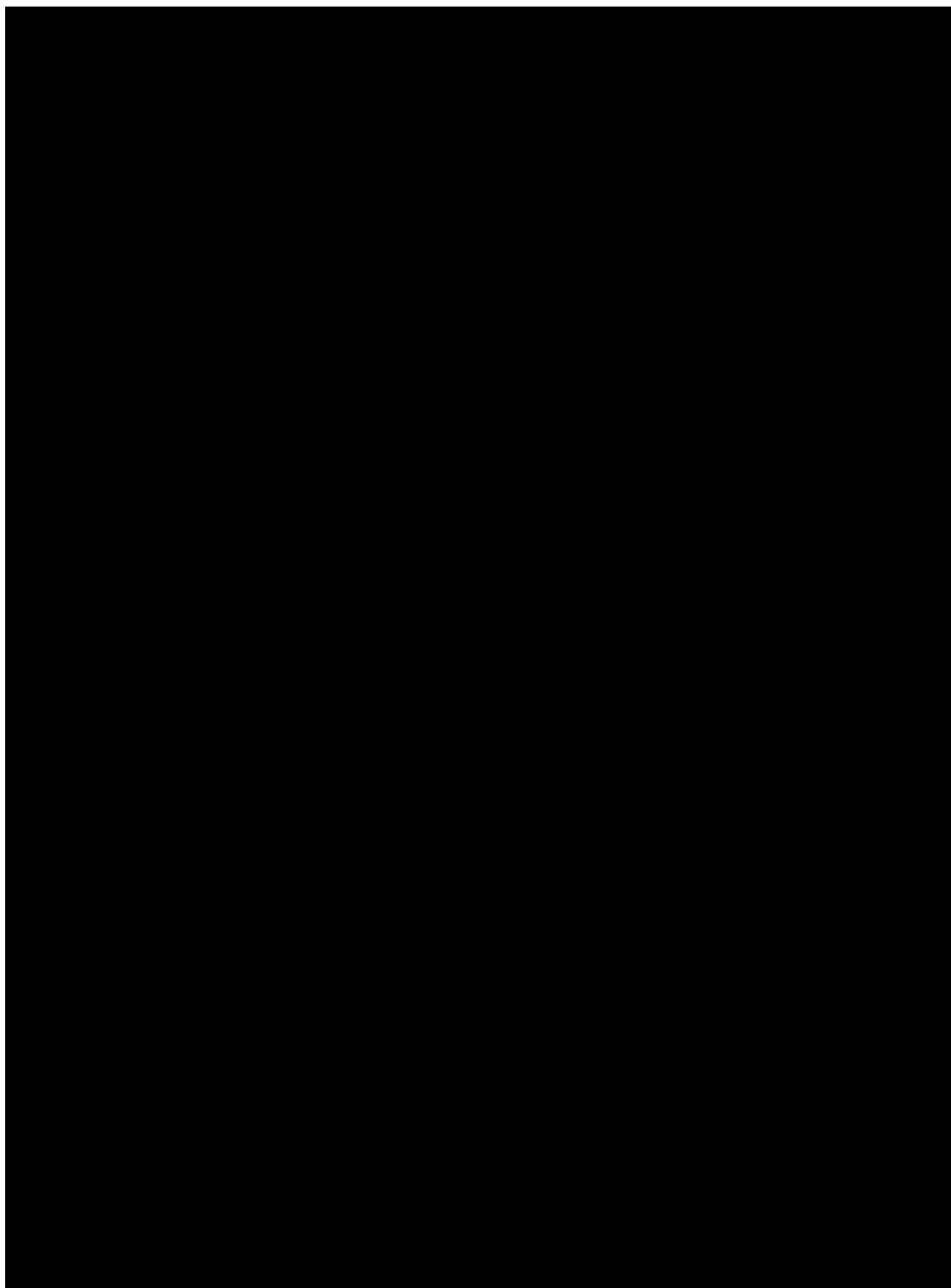








Agentic Artificial Intelligence for On-Board Spacecraft Image Processing

by
Jacob Davies BSc

Dissertation submitted in partial fulfilment for the Degree of Master of Science
in Applied Artificial Intelligence

WMG
University of Warwick

Supervisor: Dr Mayen Cunden

Declaration

I have read and understood the rules on cheating, plagiarism and appropriate referencing as outlined in my handbook and I declare that the work contained in this assignment is my own, unless otherwise acknowledged.

No substantial part of the work submitted here has also been submitted by me in other assessments for this or previous degree courses, and I acknowledge that if this has been done an appropriate reduction in the mark I might otherwise have received will be made.

The MSc Applied Artificial Intelligence project must involve an investigation of a relevant AI topic that demonstrates technical depth, meets academic standards, and addresses a well-defined research question in one or more core AI areas—such as machine learning, deep learning, natural language processing, computer vision, generative AI, or AI for industry and robotics. It must also show critical awareness of the ethical, societal, or deployment implications of AI technologies. This can be achieved through the design, implementation, and evaluation of a technical AI solution or through a systematic and critical literature review that synthesises current research and practice. Regardless of the approach, students must apply appropriate research methods, engage critically with existing literature, demonstrate depth of understanding, and justify the relevance and rigour of their work in line with academic standards.

This research aligns with the Applied Artificial Intelligence specialisation at WMG by developing a novel AI system that addresses a real-world engineering challenge and delivers tangible operational benefits to users.

Abstract

Earth Observation spacecraft, satellites which collect data about the Earth, capture more imagery than they can transmit. Access to this imagery is limited by an end user's ability to pay. Users who need very large areas or have limited budgets, i.e. small island nations wanting to track illegal fishing in their waters, are often unable to afford access to this data. By processing imagery onboard the spacecraft and transmitting only the useful data, the cost of surveying for these customers would be drastically reduced.

This research addresses how best to implement image data processing on edge hardware by simulating onboard data processing.

Following a design science approach, guided by an artefact-driven narrative literature review, several model architectures were trained on hand-annotated satellite imagery and then evaluated.

This research suggests that deploying a single shared backbone from which many user-defined models run is effective and would reduce data costs by spreading one expensive computation across many users.

Acknowledgements

I would like to thank my supervisor, Dr Mayen Cunden, for his guidance and his patience. I am much more comfortable building things than writing about them, and without his patience I would not have completed this project.

Table of contents

Project Submission Pro-Forma	II
Declaration	IV
Abstract.....	V
Acknowledgements	VI
Table of contents	VII
List of tables	X
List of figures	XI
Glossary	XII
Chapter 1: Introduction.....	1
1.1 Background and Context	1
1.2 Brief History of AI for EO.....	1
1.3 Research Aim.....	2
1.4 Research Question	2
1.5 Research Objectives.....	2
1.6 Research Scope	3
1.7 Nature of Research.....	3
1.8 Importance and Relevance of the Research	4
1.9 Structure of the Dissertation	4
Chapter 2: Methodology	5
2.1 Research Philosophy	5
2.2 Research Approach and Strategy	5
2.3 Research Design Overview	6
2.4 Dataset Selection and Pre-processing	7
2.5 Model Architecture and Training	8
2.6 Model evaluation	9
2.7 Ethical risks.....	10
2.8 Risks and mitigation	10
2.9 Summary	10
Chapter 3: Artefact-driven narrative literature review	11
3.1 Existing EO Imaging Processing Pipeline	11

3.2 Existing vessel detection systems	11
3.3 Problem identification.....	13
3.4 Spacecraft downlink constraints	13
3.5 Existing Onboard Spacecraft Processing and Processing Payloads	15
3.6 Requirements definition.....	16
3.7 AI Image Processing Architectures	16
3.8 CNNs for Image Processing	17
3.9 Vision Transformers for Image Processing	19
3.10 Self-distillation.....	20
3.11 Agentic AI.....	21
3.12 Edge Deployed Systems	22
3.13 Containerisation	23
3.14 Quantisation	23
3.15 Scaling laws	25
3.16 Artefact development.....	25
3.17 Vision Model Evaluation Approaches.....	26
3.18 Statistical analysis of vision model performance	27
3.19 System evaluation	27
3.20 Summary	28
Chapter 4: Data analyses	29
4.1 Analysis approach and success criteria.....	29
4.2 Model characterisation.....	29
4.3 Test data performance.....	31
4.4 Cloud cover.....	32
4.5 Resource cost	34
4.6 Comparison and trade-off	35
4.7 Summary	36
Chapter 5: Discussion.....	37
5.1 Answering the research question	37
5.2 Performance against prior work and the success criteria.....	37
5.3 Robustness	38
5.4 Successful economic implementation.....	38
5.5 Agentic decision making	39
5.6 Limitations and counterarguments.....	39
5.7 Summary	40

Chapter 6: Conclusion	41
6.1 Academic contribution.....	41
6.2 Industrial value	41
6.3 Further research	42
6.4 Final thoughts	42
References	43
Appendix A: Certificates	48
Appendix B: Ethical approval waiver	51
Appendix C: Planet Labs education and research program terms of service (signed)	52
Appendix D: Hyperparameter sweeps	57
Appendix D.1 Foundation CNN	57
Appendix D.2 Fine-tuned CNN	61
Appendix D.3 Fine-tuned ViT	65
Appendix D.4 Dinov3-L	70
Appendix D.5 Dino v3- 7B.....	74
Appendix E: Hyperparameter definitions.....	78
Appendix E.1 CNN.....	78
Appendix E.2 Transformers.....	80
Appendix F: GAIT	82
Appendix F.1 Claude	82
Appendix F.2 Google AI Overview.....	82
Appendix F.3 Grammarly	82
Appendix G: Word count	84

List of tables

Table 2.1 Evaluation metrics recorded per model across three weather conditions (clear, overcast, adverse)	9
Table 3.1 Reported detection performance across recent SAR and optical vessel detection studies: F1 reports instance F1, And mAP50 reports mean average precision when IoU is above 50%.....	12
Table 3.2 Onboard hardware of Phi-sat missions and Planet Labs operational fleet	16
Table 4.1 Success criteria, their thresholds, and threshold justification.....	29

List of figures

Figure 3.1 Diagram showing difference in overhead trajectory for two orbits with different altitudes and their transit time	14
Figure 3.2 One stage vs two-stage image detection	17
Figure 3.3 vertical vs horizontal edge kernels.....	18
Figure 3.4 Transformer architecture (Vaswani et al., 2017)	19
Figure 3.5 Quantisation of weight values – (a) calibrated range of ± 1 , (b) the same 16 levels after a single outlier stretches the range to ± 4	24
Figure 3.6 “Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.”(Kaplan et al., 2020) ..	25
Figure 4.1 CNN - (a) the Validation IoU with respect to number of channels in model (b) the Validation IoU with respect to number of channels in model for multiple seeds (c) Validation IoU with respect to amount of training data used.....	30
Figure 4.2 ResNet - (a) the Validation IoU with respect to number of parameters in model, (b) Validation IoU with respect to amount of training data used.....	30
Figure 4.3 ViT - (a) the Validation IoU with respect to number of parameters in model, (b) Validation IoU with respect to amount of training data used.....	31
Figure 4.4 Dinov3 backbone vs head performance	31
Figure 4.5 best performing models (of its type as per validation data) on test data – (a) metrics per masc pixel performance, (b) metrics per vessel	32
Figure 4.6 best performing models (of its type as per validation data) instance performance on test data across cloud coverage.....	32
Figure 4.7 Effect of cloud augmenting training data on validation instance f1 over 3 weather conditions	33
Figure 4.8 DINOv3-7B + object filter in 3 weather conditions for two scenes.	34
Figure 4.9 model performance on representative hardware as input size scales - (a) latency at scaled clock speed, (b) power consumption	35
Figure 4.10 Precision recall curves on vessel instance by weather type	35
Figure 4.11 test set metrics of models compared with threshold	36
Figure E.1 kernel size and dilation diagram.....	78
Figure E.2 Step size diagram.....	78
Figure E.3 width vs depth in CNN	79
Figure E.4 Padding in CNNs	80
Figure E.5 Patch embedding steps.....	81

Glossary

Term	Definition
Agentic AI	An AI system that perceives its environment, makes autonomous decisions, and takes actions in pursuit of a defined goal without requiring human input at each step.
ART-NLR	Artefact-driven Narrative Literature Review: a literature review method that stages literature engagement against the phases of design science research (problem identification, requirements definition, artefact development, and evaluation) rather than completing a single exhaustive review before empirical work begins.
Attention	A mechanism that computes, for each element of an input, a weighted combination of all other elements, allowing a model to capture relationships between distant parts of an input.
Automatic Identification System (AIS)	The transponder that ships use to broadcast where their location
Backbone	The main feature-extracting body of a model, it converts an image into a feature representation. A head is attached on top to produce the final prediction.
Bandwidth	The width of the transmitting window between the lowest and highest frequency transmitted (in Hz).
Bitrate	The number of Bits per second communicated.
Compute Unified Device Architecture (CUDA)	NVIDIA's software layer that lets code run on their GPUs.
Contact window	A period during which a spacecraft is within line-of-sight of a ground station and can communicate. Outside these windows the spacecraft cannot send data to Earth.
Convolutional Neural Network (CNN)	Convolutional Neural Network: a type of AI for images. It scans small patches across a picture, spotting edges first, then shapes, then whole objects.
CPU	A Central Processing Unit, CPUs have a small number of high performance computational cores and can run a wide variety of general tasks.
Deep Learning (DL)	An AI built from many stacked layers of artificial neurons.

DINO	Self-Distillation with NO labels: a self distillation model developed by Meta.
Downlink	The transmission of data from a spacecraft down to a ground station on Earth.
DSR	Design Science Research: a research methodology centred on the iterative development and evaluation of a designed artefact, typically structured around problem identification, requirements definition, artefact development, and evaluation.
Edge	The physical boundary between the network and the real world, where sensors, cameras, and instruments exist.
Edge computing	Performing computation directly on the device where data is generated, rather than transmitting it to a remote server or cloud for processing.
EMA	Exponential Moving Average: a smoothed average of a model's recent versions, used to give it a steadier teacher to learn from.
EO	Earth observation: the gathering of information about the Earth's physical, chemical, and biological systems.
ESA	European Space Agency: Europe's space agency, the European equivalent of NASA.
F1	The harmonic mean of precision and recall, giving a single accuracy figure that balances false positives and false negatives.
FLOP	Floating Point Operation: one piece of arithmetic. Counting them estimates how much computing a model needs, independently of the machine it runs on.
Foundation model	A large model completely trained from scratch, it can be adapted or fine-tuned for downstream tasks.
FP16 (also FP32)	16-bit (also 32-bit) floating point datatype, a way of storing each number in a model using half the usual precision.
FPGA	A field programmable gate array, they are integrated circuits that can be re arranged to perform small tasks with very low latency.
GPU	A Graphics Processing Unit, they have many small computing cores and are optimised for efficient matrix multiplication.
Ground station	A facility on the ground that connects to spacecraft.
Hyperparameters	External configuration variables, set before training a machine learning model to control its structure.

Inference	Running a trained model on new data to produce a prediction.
Information (in information theory)	A measurement information communicated (often in bits) that is equal to the number of binary yes or noes required to communicate said information.
IoU	Intersection over Union: a measure of overlap between a predicted region (e.g. a bounding box or mask) and the ground-truth region, calculated as the area of overlap divided by the area of their union.
IQR	Interquartile Range: the spread of the middle half of a set of results, ignoring the extremes.
Kernel	A small matrix of learnable weights that is convolved (slid) across an input image or feature map to detect a specific local feature such as an edge or texture.
LLM	A Large language model, well known examples include ChatGPT, Claude or Google Gemini.
Machine learning (ML)	Software that learns patterns from examples instead of being given rules.
mAP	Mean Average Precision: a standard object detection accuracy metric that averages precision across recall levels to give a single overall accuracy score.
Matrix	An array of numbers in two dimensions.
Neural network (NN)	A network of artificial neurons used to learn complex relationships between inputs and outputs.
Overfitting	When a model learns patterns specific to its training data, including noise, rather than generalisable features, resulting in strong performance on training data but poor performance on new data.
Precision	Of all the detections the model made, the proportion that were correct.
Quantisation	A technique that reduces the numerical precision used to represent a model's parameters (e.g. from 32-bit to 8-bit), reducing memory footprint and inference latency at some cost to accuracy.
Recall	Of all the objects present, the proportion the model found.
Reinforcement Learning (RL)	Training by trial and error, rewarding the system when it gets things right.

Self-distillation		Training a model using its own past or averaged weights as the teacher signal, rather than a separate larger teacher model.
Software Development Kit (SDK)	Kit	A bundle of tools and drivers that a manufacturer supplies for writing software for their hardware.
Spacecraft		A vehicle designed to operate in space.
TIMM		PyTorch Image Models: a public library of ready-trained image models that researchers can download and build on.
TPU		Tensor Processing Unit: Google's own AI chip, an alternative to a GPU. The Coral Edge TPU is the low-power version.
Vector		A list of numbers, they can represent a direction and magnitude in a multidimensional Euclidean space.
Vision Transformer (ViT)	Transformer	A transformer-based architecture applied to images by dividing them into fixed-size patches, embedding each as a vector, and processing the resulting sequence with self-attention.
VPU		A Visual Processing Unit, functionally a GPU, they have many small computing cores and are optimised for efficient matrix multiplication for visual processing.
Weight		A learnable parameter in a neural network that scales (multiplies) an input during forward propagation. Weights are what is adjusted during training to minimise the model's error.
YOLO		You Only Look Once: a well-known fast object detector. It finds everything in a picture in one pass rather than checking regions one at a time.

Chapter 1: Introduction

Earth observation spacecraft can observe far more of the planet than they can relay back to Earth. This chapter explains where that gap comes from, argues that the real problem is economic rather than technical, and sets out why engineering a perception system for an agent running on the spacecraft itself is a credible way to close it.

1.1 Background and Context

EO (Earth Observation) spacecraft are satellites designed to observe Earth from space. They collect data about land, water, atmosphere, and climate and are used for tasks including: weather monitoring, terrain mapping, environmental change tracking, agriculture, disaster response, and urban planning.

EO spacecraft do not have continuous high-bandwidth connectivity to the ground; instead, they downlink data only during limited contact windows (Furano *et al.*, 2020). The data collection capabilities of spacecraft far exceed their downlink capacity, so spacecraft operators prioritise high-demand imagery over land and areas of high economic value. Capturing imagery of large low-value areas such as open ocean is deprioritised, as transmitting this is both technically and economically unviable. This results in spacecraft operating significantly below their theoretical observational capacity. Despite being capable of capturing imagery across vast geographic regions, they remain idle during periods when it is not economically valuable to transmit imagery (Furano *et al.*, 2020; OECD, 2023). However, these areas are only low-value because the cost of gathering and transmitting imagery outweighs the return. If that cost could be reduced, the sheer scale of these regions could mean that the aggregate value of the data becomes worthwhile.

Advances in artificial intelligence create an opportunity to address this inefficiency (Furano *et al.*, 2020; Jet Propulsion Laboratory, 2025). An AI system capable of processing imagery onboard and then relaying a small volume of high-value results to the ground, without requiring additional hardware or putting a significant resource strain on the spacecraft, could fundamentally change this dynamic, freeing up downlink capacity and unlocking the full observational potential of existing spacecraft fleets (Furano *et al.*, 2020; Jet Propulsion Laboratory, 2025).

1.2 Brief History of AI for EO

Current EO systems operate using a simple workflow: spacecraft capture imagery, store the data onboard, and transmit it to ground stations when communication windows become available (when the spacecraft is in line of sight of a ground station and able to connect to it).

Historically, this has relied on ground-based processing architectures: spacecraft transmit the raw data to ground stations, where computationally intensive analysis is performed only after receipt, using high-performance computing infrastructure (Furano *et al.*, 2020).

As technology has improved, spacecraft have shrunk, and their cost has fallen drastically, but this introduces new power and transmission constraints. Modern imaging spacecraft therefore

can image vastly more than they can transmit back to ground (European Union Agency for the Space Programme, 2022; OECD, 2023).

Research has been conducted to filter out imagery unsuitable due to predefined issues, such as cloud cover, such as the ‘Cloud Scout’ project on Phi-Sat-1 (Giuffrida *et al.*, 2020, 2022). This allows modest gains in information transmitted; however, these approaches are narrowly scoped and do not significantly expand the spacecraft's underlying observational capability.

1.3 Research Aim

This research aims to investigate approaches to processing satellite imagery for an agentic artificial intelligence system capable of analysing spacecraft imagery in real time and autonomously processing observations for transmission. An agentic system would need to perceive what a satellite can see, reason about it and effect the satellite. Controlling spacecraft has been done for decades, and agent reasoning has attracted significant work to be surveyed (Wang, 2025); perception has attracted considerably less work and is the area where we can have the greatest effect.

The proposed system would operate directly onboard an imaging spacecraft. Rather than transmitting all captured data, the AI system would process imagery as it is collected and transmit the output of user-defined experiments. By using the spacecraft as an intelligent processing platform rather than a passive data collection device, it may become possible to monitor significantly larger regions while operating within existing communication constraints.

A key limitation on any edge-deployed (‘edge-deployed’ means it runs on the spacecraft's hardware) spacecraft system is the processing, power, and thermal constraints inherent to the platform. Any solution must not simply solve the detection problem but solve it on small, power-limited hardware without generating excess heat.

1.4 Research Question

How can image processing for a vision-based agentic system be effectively implemented to process large quantities of spacecraft imagery on the edge?

1.5 Research Objectives

To address this research question, the project will pursue these objectives:

1. To review how existing AI systems are implemented and to identify design lessons and requirements for the proposed vision-based system.
2. To determine what hardware, system inputs, and system outputs are available for an edge-deployed EO image processing system; by memory, power, and hardware budgets.
3. To design, build, and test five model implementations across three model architectures.
4. To evaluate what makes the implementation of such a system effective, according to the success criteria.

1.6 Research Scope

The research scope is limited to an onboard artificial intelligence system for real-time maritime vessel detection in Earth observation imagery, deployed on resource-constrained satellite hardware.

Case Boundaries:

- Domain: Maritime vessel detection
- Geographic: Maritime regions (oceans, waterways, and coastal waters)
- Operational context: Satellite-based observation with real-time edge processing constraints
- Technical scope: Vision-based vessel detection and prioritisation for downlink

What is included in this scope:

- System design with specified inputs, outputs, and operational constraints.
- Vision-based processing pipeline for vessel detection.
- Edge deployment on representative hardware under power/compute constraints.
- Performance evaluation across three weather conditions.

What is excluded:

- Post-detection analysis (vessel behaviour, tracking, classification)
- Data collection methods
- Extension to other domains (forestry, agriculture, land)
- Live spacecraft deployment
- Proprietary spacecraft hardware or avionics
- Agent reasoning, planning, and spacecraft tasking

1.7 Nature of Research

This project adopts a design science research (DSR) approach, combining machine learning experimentation with system prototyping and evaluating the resulting artefacts against success criteria derived from an Artefact-driven Narrative Literature Review (ART-NLR) (Trigo, 2026).

The research involves developing a computer vision model using spacecraft imagery datasets and evaluating it, followed by deploying the trained model on representative hardware platforms to determine the best implementation approach. The literature engagement follows ART-NLR (Trigo, 2026), which structures the review around the stages of design science research: problem identification, requirements definition, artefact development, and evaluation. This ensures that the literature review in Chapter 3 directly informs the model design decisions and the success criteria used in Chapter 4 to evaluate our models.

Experimental evaluation will be used to quantify performance by assessing model accuracy, precision, and recall, the model's computational performance, and subsequent suitability for real-time operation. This will demonstrate whether this approach is economically and technically advantageous compared to traditional methods of gaining this capability, such as increasing downlink throughput.

1.8 Importance and Relevance of the Research

Improving the effectiveness of EO spacecraft has significant implications for environmental monitoring, maritime security, and disaster response (European Union Agency for the Space Programme, 2022; OECD, 2023). Large areas of the planet remain difficult to observe due to limitations in spacecraft data transmission capacity.

By enabling spacecraft to process data pre-transmission, onboard AI systems could dramatically increase the amount of useful information collected. This would allow oceans to be monitored more efficiently (OECD, 2023) and reduce data-collection costs enough to make new customer groups viable.

1.9 Structure of the Dissertation

This dissertation follows a path from problem definition, through empirical investigation, to interpretive synthesis. The structure progressively narrows the research focus by first establishing the real-world problem and research scope, reviewing what is already known from prior work, conducting the research investigation, and interpreting the findings and their implications in the following chapters.

Chapter 1: establishes the problem domain, defines the research vision, and bounds the scope precisely to maritime monitoring, DSR problem identification.

Chapter 2: specifies the empirical approach. It describes the datasets, model architecture, training procedures, hardware platforms, and metrics for evaluation.

Chapter 3: conducts an ART-NLR, structured around the stages of design science research: problem definition, design requirements, artefact development, and evaluation, covering Earth observation systems, computer vision for spacecraft imagery, and edge AI deployment within that structure, DSR requirements definition.

Chapter 4: presents the model performance under different weather conditions, computational performance on edge hardware, and power consumption. The DSR artefact development and measurement section does not explain what the performance means.

Chapter 5: compares the empirical results against prior work and the requirements and evaluation criteria established through the ART-NLR, evaluates which design approaches work better and why, and assesses whether the original research question has been answered, enabling DSR evaluation.

Chapter 6: transforms the findings into practical recommendations and suggests how practitioners should approach similar problems.

Objectives 1 and 2 are achieved in chapter 3, objective 3 in chapters 2 and 4, and objective 4 in chapters 4 and 5.

Chapter 2: Methodology

Published literature on the behaviour and performance that a detection system onboard a satellite must achieve is limited, with commercial deployments kept as trade secrets and published platforms operating as demonstrators (Marin *et al.*, 2021; Giuffrida *et al.*, 2022), making a traditional literature review difficult. An artefact-driven narrative literature review (ART-NLR) was adopted to guide the development, training, and quantitative evaluation of a variety of vision-based models against literature-derived requirements. By defining the requirements for that development and allowing them to drive the research, ensuring our research remains relevant and addresses the limitations of a traditional LR.

2.1 Research Philosophy

The optimal trade-off between accuracy and resource use that fits within onboard hardware constraints given the required latency thresholds, power draw limits, and minimum recall, drawn from literature informs the research philosophy.

This research adopts a pragmatic philosophical position: knowledge is valued by what it enables. Rather than simply reasoning about which architecture should perform best, this research adopts a build-and-evaluate approach. The research builds, tests, and scores candidate model architectures against identified operational constraints, producing findings that are directly applicable to hardware selection and system design.

The research combines a mixed methodological approach. Quantitative data, such as model accuracy and inference speed, give objective performance measurements. The ART-NLR provides qualitative information required to judge whether those measurements are appropriate, drawing from published constraints on power, latency, and detection accuracy into the thresholds used to interpret the results.

2.2 Research Approach and Strategy

A conventional systematic literature review assumes a single search to answer a predefined question is completed before the empirical work begins. Instead, this research uses literature to inform decisions as they surface.

This research therefore adopts an ART-NLR (Trigo, 2026), which stages literature engagement against these design science research (DSR) activities: problem identification, requirements definition, artefact development, and evaluation. Each stage defines a question and a gap, searches for literature, evaluates the sources, and produces a critical evaluation. This allows the review to stay transparent at each point without the added difficulty of a single exhaustive review completed up front.

Alternatives to an ART-NLR were considered; simulation was rejected because an accurate representation requires deployment to the device and no satellite system was accessible. A survey was ruled out because the expert pool is small and documentation on the key variables is unavailable. A systematic literature review was ruled out as a monolithic search of the literature would spend a significant portion of time reviewing literature that would not be applicable to the artefact implementations.

This staged ART-NLR design translates into the following research structure.

The requirements stage of the ART-NLR surfaces the operational constraints not captured in the architecture-focused literature reviewed in this chapter 2. This includes acceptable inference latency thresholds, power budgets, and tolerance for detection error in maritime surveillance operations, drawn from mission reports, technical specifications, and published case studies.

The research is deductive in structure; it tests whether established computer vision architectures and commercially available edge hardware can satisfy constraints derived from literature on spacecraft compute budgets and downlink limitations, with adequate performance to make it useful.

The research strategy follows a design-and-evaluation structure. A system is built, tested on representative hardware, and evaluated against a defined success criterion. This distinguishes the study from a case study, as the focus is on the performance of a designed artefact under specified constraints, rather than on an organisation or event. This is consistent with a DSR strategy, in which the artefact itself is the unit of evaluation.

2.3 Research Design Overview

The study proceeds in four phases, aligned with the DSR activities the ART-NLR is staged with: problem identification, requirements definition, artefact development, and evaluation.

Firstly, the problem identification and requirements definition define the success criteria and engineering limitations from the end user and engineering limitations, respectively.

Next, the models will be developed, beginning with vessel instances from the satellite imagery being manually annotated by the researcher to create segmentation masks, and the annotated dataset is split 70:15:15 across training, validation, and test data. Five model variants are optimised and trained: a foundation CNN, a fine-tuned CNN, a fine-tuned Vision Transformer, and two head-classifier models built on top of frozen DINOv3-L and DINOv3 satellite 7B backbones.

The strongest of our models will be developed further to create a maximally competitive headline model. Hyperparameters are optimised by varying them individually via sweeps and evaluating performance on validation data. The comparison models methodology will also be reproduced on our data, to control for performance differences caused by differences in data and evaluate only the effect of differences in approach.

After which the models will be evaluated on the test data, and inference latency will be measured; throughput, power draw, memory footprint, accuracy, precision, recall, and F1 will be recorded across the three weather conditions.

Then these results will be compared against the success criteria to determine which design choices produce the best trade-off between accuracy and resource use, and to answer the research question. This is the evaluation activity in the DSR cycle: it is where the requirements

derived from Phases 1 and 2 and the artefacts built in 3 are brought together and judged against each other.

2.4 Dataset Selection and Pre-processing

The model will be trained and evaluated on images accessed through the Planet Labs Education and Research Program (Planet Labs PBC, 2024a), a satellite imagery dataset containing millions of satellite images. A subset of imagery will be manually annotated to create corresponding training masks. The data must represent the conditions the system will encounter in operation, including open ocean, coastal waters, and varying weather states, so that a realistic assessment of model performance can be made. The Planet Labs education and research program is open to university-affiliated researchers and access is granted on application with a university email. The preprocessing pipeline, model architecture, hyperparameters, and evaluation criteria are fully documented so that a researcher with equivalent access could replicate the study. Access is through a program Planet Labs designed specifically for academic use, so its use here is consistent with the program's terms of service.

The pre-processed dataset will contain over 100,000 km² of satellite imagery at a 3m resolution from Planet Labs imagery, and is supplied through its education and research program, this imagery captures real operational data from an active commercial constellation (Planet Labs PBC, 2024c). These are hand annotated by the researcher with a mask to outline the vessels. The training, validation, and test split follows a 70:15:15 split, with the test partition held out until final evaluation to prevent data leakage. Hand annotation by a single person has the risk of producing inaccurate and noisy data, and it is not industry standard. However, it is the approach taken as it is a time intensive task and bringing in extra resources for this falls outside the scope of this dissertation, therefore slightly lower-accuracy annotation will have to be accepted.

Three weather conditions, clear, overcast, and adverse (<15% cloud cover, 15-50% cloud cover, >50% cloud cover), are selected to reflect the operational range of a maritime surveillance system. Test sets for each condition are drawn from held-out imagery matched to those conditions. Reporting performance across conditions assesses the model's operational range rather than its peak accuracy under ideal conditions. Due to the annotation strategy, there is a risk of high error rates in the data in cloudy conditions (if a vessel is concealed by clouds, it cannot be seen, and it cannot be labelled). Therefore, overcast and adverse conditions will be created by taking images annotated in clear conditions and adding clouds to the desired level of cloud cover.

Fixed train, validation, and test splits maintain validity as the same data is used to train validate and test models. The success criteria were frozen before testing began, so the results reflect genuine differences, not data leakage or bias optimisation. Additionally, data leakage is prevented by constraining frames from the same image to the same set; the three weather conditions and the mix of coastal and open-ocean scenes support generalisation across operational contexts.

2.5 Model Architecture and Training

We train and compare multiple model variants on the annotated dataset. Evaluating architectures against the same data and hardware allows architecture evaluation on the accuracy-efficiency trade-off. The following model architectures will be trained and evaluated, and the model with the best performance will be developed further. The external comparison models used for our comparison will be reproduced on our data, hyperparameters will be defined by the paper (Magalhães, Falcão and Barbosa, 2025).

Foundation CNN: A convolutional neural network trained on the satellite data. This is important, showing the performance achievable with a foundation model, which is important as it allows complete control over all model parameters. The primary focus here should be evaluating how different design constraints effect performance, and projecting how performance will change if compute, model size, and data increases, to inform how a real deployment may work.

Fine-tuned CNN: pre-trained CNNs from the Resnet family, will be fine-tuned end-to-end on the satellite data. This tests whether general-domain visual features transfer to satellite imagery and provides a comparison for the ViTs as ViTs will be finetuned.

Fine-tuned ViT: A pre-trained Vision Transformers fine-tuned on satellite data. While they typically require more data and compute to train than an equivalently sized CNN, their long-range modelling capability may be useful for distinguishing vessels from clutter in complex coastal scenes. They have proven to have improved performance over CNNs, but this can come at the cost of dramatically larger parameter counts.

DINOv3-L with detection head: The (dinov3-vitl16-pretrain-sat493m) backbone is a generalised model that produces contextualised embeddings from satellite data; a lightweight detection head is built based on these contextualised embeddings. This model is distilled from DINOv3(7B). An evaluation on how best to build a head for this model and how quantisation affects performance will be conducted.

Satellite DINOv3 (7B) with detection head: same as DINOv3-L but using the much larger dinov3-vit7b16-pretrain-sat493m backbone. This model achieved the highest performance and was developed further using an objectness filter to improve precision and F1.

Hyperparameters are optimised through a hyperparameter sweep search, with validation IoU used to select the best configuration for each model. Early stopping is applied when validation loss plateaus for 5 epochs. All five models use the same augmentation pipeline (random horizontal flipping, rotation, scale jitter, and colour augmentation) differences in performance reflect architecture rather than training setup (The reproduced comparator uses the augmentation from (Magalhães, Falcão and Barbosa, 2025)).

2.6 Model evaluation

Each model is evaluated on the images held for test data, measuring the metrics below in every test.

Metric	Measurement	Rationale
Precision	True positives / (true positives + false positives)	Measures the false alarm rate; a high-precision model rarely flags open ocean as a vessel
Recall	True positives / (true positives + false negatives)	Measures the miss rate; a high-recall model rarely fails to detect a vessel that is present
F1	Harmonic mean of precision and recall	Captures the accuracy-recall trade-off in a single figure for cross-model comparison
Mask IoU	Pixel overlap between predicted mask and ground truth mask, divided by the total pixels covered by either	Measures whether the model correctly delineates the vessel boundary, not just its location
Inference latency	Milliseconds per frame	Must fall below the threshold needed to keep pace with the satellite's image capture rate
Throughput	Frames per second	Derived from latency; used to compare against capture rate directly
Peak memory	Megabytes	Must fit within the memory available on the evaluation platform

Table 2.1 Evaluation metrics recorded per model across three weather conditions (clear, overcast, adverse)

For each metric: mean, median, standard deviation, IQR, minimum, and maximum are calculated per model per weather condition. This makes variation across conditions visible. Reporting per condition rather than averaging across them, as a model that performs well under clear conditions but degrades sharply in adverse weather is less operationally useful than one with more consistent performance across conditions, even if its peak accuracy is higher.

The five models are compared using these success criteria as a structured pass/fail framework. Among models that pass all identified thresholds, the accuracy-efficiency trade-off is examined by comparing performance against computational cost across different weather conditions. This identifies whether the accuracy gain from a larger model justifies the additional resource demand, and which architecture offers the most practical choice for deployment on constrained spacecraft hardware.

These different metrics will be evaluated together against the success criteria set out to establish which models are the most appropriate for real-world data.

The larger models are expected to be more accurate, but also more expensive; a critical evaluation will be conducted to identify the point at which the accuracy gain from a larger model no longer justifies the additional cost. A single metric cannot determine this; therefore, a critical analysis will be conducted to evaluate the model's real-world net value to a user.

2.7 Ethical risks

All data used was secondary satellite imagery with a ground sample distance of approximately 3 metres, at which resolution an individual cannot be resolved or identified; no personally identifiable data was collected, analysed or stored at any point; and no human participants were involved. However, there is a substantial ethical risk in the possible duality of purpose; automated vessel detection does help combat illegal fishing, but it could also be used for military surveillance. Considering that the theoretical capability this system creates does already exist, although by a different approach, this risk is judged to be acceptable because the project has no military funding or collaboration, no system was deployed operationally, and no detections were shared with any third party.

2.8 Risks and mitigation

Originally, this project planned to use a series of interviews to inform the model success criteria. Unfortunately, the interviews did not go ahead, and the original risk mitigation of pivoting to a literature-derived set of requirements was not implemented, and an artefact-driven narrative literature review more strongly guided the research.

There are still risks with this approach, the largest of which is that there are no published examples of our use case being implemented. There is therefore a reliance on non-users to derive user requirements, which risks reinforcing poor assumptions and producing research that is not useful to us. To mitigate this we compare our comparison papers against other papers with similar objectives and ensure that the targets for our models are similar to many other papers.

Additionally, there is a risk with using hand annotation from a single annotator that the annotation could produce incorrect data. To mitigate this effect, the data annotated has low cloud cover and clouds are artificially added.

2.9 Summary

This chapter specified the method through which all four learning objectives will be achieved: ART-NLR staging, artefact build, and evaluation protocol. Chapter 3 will conduct the ART-NLR, identifying the problem, defining requirements, developing an artefact, and the evaluation criteria; deriving candidate architectures and success criteria from literature.

Chapter 3: Artefact-driven narrative literature review

This chapter focuses on the literature used to form an ART-NLR (artefact-driven narrative literature review) through 4 stages:

1. Identifying and defining the problem
2. Defining the requirements needed for a solution to be successfully deployed
3. Evaluating the literature for developing the Artefact, including architectures and implementation constraints
4. Concluding by setting out objective evaluation criteria.

A large issue dealt with is the lack of published literature for specific issues; answers are at times derived from grey literature or inferred from similar examples.

3.1 Existing EO Imaging Processing Pipeline

We begin by synthesising ideas from a referenced review of relevant source material for processing EO images. Across satellite image providers, the general process is similar; for simplicity, we use Planet Labs as the reference for specific details.

The imaging processing pipeline starts as follows: a customer places a request for imagery of a specified area, possibly in a specified time window or at a specified time. They may also specify special requirements for spatial and spectral resolutions, and cloud cover, and negotiate a price. This is often done alongside hundreds, if not thousands, of image requests. This is inputted into a database alongside all other requests the satellite provider has received from their customers. It is then fed into a scheduler which decides the most optimal imaging schedule for each spacecraft in the constellation and updates the satellite's internal schedule.

Then, at the specified time, the spacecraft will orient itself towards the area to be imaged and capture imagery. This is then stored internally until the spacecraft is in line of sight of a ground station, where it downlinks as much imagery as possible while overhead. This imagery is then handed to the customer.

The bottleneck in this system is that high-resolution cameras onboard the spacecraft can image almost constantly, but the downlink capacity allows only a fraction of imagery “Currently, only an estimated 2% of recorded satellite data reaches Earth” (SatNews, 2026). This specific number should be treated as indicative, it comes from a company that is raising money to solve what they refer to as the “2% problem”, however even if it is fractionally truthful it would result in the deletion of hundreds of terabytes of useful observations per day (SatNews, 2026).

3.2 Existing vessel detection systems

Currently, there are two approaches generally used for vessel detection: Automatic Identification System (AIS)-based systems that rely on AIS and non-AIS-based systems that rely on imagery or radar. Non-AIS are generally more costly and less accurate, but do not require the vessel to actively cooperate.

There is a difference between identification and detection; identification labels a vessel type/activity given that it is known to be a vessel, whereas detection establishes that a vessel is present.

AIS is highly accurate at identifying a vessel's location. Identification systems can use AIS data to identify whether a vessel is fishing and achieve >90% accuracy (Kroodtsma *et al.*, 2018); however, recall is worse because vessels can turn their AIS transponders off. (Paolo *et al.*, 2024) found that over 75% of the world's industrial fishing vessels do not appear in public AIS data; AIS-based systems can at most expect to achieve 25% recall. The focus will be on non-AIS detection. Below are the stated performance figures of 4 systems identified in the relevant prior study material:

Study	Sensor	Precision	Recall	Other reported metrics
(Paolo <i>et al.</i> , 2022)	SAR (Sentinel-1)	not separately reported	not separately reported	F1 = 0.77 overall, F1 = 0.53 close to shore
(Paolo <i>et al.</i> , 2024)	SAR (Sentinel-1) + some optical	not separately reported	not separately reported	>97% object detection accuracy
(Mäyrä <i>et al.</i> , 2025)	Optical (Sentinel-2)	87.6%	87.4%	mAP50 $\approx$ 0.85
(Magalhães, Falcão and Barbosa, 2025)	Optical (Sentinel-2)	90.47%	86.98%	F1 = 88.69%, mAP50 = 87.53%

Table 3.1 Reported detection performance across recent SAR and optical vessel detection studies: F1 reports instance F1, And mAP50 reports mean average precision when IoU is above 50%

All four studies note performance drops near coastlines and for smaller vessels. Inconsistent benchmarks, labelling conventions, and IoU thresholds add further difficulty.

Ideally, thresholds would be derived from stakeholder requirements; however, there is a lack of literature or documentation on recall and precision targets needed for business operations (the studies in 3.1 report only achieved metrics not success thresholds). To address this, a supply-side comparison is performed instead by looking at what published systems have achieved and treating this as indicative of real-world performance requirements. (Magalhães, Falcão and Barbosa, 2025) is selected as a comparative analysis with our data, given the precise reporting on performance metrics and equivalent sensor modality.

(Magalhães, Falcão and Barbosa, 2025) will be used as the threshold to evaluate our model. One issue with this is that the performance metrics reported are not solely due to their approach, they are the product of their approach trained and tested on their data. By implementing their approach on our data, we will still not have an objective figure, but compared in the context of other models trained on the same data, fair comparisons can be made.

The studies above report mAP at a fixed IoU threshold; this assumes that a prediction's precise boundary affects downstream performance. Any similarity between a predicted location and a vessel's location is equally useful, since a vessel may move kilometres between detection and interception. The performance metrics need to reflect this, and not punish models that are off by just a few meters.

3.3 Problem identification

There is a clear gap in the literature regarding published numbers on what a system would need to achieve to be useful, as this is a new technology that has a limited end-user market. We observe similar systems, and requirements are derived from them.

Imaging capacity far exceeds downlink capacity. The systems reviewed run on ground hardware post-downlink, so detection reaches only a small fraction of the imagery that makes it through this bottleneck.

There is inconsistency in how these metrics are measured and inaccuracy in reported numbers. It is also unclear what ground truth the reported metrics are basing their numbers on. (Magalhães, Falcão and Barbosa, 2025) is used as a benchmark for its strong performance and clear, transparent metric reporting and the fact that it is consistent in its performance goals with other comparative models.

3.4 Spacecraft downlink constraints

This section covers why there is a bandwidth bottleneck, what would be required to remove it, and why working around it is preferable.

There are two factors that determine the throughput: the contact window (amount of time that satellites can downlink) and the bitrate of the downlinks (how much data per second can be transmitted). Together, these two factors determine the throughput of the overall system.

Spacecraft can only downlink to ground stations when they are in line of sight of the ground station, and the ground station can connect to the satellite. Ensuring a ground station has the capacity to connect to a satellite is trivial, a more important issue is the amount of time a spacecraft has line of sight to a ground station. This is determined by two main factors: the orbit of the satellite, and the positioning of ground stations.

Increasing the number of ground stations is, in theory, a trivial development. However, in practice it is difficult. It only makes sense to build a ground station if it can see satellites and those satellites aren't already transmitting to an existing ground station. In practice, this means there are very few places where it is both economical and useful to set up a functioning ground station. While it might be useful to have a ground station in the middle of the ocean, it would be very impractical and expensive to build. A ground station near a major equatorial city would be well connected to both spacecraft and the Internet, but most of these positions are already saturated, and few satellites flying overhead won't already be transmitting to existing ground stations.

Given that it's difficult to increase the number of ground stations, a reasonable alternative is to increase the time a spacecraft is in line-of-sight to a ground station. Operating a spacecraft at a higher altitude can increase the window. This gives the spacecraft a wider horizon, allowing it to see more of the Earth at a given point and, consequently, spend longer in line-of-sight of ground stations. However, being further from the planet limits the maximum achievable resolution for a given lens or mirror size (the Rayleigh Criterion states that the maximum angular resolution of a system is $1.22 \cdot \frac{\lambda}{D}$, where λ is the wavelength of light and D is the

diameter of the optics) (Born, Wolf, and Avadh Behari Bhatia, 2016), devaluing the image quality and making it more expensive to put the same-weight satellite at a higher altitude.

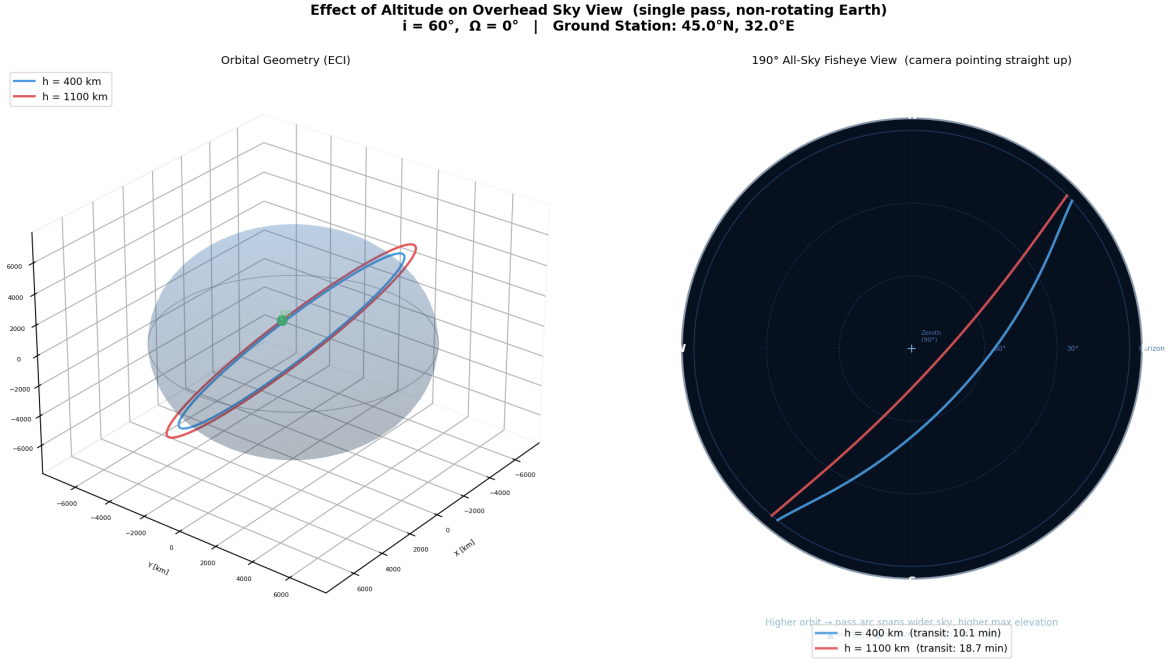


Figure 3.1 Diagram showing difference in overhead trajectory for two orbits with different altitudes and their transit time

Alternatively, the throughput of the downlinks could be increased; this would mean that more data could be transmitted in the same period of time. There is a complex relationship between transmitter power, operating frequency, bandwidth, and background noise. The Shannon-Hartley Theorem dictates the maximum bitrate you can transmit dictated by the equation:

$$C = B \cdot \log_2(1 + SNR)$$

$$SNR = \frac{P_t G_t G_r}{LN}$$

Where *C* is bitrate (bits of information per second), *B* is bandwidth (Hz between highest and lowest frequency available), *SNR* is the signal to noise ratio, *P_t* is the transmitter power, *G_t* and *G_r* are the gain of the transmitter and receiver (efficiency), *L* is the Path loss (drop in the transmissions power density), and *N* is the Noise figure (a measure of *SNR* degradation) (Shannon, 1948).

Bandwidth cannot be increased without purchasing more spectrum and worsening path loss dramatically. The power of the transmitter is limited by the spacecraft's power budget, transmitter gain is limited by the antenna design, and path loss and noise figure are not controllable. The two approaches explored that could increase throughput of a system, transmitting more quickly and transmitting for longer, run up against practical and physical limitations. Hence, increasing a spacecraft's capabilities requires a new approach that is able to do more with less.

3.5 Existing Onboard Spacecraft Processing and Processing Payloads

Onboard spacecraft processing long predates modern AI approaches. How they have been used, what changes have been made in recent years, and the hardware utilised on modern systems will be presented in this section.

To begin, let's consider compression algorithms; these reduce the size of transmitted data while retaining the amount of information communicated by compressing data before transmission and decompressing on the ground. They have proven to be effective and are almost universally used; however, the effectiveness of compression algorithms is approaching a fundamental limit. Information theory (Shannon, 1948) shows a fundamental maximum amount of compression is possible (determined by entropy, or randomness), reducing the amount of data transmitted further requires reducing the amount of information communicated.

A new approach to this is using AI to reduce the amount of information transmitted without reducing its utility. A significant recent advancement was made with the ESA (European Space Agency) Phi-sat-1 mission in 2020, the first satellite known to carry a dedicated onboard AI payload specifically designed for real-time image filtering. It runs a deep neural network called 'CloudScout' which detects and discards cloud-covered images before downlink, demonstrating that AI-based inference was viable on operational hardware and increasing the amount of useful information gathered by cleaning the data before transmission (Giuffrida *et al.*, 2020, 2022).

The Phi-sat-2 mission, launched in 2024, extended this concept with a more flexible architecture that allowed remote upgradability, meaning operators could install or update models post-launch, this allowed the satellite to be repurposed for new tasks without hardware modification (Marin *et al.*, 2021). This change from fixed-function payloads toward reprogrammable systems allows the use of more novel unproven systems to be tested.

A major enabler of these systems is the onboard hardware; the amount of hardware can vary dramatically between spacecraft. Below is an overview of the onboard hardware of the two Phi-sat spacecraft and the 4 operational or planned spacecraft in the Planet Labs fleet (Owl is planned to enter operation). The table is assembled from institutional documentation, manufacturer datasheets and trade reporting and allows us to clearly see what hardware is available and what is required for any model we are developing.

	ESA		Planet Labs			
Hardware	Phi-sat-1	Phi-sat-2	Super Dove	Owl	Pelican	Tanager
Onboard processing	Back-End Electronics BEE board and an on-board data-handling (OBDR) computer	includes a BEE, CPU, GPU, FPGA and mass-memory units	Always-on main on-board computer, An undisclosed CPU and an undisclosed FPGA	Undisclosed CPU and FPGA, with a Jetson Thor GPU	Always-on main on-board computer, 12-core Arm Cortex-A78AE CPU and NVIDIA Jetson AGX Orin a CPU and an undisclosed FPGA	Undisclosed CPU, GPU, and FPGA
AI specific processing	Intel Movidius Myriad 2 vision processing unit (VPU)	Intel Movidius Myriad 2 VPU	None	NVIDIA IGX Jetson Thor (2560-core GPU and 96 Tensor Cores)	NVIDIA Jetson AGX Orin (2048-core Ampere GPU with 64 Tensor Cores), NVIDIA IGX Jetson Thor also tested (2560-core GPU and 96 Tensor Cores)	Undisclosed
Memory	Undisclosed	Undisclosed	Undisclosed	128 GB	64 GB	Undisclosed
Storage	Undisclosed	Undisclosed	Undisclosed	< 2TB	Undisclosed	< 2TB
Transmitter(s) (peak data rate)	S-band transceiver and an X-band transmitter for high-rate data downlink (uplink $\sim 4 \text{ kb s}^{-1}$, downlink $\sim 32 \text{ kb s}^{-1}$)	S-band transceiver and high-rate X-band downlink (up to $\sim 160 \text{ Mbit s}^{-1}$)	UHF band telemetry transmitter ($\sim 3600 \text{ bit s}^{-1}$) X-band downlink (up to 1.8 Gb s^{-1})	S-band telemetry (16 kb s^{-1}), an X-band TT&C radio (64 kb s^{-1}), an S-band high-speed uplink (512 kb s^{-1}), and a Ka-band downlink (10 Gb s^{-1})		

Table 3.2 Onboard hardware of Phi-sat missions and Planet Labs operational fleet

(Ahumada, Doerksen and Zeller, 2021; Devaraj, 2022; Giuffrida *et al.*, 2022; NVIDIA, 2022; Planet Labs PBC, 2024b, 2024d, 2026a, 2026b; Jewett, 2025; Ouellette, 2025; Wang, 2025; European Space Agency, 2026)

3.6 Requirements definition

Downlink bandwidth cannot be economically increased. To increase the amount of useful information communicated, a workaround for that constraint is needed. This approach addresses this by processing data onboard and drastically reducing the amount of data transmitted.

While there is onboard processing for imaging-specific tasks (cloud filtering), there is a lack of user-defined use cases. A system must be reprogrammable, utilise the available high-performance hardware (Jetson Orin (NVIDIA, 2022)) and work around downlink limitations. Satellite manufacturers won't be willing to sacrifice their primary proven business operations.

3.7 AI Image Processing Architectures

AI image processing applies machine learning techniques to analyse, interpret, and manipulate visual data, enabling systems to perform tasks such as object detection (identifying a vessel in the water), classification (classifying the type of vessel), and segmentation (creating a mask of what parts of an image are vessels and what parts of an image are water). There are trade-offs between performance, latency, and compute. A larger model is better but more computationally demanding, running it on less hardware makes it slower, latency can be decreased by reducing the size. It is also possible to keep performance nearly identical but reduce the model size by

reducing precision of its parameters instead, quantization represents numbers using less accuracy for a reduction in memory footprint and inference latency. The correct balance of these factors is determined by the constraints of the system and the performance requirements of the models these unknowns need to be constrained to optimise a solution.

One trade off often used to reduce compute for similar or the same performance is two stage systems. A two-stage system firstly uses a lightweight, high recall, model to draw bounding boxes for possible objects before using a larger high accuracy model to identify if and what the objects are. This cuts down on the compute as the expensive model is only being used on a fraction of the overall image but comes at the cost of the latency of having to run consecutive models. Recently models such as YOLO (You Only Look Once) have shown exemplary performance for identification tasks, this points us towards this approach of this detection task.

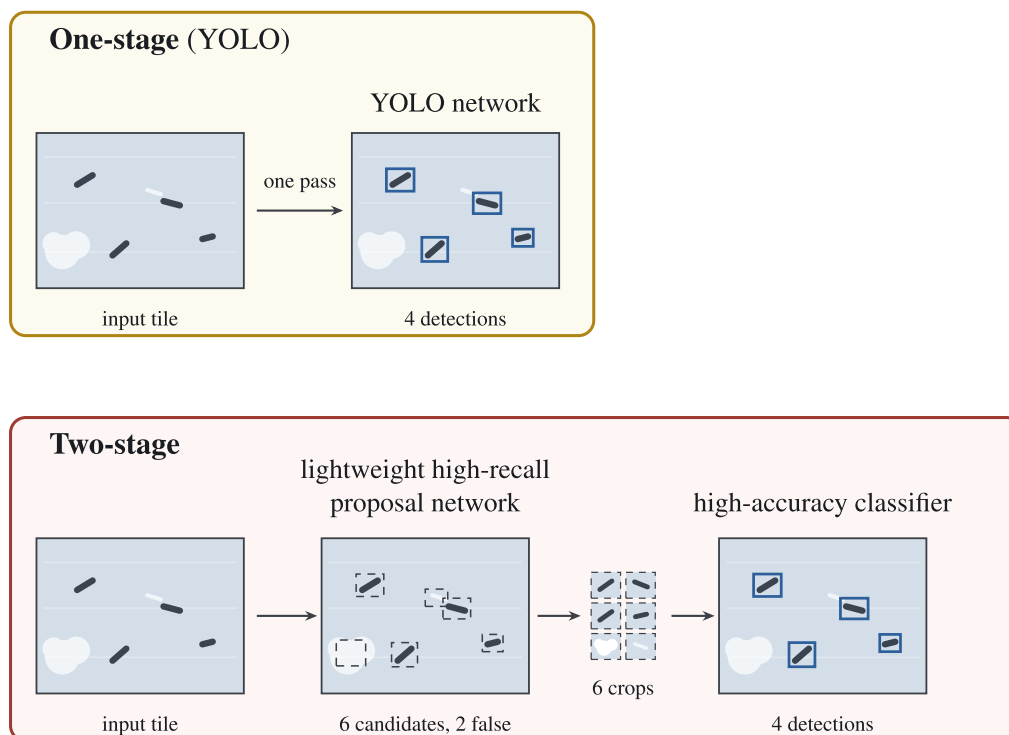


Figure 3.2 One stage vs two-stage image detection

Historically fully connected Neural networks have been used for basic image processing, while useful for very specific controlled data (character recognition) it performed poorly at tasks where there is large variability in inputs, making it not useful for this use case. A major improvement came with the adoption of Convolutional Neural Networks (CNNs). This was improved on by transformer-based systems which allows even more complex data to be accurately processed. This analysis will compare these two architectures and evaluate their applicability to our research question.

3.8 CNNs for Image Processing

Convolutional Neural Networks (CNNs) are deep learning models built from convolutional layers, which process structured input data such as images (Lecun *et al.*, 1998). This section provides an overview of CNNs, covering their functionality and reviewing examples of CNNs.

CNN-specific terminology:

- Inputs, typically images, are matrices (grayscale) or tensors (multy(3)-dimensional matrices) (full colour has one matrix for red, one for green and one for blue), where each matrix element corresponds to pixel brightness.
- A kernel or kernel function is a matrix of weights (weights are numbers that are multiplied by an input to produce an output).
- A convolution is a process where a kernel function is flipped (typically not shown in diagrams for simplicity) and shifted across an image, multiplying the input elements by kernel weights as it moves.

CNNs learn features in data via convolutional operations. A simple example of this is identifying vertical or horizontal gradients. Below you can see the prescience of vertical and horizontal gradients as defined through two convolutions.

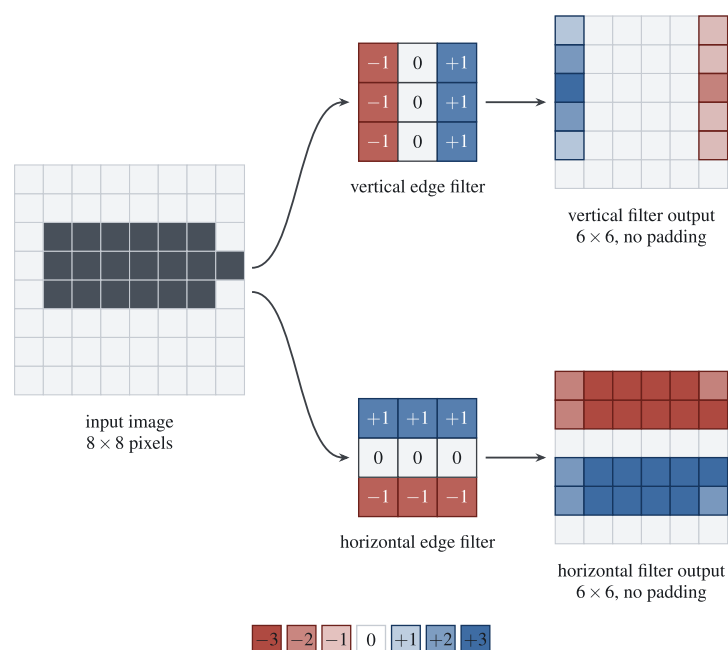


Figure 3.3 vertical vs horizontal edge kernels

In each layer of a CNN, a kernel is scanned across the data via convolution, producing ‘pool layers’. Pool layers highlight where filter outputs are strong across the data. Filters are learned during training and capture features that most affect the output. You can feed these pooled outputs into new layers to repeat the process. The same process can be repeated to analyse more complex features and structures.

CNNs have several hyperparameters that can be tuned to change a model's behaviour to fit a specific use case. The hyperparameters that were found to be optimal for the models trained, and what the hyperparameters change about the model can be found in Appendix D.1, D.2, and E.1.

Their use in image processing is well documented, covering a broad set of uses, including masking, object detection, and classification. CNNs have been the dominant computer vision architecture since 2012 with AlexNet for large-scale image classification (Krizhevsky, Sutskever and Hinton, 2012). The architecture has proven to perform well on a variety of tasks,

and have been iteratively improved; VGGNet, ResNet, DenseNet, MobileNet, and EfficientNet (Simonyan and Zisserman, 2014; He *et al.*, 2016; Howard *et al.*, 2017; Huang *et al.*, 2017; Tan and Le, 2019). The MobileNet and EfficientNet families specialise deployment on resource-constrained hardware, demonstrating viability of CNN's for our resource constrained requirements.

The literature shows CNNs are highly versatile for computer vision, effective in various tasks, and can deliver good results with small models. The hardware used by satellites can run a CNN, although more testing is needed to find the right model size for desired performance.

3.9 Vision Transformers for Image Processing

Transformers are a DL NN (Deep learning neural network) architecture originally designed for contextualising words and next-word prediction; they were first proposed in 'Attention is all you need' (Vaswani *et al.*, 2017). Transformers were originally developed for language processing and have most visibly been used for LLMs; they have proven to work well with other uses such as time-series modelling, protein and image processing (Dosovitskiy *et al.*, 2021).

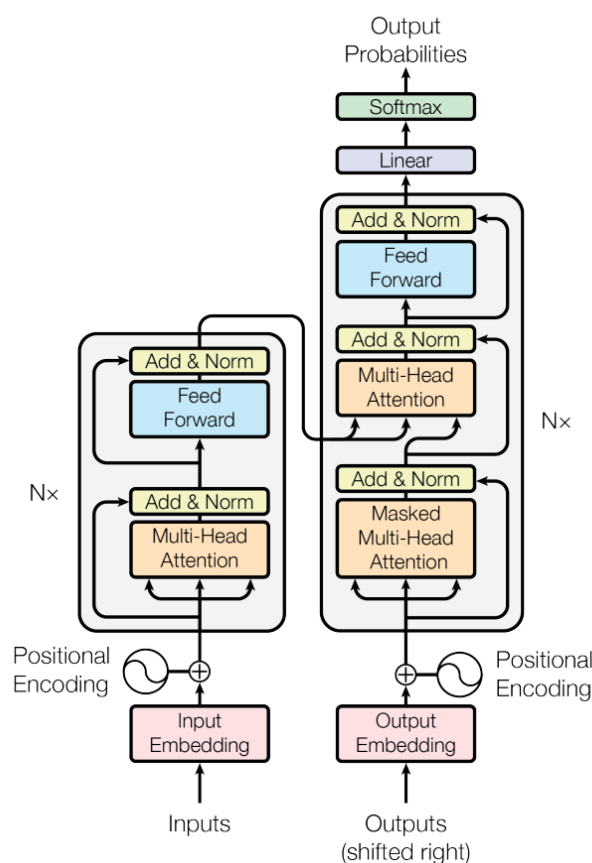


Figure 3.4 Transformer architecture (Vaswani et al., 2017)

Transformers use self-attention, which allows every part of an input to be modelled in relation to every other part (encoder-decoder models also have cross-attention, which compares parts of one input to parts of another). This allows the model to build both local and long-range dependencies across the underlying data, unlike CNNs, which can only compare local changes. Transformers operate on sets of vectors. In classical language modelling, these vectors

represent the semantic meanings of words and sub-words. However, in vision applications, they represent patches of an image. A diagrammatic description of how patch embedding systems work is available in Appendix E.2.

There are 3 types of transformers: an encoder, which turns inputs into vectors; a decoder, which turns inputs into one output (the next word); and an encoder-decoder, which decodes inputs based on context from a secondary input.

Transformer models have multiple sequential layers; each layer feeds an input into an attention block, then feeds the attention block output into a fully connected network. The attention block learns 3 sets of weights for each head: key, query, and value. Multiple heads allow the model to process multiple relationships. Attention blocks calculate attention scores between tokens and all other tokens. This lets the model judge the degree to which different regions relate to one another. ‘Attention score’ represents how much parts of the input should affect an output, based on the dot product of the query and key vectors.

Query vectors represent what a token is looking for, key vectors represent what information a token has, and value vectors are the information at that vector; the output value is weighted by the attention scores.

Transformers, unlike CNNs, do not intuitively understand structure, they treat input data as an unordered set. Positional encoding resolves this issue for each input vector, it encodes the vector's position and allows the model to understand positional relationships between vectors.

Transformer models have several hyperparameters that can be changed to adjust the model's behaviour for a specific use case. The hyperparameters that were found to be best for the models trained, and what the hyperparameters change about the model can be found in Appendix D.3 and E.2.

Transformers can produce better performance compared to CNNs, especially when identifying features that span large distances. This comes at a training and computational cost; transformer models can get exceptionally large, and it may not be the best-performing system if its size is overly constrained.

3.10 Self-distillation

Self-distillation is a knowledge distillation technique wherein a model learns from itself, or from an earlier version of itself. Traditionally, in knowledge distillation, a teacher is used to train a student and the training data labels (Hinton, Vinyals and Dean, 2015). Self-distillation does this in a single training run, removing the need to train a teacher beforehand, there are a few approaches to this system.

One technique is ‘data-augmentation-based self-distillation’ by ‘Xu and Liu (2019)”; two ‘perturbed views’ of the input are passed through a model, and a consistency loss system trains the model to output similar predictions.

‘Temporal self-distillation’, or the ‘Mean Teacher framework’ (Tarvainen and Valpola, 2017), uses an exponential moving average (EMA) of the model's past weights to act as the teacher for the current training step. originally proposed as a semi-supervised method rather than self-

distillation, but using an EMA of past weights as the teacher is the same system used in ‘temporal self-distillation’.

There are others such as ‘Layer-wise self-distillation’ (Zhang *et al.*, 2019), which adds ‘exit branches’ at several depths in the network. These branches are trained to match the final layer’s predictions so that the network’s deeper layers teach its shallower ones more complex relationships.

DINO, ‘self-Distillation with NO labels’ (Caron *et al.*, 2021), combines the first and second techniques (data-augmentation-based self-distillation and temporal self-distillation techniques). A student network and a teacher network share the same architecture, with the teacher’s weights set by an EMA of the student’s. Multiple augmentations of training images at different scales are passed between the student and the teacher, and the student learns to match the teacher’s outputs. The teacher’s outputs are centred and sharpened to prevent the model from learning to match predictions by always predicting the same thing. DINO uses no labels; the output features are learned from the structure of the training images. Since the model outputs patch embeddings, it’s easy for users to train a system to identify and classify images: measure the similarity between output patches and a few known examples.

One downside of self-distillation is that there’s a risk the model reinforces its own errors, since an incorrect prediction can become part of its own training data. Some research mitigates this by blending in a small proportion of ground-truth supervision alongside the self-generated soft labels (Zhang *et al.*, 2019), but these strategies are not adopted by the models considered here.

3.11 Agentic AI

Agentic AI refers to a set of autonomous systems that can perceive their environment, reason, plan, and execute sequences of actions with minimal or no human intervention. (Masterman *et al.*, 2024) Classifies agentic architectures into three core components: reasoning, planning, and execution. These systems can have differing levels of autonomy, from basic reactive systems to agents that can maintain an internal model of the world and reason about future consequences.

The recent deployment of agentic AI has been accelerated by advances in LLMs which have demonstrated good performance in multi-step reasoning, tool use, and instruction following (Wang *et al.*, 2024). However, LLM-based agents are not the only example of agentic systems; policy networks, rule-based planners, or RL (reinforcement learning) decision procedures can unlike LLMs operate on resource-constrained deployment environments and operate within the power and memory budgets of embedded hardware (Chien *et al.*, 2005).

Agentic behaviour offers a new range of capabilities for EO spacecraft. A spacecraft operating under the traditional ground-centric model has no capacity to adapt its behaviour based on what it observes, because latency drastically reduces throughput. An agentic system can autonomously adjust imaging priorities based on real-time detections, increasing image frequency over an area with noteworthy maritime activity, or deprioritising imagery collection over areas with heavy cloud cover.

Despite the growing literature on both edge AI for spacecraft and agentic AI, there is a lack of literature on the crossover of these two domains. Current onboard AI systems on Phi-sat-1 and

Phi-sat-2 spacecraft utilise inference pipelines: classifying or filtering imagery according to a fixed model, and have no ability to adapt their strategy in response to new information. The design of a system that merges real-time image processing with autonomous satellite behaviour on constrained hardware has not been addressed adequately in the literature. This gap constitutes the research problem this dissertation seeks to address.

3.12 Edge Deployed Systems

Edge computing places computation at or near the source of data generation rather than at a centralised facility (in the cloud). Understanding these systems and how they are built and maintained is central to this research, as it carries over directly to spacecraft. To understand these systems, we will explore the hardware that these systems run on, and the limitations placed on it and how that differs from non-edge-deployed systems, followed by a look into how software is deployed to these platforms.

Hardware purpose-built for edge AI has seen massive improvements over the past decade, and it has matured to the point where useful ML models can be run locally.

NVIDIA's 'Jetson' family is the most popular family of platforms for edge computing, the Jetson Nano, released in 2019 contains a 128-core Maxwell GPU with a quad-core ARM CPU on a single board drawing between five and ten watts. The newer AGX Orin, released in 2022, has a 2048-core Ampere GPU within a 15 to 60 watt range. Both run NVIDIA's 'JetPack' software development kit (SDK). These features allow the deployment of large and capable models.

The Intel Movidius Myriad 2 is an alternative system developed by Intel and used in Phi-sat-1. It has an order-of-magnitude lower power draw at just 0.5W. Google's Coral Edge TPU is another popular low wattage system that runs at just 2 watts.

Deploying software to these devices is more challenging than server deployment. One of the problems is that updates of individual components are not possible, and an update of firmware may often require a parallel software update because packages cannot be updated individually. Jetson boards ship with JetPack versions tied to specific Linux kernel, CUDA versions, and cuDNN builds. This presents a risk during deployment on an edge device, as a failed update can take a unit offline, possibly permanently. Containerisation can alleviate this issue: A Docker image bundles the exact libraries, framework versions, and application code the inference pipeline requires, which prevents mismatches of software, firmware and hardware versions, at the cost of modularity, requiring an update of the whole system to add new packages.

NVIDIA's container toolkit is distinct from other containerisation structures; as it exposes the host GPU to a running container without requiring the container to include GPU drivers, the same image can run on any device with a compatible host driver. When a new image is loaded to the device, the runtime replaces the previous container automatically and reverts automatically if the new container fails its health check. This allows a fleet with hardware variation to operate agnostically to the hardware differences, allowing operators to maintain one image per JetPack version rather than one build per device.

3.13 Containerisation

Containers are isolated environments that maintain their own filesystem, process tree, and network interfaces, unlike virtual machines, which emulate an entire hardware stack and boot a full guest operating system. This makes them faster to start and cheaper to run while still isolating dependencies from the host.

Docker is a tool used for containerisation. Docker packages an application and its full dependency tree into a container image (a read-only, layered filesystem-built instruction by instruction from a Docker file).

Containers solve reproducibility problems, deploying an application to a new machine often requires manually resolving dependency conflicts, a problem colloquially known as ‘dependency hell’. As container images carry their full dependency graph, the same image runs identically on all hardware.

Containers also simplify over-the-air updates, as a new image replaces the old running container atomically, and if the new container fails to start it is able to automatically roll back, limiting downtime and failures.

One issue with using Docker containers for ML deployments is the size of the weights of large models, because the model weights must be bundled into the container image, and a large model can be several gigabytes or even terabytes. Every new run produces an entirely new image, which must be uploaded to the device in full. There are a few ways this can be addressed.

The most direct approach is to reduce the size of the weights themselves through quantization, before it is ever packaged into an image. This shrinks subsequent transfers, whether a full image or a delta update, without needing any change to how the container itself is deployed.

Also, weights could be stored as an external volume rather than baking them into the image. A weight update transmits only the new weight file, then the container restarts without having to pull a full image.

A further improvement can be made by transmitting only the changed bytes between two weight files, a ‘delta update’; this is especially useful for parameter-efficient fine-tuning-driven updates, as they can modify a small proportion of parameters, so the update is a fraction of the full file.

3.14 Quantisation

Quantisation reduces the numerical precision used to represent a model's weights and activations; fewer bits per value requires less memory and less compute are used for operations. This is important for edge-deployed applications because edge hardware typically has less memory bandwidth and a lower power budget.

One approach for effective quantisation is post-training quantisation (PTQ), which converts an already-trained FP32 model to lower-precision with no retraining; it is fast and easy to apply but at the cost of accuracy. Alternatively, quantisation-aware training (QAT) simulates low-precision arithmetic during training, allowing a much lower drop in accuracy at the cost of extra training.

Regardless of which approach is used, the underlying operation is the same: a range of values is divided into ‘levels’ based on bit width, weights are replaced with the nearest level. The difference between a value and the level it is rounded to is the quantisation error, and because the number of levels is fixed, a wider range results in coarser spacing and a larger error.

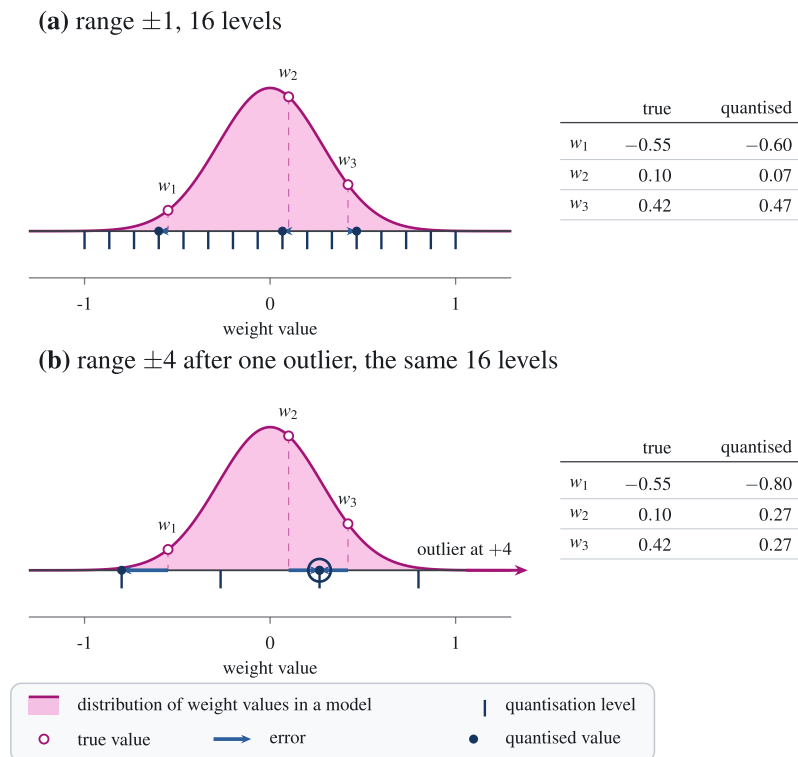


Figure 3.5 Quantisation of weight values – (a) calibrated range of ± 1 , (b) the same 16 levels after a single outlier stretches the range to ± 4

Dynamic quantisation needs no calibration data and quantises activations at runtime. Static quantisation precomputes calibration ranges and runs faster at inference. Mixed-precision quantisation assigns different bit-widths to different layers, such as keeping a ViT's attention layers at higher precision than its convolutions.

NVIDIA's ‘mixed-precision training’ (Micikevicius *et al.*, 2018) stores and multiplies values in FP16 but accumulates results in FP32. Almost all the accuracy loss under FP16 is caused by rounding error, individual multiplications contribute very little. So this recovers close to full FP32 accuracy at roughly half the memory footprint. NVIDIA's has implemented this natively in hardware since 2017, including on the Jetson-class hardware.

The accuracy drop from quantisation is more prevalent on ViTs than CNNs; any implementation of quantisation requires adequate testing beforehand.

3.15 Scaling laws

The relationship between a model's size, the volume of data it is trained on, and its resulting performance has been shown to be predictable. (Kaplan *et al.*, 2020) trained and tested over a hundred models and found that test loss falls as a power law for the number of model parameters, the size of the training dataset, and the total compute used for training. The practical consequence is that performance can be forecast, and how performance would improve with more resources can be predicted.

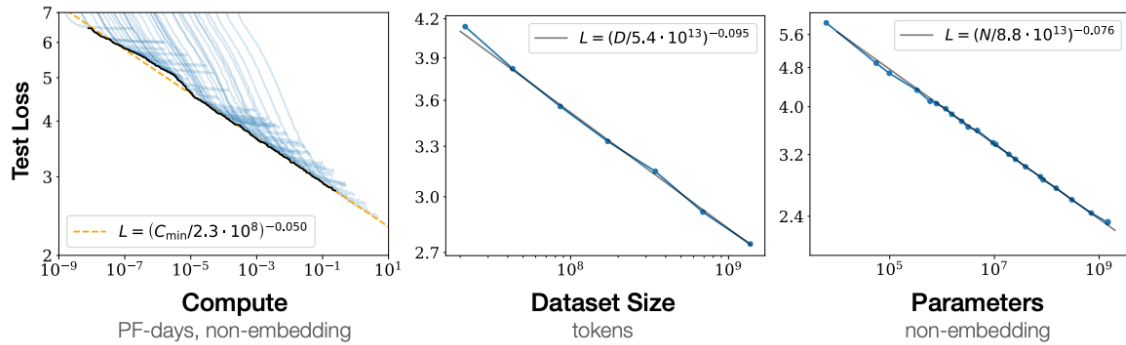


Figure 3.6 “Language modeling performance improves smoothly as we increase the model size, dataset size [sic], and amount of compute used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.” (Kaplan *et al.*, 2020)

These laws were established on language models, but research by (Sun *et al.*, 2017) shows that performance on vision tasks, also increases logarithmically with the volume of training data, (Zhai *et al.*, 2022) extended this testing to Vision Transformers.

The dominant controlling factor on model performance is not just model size, (Hoffmann *et al.*, 2022) showed that model size and training data should be scaled in equal measure, and that a smaller model trained on more data can outperform a larger one trained on less. Adding parameters yields little unless the dataset grows to match, which motivates treating performance as a function of dataset size and not model size alone.

This literature argues that, to be informative, we should compare models across a range of sizes, including the 0.3B and 7B DINOv3 variants, to investigate how compute cost and performance scale up or down. This supports training each model on increasing fractions of the dataset to observe how quickly it improves, and derive future performance.

3.16 Artefact development

While there are general trends in the literature pointing towards which model architectures perform better, they are not concrete. There is a need to practically test models and also look at how they may perform in the future.

The artefact is a set of five models representing a variety of approaches to this problem, along with data on how their performance scales with model size and training data. From the

literature, a deduction can be made on which approaches are known to be effective; they will then be implemented with our exact use case.

The five models are: a custom CNN trained from scratch, a fine-tuned ResNet, a fine-tuned TIMM Vision Transformer, DINOv3 ViT-L (satellite), and DINOv3 ViT-7B (satellite).

One model family (ResNet/TIMM ViT) is used for each approach. This cannot fully represent an entire approach, as another model could perform better, this is acceptable as our objective is to identify the best approach to building such a system, not to produce the single best-performing model.

DINOv3 ViT-7B (satellite) achieved the best performance, so it was developed further; an observation made was that recall was much higher than precision, and it was holding back performance. This issue was alleviated through the addition and development of an ‘objectness’ filter that decides if the predicted area is really a vessel, significantly improving precision for a small drop in recall.

3.17 Vision Model Evaluation Approaches

The evaluation of vision models has largely been standardised in literature by a few benchmarks. The PASCAL VOC challenge used mean average precision (mAP) as the standard accuracy measure for object detection, calculated from the precision-recall curve at an intersection over union (IoU) threshold of 0.5 (Everingham *et al.*, 2010). The COCO benchmark extended this by averaging mAP over IoU thresholds from 0.5 to 0.95, which rewards tighter localisation, and evaluation covers segmentation masks as well as bounding boxes (Lin *et al.*, 2014).

Segmentation quality is measured with the Dice coefficient, which scores the overlap between the predicted and ground-truth masks, and because it is computed per image it can be averaged across the test set and resampled the same way.

Precision, recall, and F1 remain in wide use alongside mAP because they allow false alarm rate and miss rate to be evaluated separately. This makes integrating the performance of a model in terms of its real world effects easier.

In terms of inference behaviour, computational costs can be estimated through parameter counts and FLOPs (floating point operations) (Howard *et al.*, 2017; Tan and Le, 2019), but predictors of latency are less reliable, because memory access patterns and hardware support have a major effect on performance.

Benchmark test sets are typically drawn from the same distribution as the training data, so headline scores risk overstating performance under real operating variation. (Hendrycks and Dietterich, 2019) showed that model accuracy often degrades sharply under common corruptions such as noise, blur, and weather effects, and argue for reporting performance across corruption types rather than as a single average.

Evaluating vision models should consider three things: accuracy, resource cost (measured on equivalent target hardware), and consistency across operating conditions.

3.18 Statistical analysis of vision model performance

A brief overview of literature shows that metrics are analysed inconsistently by different sources, the dominant practice is to report a single headline score per model from a single training run. Bouthillier et al. (2021) shows this is unreliable as retraining the same architecture with a different random seed, data ordering, or train-test split can shift metrics by benchmark scores by more than the gap between competing models. Picard (2021) demonstrates the same effect on standard vision benchmarks, finding that seed choice alone can move results by more than a percentage point. A comparison based on single runs can therefore not be treated fairly.

For detection and segmentation metrics such as mAP, which are computed over a whole test set rather than per sample, resampling the test set with replacement produces an empirical confidence interval around any metric without distributional assumptions (Efron and Tibshirani, 1993).

3.19 System evaluation

There is no clear way to analyse model performance, and the literature often use varying metrics depending on the model and its requirements. As a result, there is considerable flexibility in the design of statistical analysis. In m, an evaluation effort that more closely aligns with business requirements may be possible, but that would require these models to be deployed within the business case being researched.

For our approach, reporting metrics per model and per weather condition, , allows us to reason about both the model’s performance and its consistency. Evaluation of the models is conducted on three axes: performance, robustness across operating conditions, and resource cost on the target hardware. Vessels cover only about 0.016% of pixels, which creates a class imbalance.

Performance is measured at two levels. Pixel metrics (IoU, Dice, precision and recall) describe the quality of the segmentation masks, and instance metrics record whether each vessel is detected at all, using confusion matrices. The instance-level detection score is used as the headline result, as it directly corresponds to the end-user use case.

Robustness is measured by computing these metrics for each model across 3 different cloud conditions.

Resource cost is measured with an edge-compute benchmark approximating an NVIDIA Jetson AGX Orin, recording latency, power and memory against the defined hardware budget. The benchmark runs on a desktop GPU rather than flight hardware, so its figures are reported as a proxy.

3.20 Summary

This chapter reviewed the literature across the four ART-NLR stages. Problem identification identified downlink is the limiting factor in data collection, and other approaches such as AIS detection have unchangeably low recall. Programmability, edge hardware, and minimal disruption to primary tasks are requirements any solution must meet. Artefact development found that literature trends are indicative, so designs must be tested empirically. Evaluation found that instance metrics across conditions align with business needs and so are to be used.

Chapter 4: Data analyses

This chapter presents how the five models performed on the test data across the three weather conditions, compared with our comparison model, and examines how the models perform on the edge-compute benchmark. They are then measured against the threshold success criteria.

4.1 Analysis approach and success criteria

The five models are judged on three criteria (determined in section 3.19): performance, robustness across the weather conditions, and resource cost on edge hardware equivalent. Instance-level detection is used as the headline result, because it records the direct effects on the end user. Pixel metrics (IoU, Dice, precision and recall) are used secondarily to describe how well each mask outlines the vessel and tune hyperparameters. The mean average precision (mAP) is not used because vessels cover only about 0.016%, meaning that close predictions with small overlaps may be genuinely useful but not reach an IoU threshold.

The success criteria are treated as an indicative bar. The scores from (Magalhães, Falcão and Barbosa, 2025) use a fixed IoU threshold, but ours do not as we consider a prediction close to a vessel as useful, so the comparison is approximate.

Criterion	Threshold	Source / rationale
Detection F1 (instance)	88.7%	Headline accuracy bar. Score reached by on comparable (Magalhães, Falcão and Barbosa, 2025) Sentinel-2 imagery, used as an indicative comparator.
Detection precision	90.5%	(Magalhães, Falcão and Barbosa, 2025)
Detection recall	87.0%	(Magalhães, Falcão and Barbosa, 2025)
Inference latency	14 ms	14 ms per 512x512 tile represents 100% throughput at 100% resolution
Peak memory	64 GB	Jetson AGX Orin unified memory

Table 4.1 Success criteria, their thresholds, and threshold justification

Because published figures reflect the combination of a method and a dataset rather than the method alone, (Magalhães, Falcão and Barbosa, 2025) configuration was reproduced on this dataset. This reproduction will not be used as the threshold but instead used to contextualise our findings by providing a comparison to an optimised model from a peer reviewed paper.

4.2 Model characterisation

Each model family is characterised on validation data to see how it improves as model size and data size scale and to see the best configuration and hyperparameters carried forward. Seventy-two training runs were used to establish these. The reproduction is excluded from this, its hyperparameters are fixed by (Magalhães, Falcão and Barbosa, 2025).

The foundation CNN performed poorly compared with the other architectures. Validation IoU remained between 0.44 and 0.49. Larger models improved performance only marginally, and seed variation showed high variance by a similar margin. It did, however, improve quickly as

more data was introduced: at 5% of training data, its IoU fell to 0.25, and when it was given all the training data, it scaled to 0.485 IoU. This suggests that, while this model performed poorly, its performance would improve with additional training data.

Figure 4.2.1 From-scratch CNN

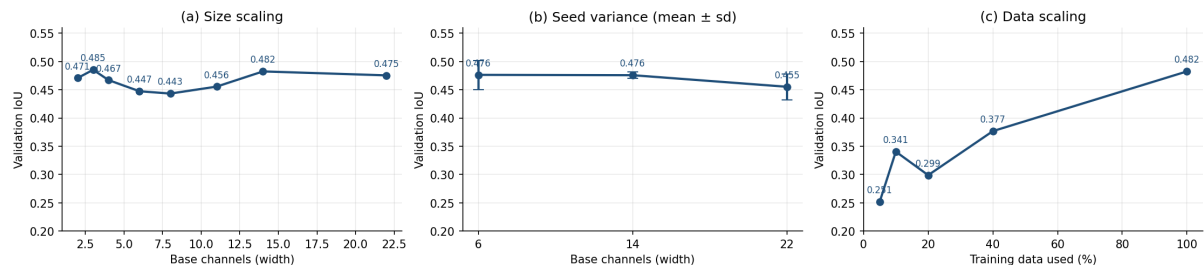


Figure 4.1 CNN - (a) the Validation IoU with respect to number of channels in model (b) the Validation IoU with respect to number of channels in model for multiple seeds (c) Validation IoU with respect to amount of training data used

The ResNet backbones perform better. Fine-tuning outperforms freezing the backbone. The fine-tuned ResNet-152 performed best, achieving an IoU of 0.517. Performance was good even with smaller fractions of the training data; however, performance improved more slowly as more data was introduced, suggesting that further improvements would be more costly than with the foundation CNN.

Figure 4.2.2 ResNet backbones

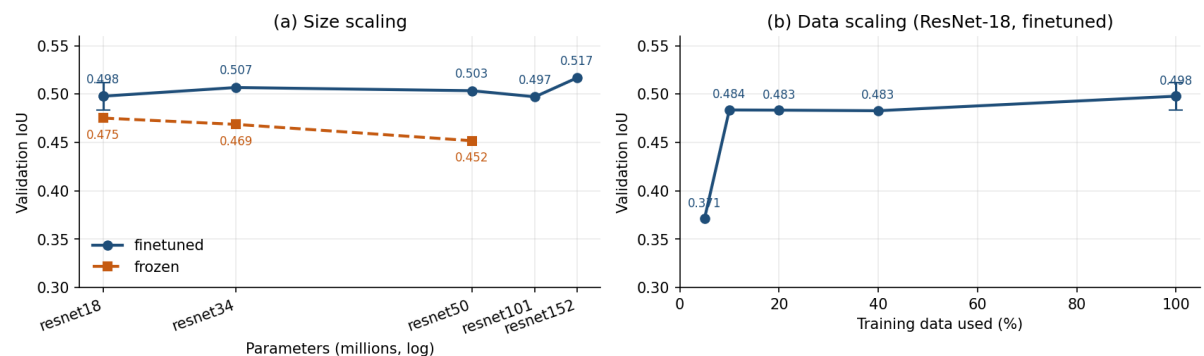


Figure 4.2 ResNet - (a) the Validation IoU with respect to number of parameters in model, (b) dation IoU with respect to amount of training data used

The ViT backbones behaved similarly, fine-tuning conferred a larger performance advantage, with frozen ViTs scoring 0.02 to 0.16 IoU lower than their fine-tuned counterparts. The largest ViT overfit and performed worse than the smaller models on test data; fine-tuned ViT-small performed best and matched ResNet-152 at 0.517 IoU. Performance remained reasonable at smaller data fractions and improved only slowly as more data was added, so further gains would likely be more costly to obtain.

Figure 4.2.3 ViT backbones

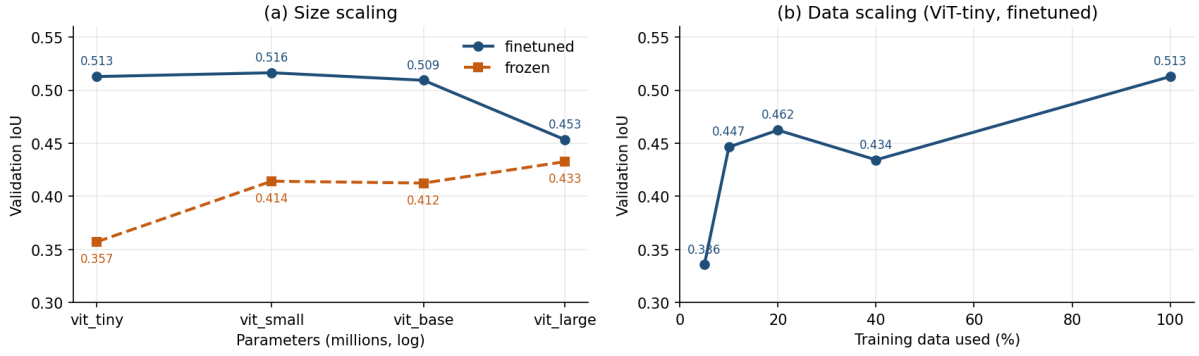


Figure 4.3 ViT - (a) the Validation IoU with respect to number of parameters in model, (b) Validation IoU with respect to amount of training data used

The frozen DINOv3 backbones were evaluated using only trained heads, with the best achieving 0.483 IoU on the large backbone. A single linear layer was sufficient to detect vessels, achieving 0.453 IoU. Very poor performance was observed in low-data runs and with cosine heads. Quantising the backbone to int8 or int4 changed the score by less than 0.01 IoU. The small gain from the large backbone to the 7B suggests that lighter models, if they were created, would likely see a small drop in performance.

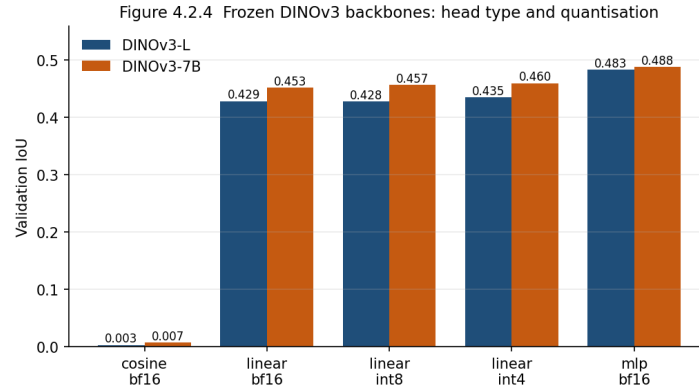


Figure 4.4 DINOv3 backbone vs head performance

4.3 Test data performance

From the previous section, hyperparameters can be selected that produce the best performance on the validation data. Those models can now be tested on the test data, giving us an objective measure of performance, without choosing models that perform well only on the test data.

Figure 4.3.1 Clear test-set performance (best configuration per model)

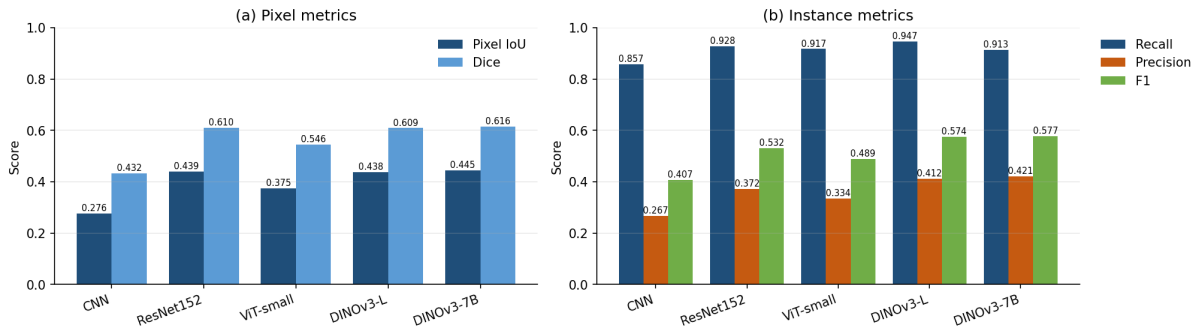


Figure 4.5 best performing models (of its type as per validation data) on test data – (a) metrics per masc pixel performance, (b) metrics per vessel

On clear test data at a probability threshold of 0.5, all models detect more than 85% of vessels but raise many false alarms: recall is between 0.86 and 0.95, precision is 0.27-0.42. Precision is weaker and has a negative effect on F1. The two frozen DINOv3 backbones lead at 0.577 and 0.574 F1, prompting us to further develop these models, followed by ResNet-152 at 0.532, ViT-small at 0.489, and the CNN at 0.407. The DINOv3 advantage comes from higher precision rather than recall, which suggests their features separate vessels from background clutter more reliably. Even so, every model falls well short of the 0.887 F1 threshold, so clear conditions alone do not yet produce a deployable detector.

4.4 Cloud cover

How robust our models are as cloud cover increases needs to be measured. Each model is evaluated on the three test conditions: clear, overcast, and adverse. Figure 4.6 shows the performance on the three weather conditions of the best models trained on clear weather.

Figure 4.4.1 Instance F1 across weather conditions (selected cloud_prob per model)

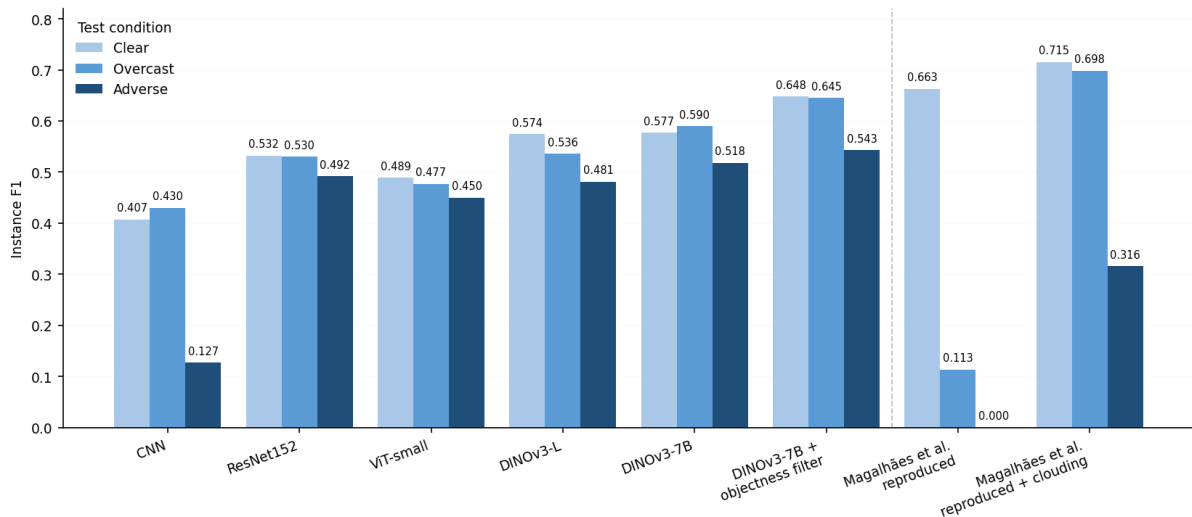


Figure 4.6 best performing models (of its type as per validation data) instance performance on test data across cloud coverage

The (Magalhães, Falcão and Barbosa, 2025) reproduction improves when we use our technique of adding clouds to some training frames even on test frames with no added clouds; this was also observed with our models. This is interesting because these models have fewer examples of clear images during training, so the expectation would be that they perform worse on clear images during testing. It seems that forcing the models to identify ships in difficult circumstances measurably improves performance in easy circumstances.

Below is instance F1 against the probability that a training frame had overcast or adverse clouds added to it in training for our models.

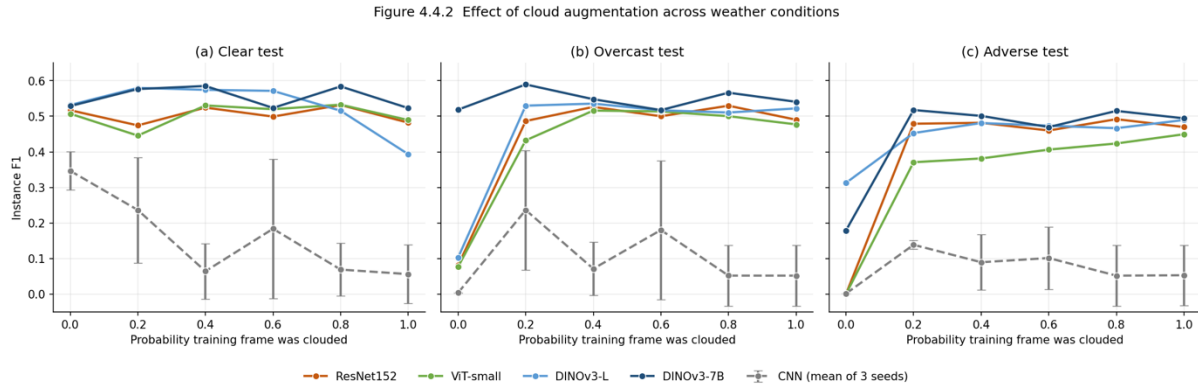


Figure 4.7 Effect of cloud augmenting training data on validation instance f1 over 3 weather conditions

The introduction of a small number of clouds in the training data significantly improves the performance of the models on overcast and adverse conditions with minimal effect on their performance on clear conditions. Introducing further clouded frames has very little effect, with slight negative trends being observed at 80% and 100%. The exception to this is the for foundation CNN, which performs better on adverse and overcast frames with some added clouds in the training data rather than none but quickly deteriorates, and clearly worsens its performance on clear frames with any clouded frames in the training data. The performance of the CNN is highly variable, so an average of 3 seeds is shown.

Below is a figure showing the prediction of the DINOv3-7B + objectness filter model in 3 weather conditions for two scenes, showing that while there is a drop in performance for clouded frames, it is remarkably resilient when trained on similar situations.

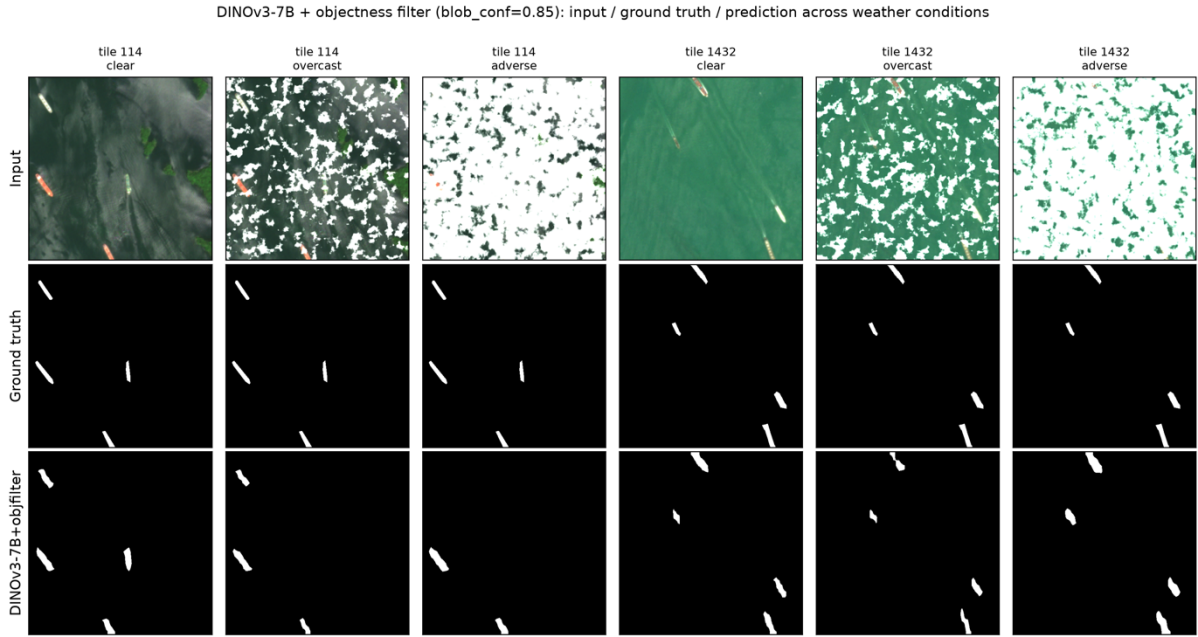


Figure 4.8 DINOv3-7B + object filter in 3 weather conditions for two scenes.

4.5 Resource cost

Resource cost was measured using a benchmark that approximates an NVIDIA Jetson AGX Orin. Latency and memory footprint were recorded for each model across a range of input sizes, with results read at the 512-pixel operating size. The benchmark runs on a desktop GPU, so the figures are adjusted approximations.

At 512 pixels, latency was 15.5 ms for the CNN, 50.5 ms for ViT-small, 95.7 ms for DINOv3-L, and 99.5 ms for ResNet-152, DINOv3-7B was far higher at 1023 ms.

Measured GPU power ranged from 72 W for the CNN to 143 W for the 7B, whose draw stayed flat across all input sizes.

Peak memory ranged from 0.02 GB for the CNN to 12.6 GB for the 7B, all within the 64 GB budget.

The reproduced (Magalhães, Falcão and Barbosa, 2025) model ran at 43.4 ms and 0.15 GB, second only to the CNN on latency. The objectness filter added 2.9 ms to the DINOv3-7B pipeline, under 0.3% of its backbone cost.

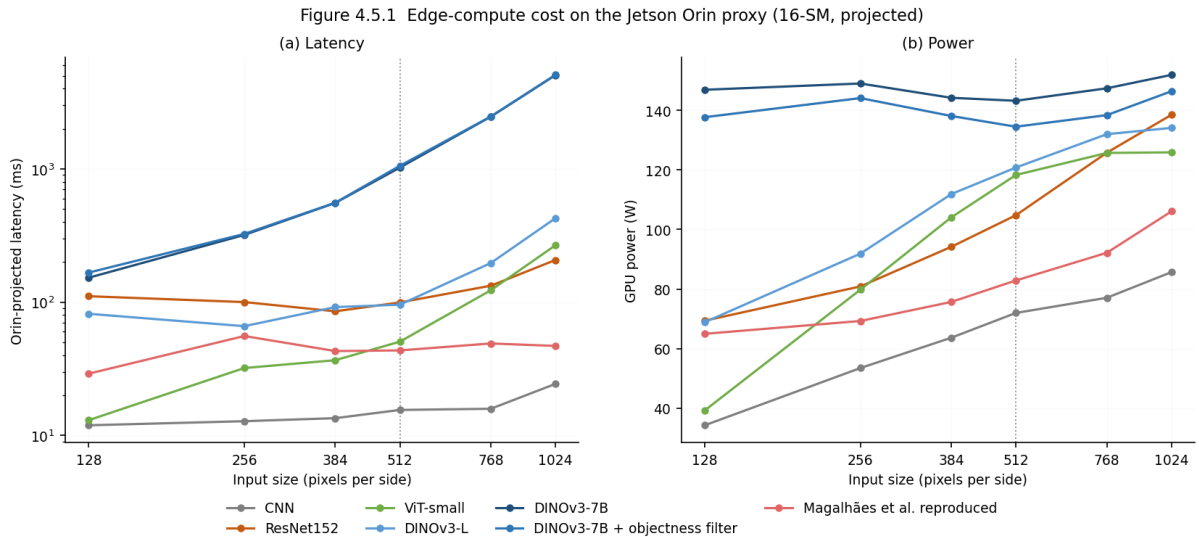


Figure 4.9 model performance on representative hardware as input size scales - (a) latency at scaled clock speed, (b) power consumption

The latency figures are projected to the Orin's 1.3 GHz clock. Power was measured at the boost clock and is not indicative of the real power use of the deployment to real-world hardware.

4.6 Comparison and trade-off

A model's operating performance is set by a decision threshold which allows trading precision and recall. Below is a chart that shows instance-level precision and recall at a range of decision thresholds for 3 different weather conditions.

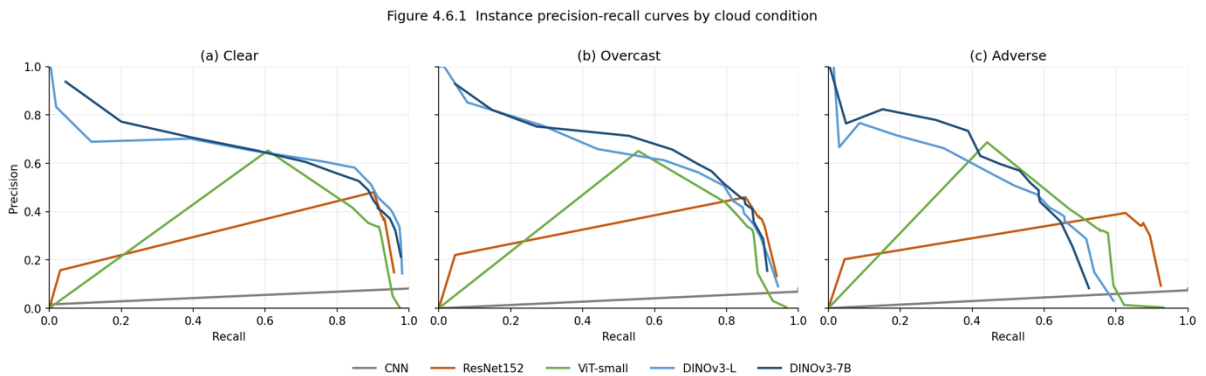


Figure 4.10 Precision recall curves on vessel instance by weather type

The DINOv3 backbones hold the highest precision across the recall range and start near a precision of 1 at low recall, unlike the other models that collapse to 0,0. The CNN stays near 0.08 precision throughout, and ResNet-152 and ViT-small fall between the two. Every curve moves inward as cloud increases; the DINOv3 advantage is widest under adverse weather.

Figure 4.6.2 Models against the success criteria (clear test set, default threshold)

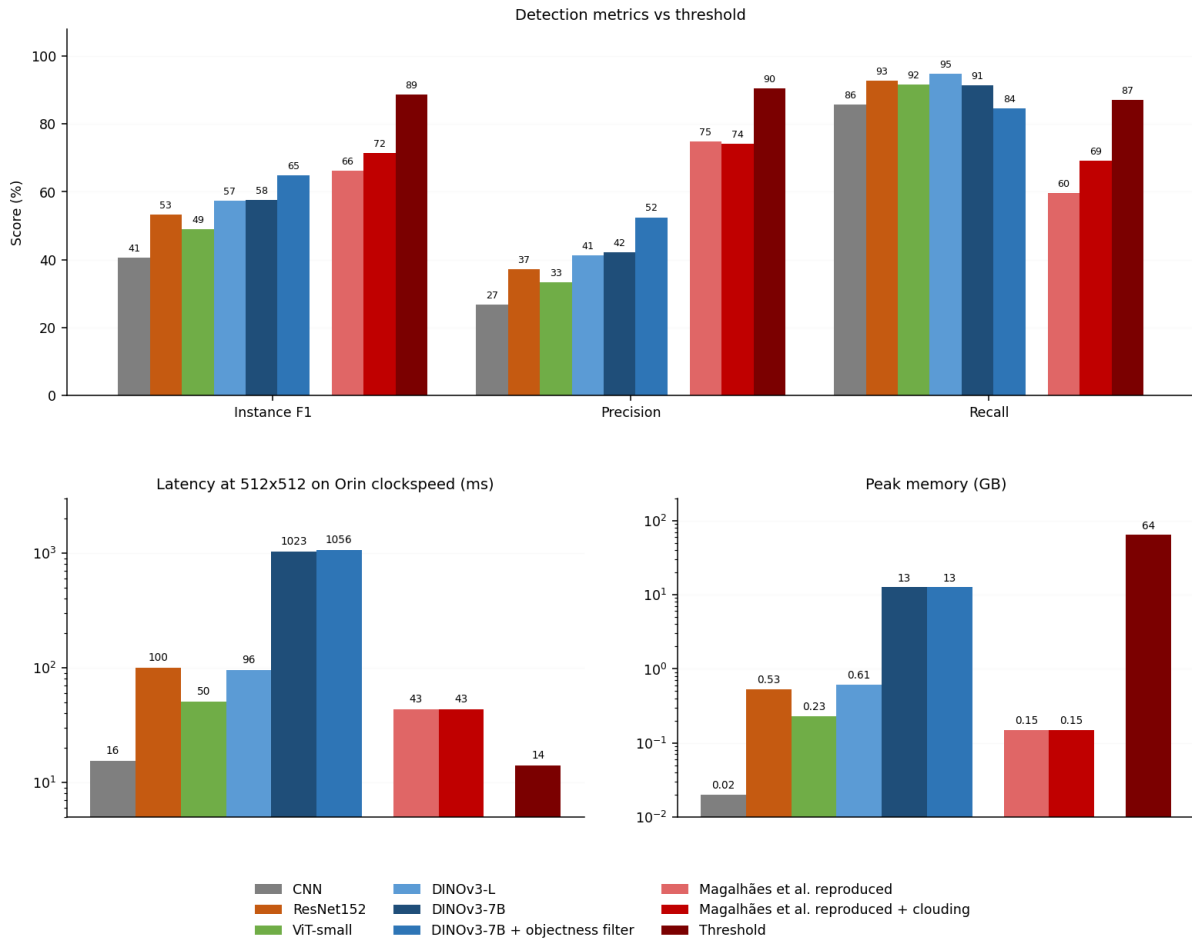


Figure 4.11 test set metrics of models compared with threshold

No model meets the accuracy or F1 thresholds set; instance F1 reaches 0.65 against a target of 0.89, and precision 0.52 against 0.905. Recall is the exception, clearing 0.87 for four of the six models, it fell below the threshold for our model with objectness filter to optimise for F1. Latency is never below 14 ms, suggesting that other models would have to work on fractions of the data or on downscaled image frames; all models fit the memory budget.

4.7 Summary

This chapter interpreted how performance changes across architectures and design decisions. Simulated cloud, single-annotator labels, one provider and proxy hardware qualify these findings. The contribution that the one shared backbone serving many lightweight user heads is the lowest cost per customer solution, and artificially adding clouds to training data improves performance even when there are no clouds in the test data. Chapter 5 discusses how these findings can be translated into useful real world findings.

Chapter 5: Discussion

This chapter provides context, evaluation reasoning, and criticism for the results. How the performance compares to the comparator and the wider literature, whether the economics make this system actually useful, and what agentic capabilities this perception layer would make possible.

5.1 Answering the research question

How can image processing for a vision-based agentic system be effectively implemented to process large quantities of spacecraft imagery on the edge?

The most effective approach identified is using quantisation to scale down a self-distilled backbone trained on satellite data to produce embeddings for image patches for satellite images on the craft, passing those vector embeddings to a collection of user-defined classifiers, and downlinking the results of said classifiers.

5.2 Performance against prior work and the success criteria

The success criteria set by (Magalhães, Falcão and Barbosa, 2025) (instance precision 0.905, recall 0.870, and F1 0.887) are not achieved by any model trained here. On clear test data, DINOv3-based models achieve the highest performance at 0.65. Recall is good; four of the models clear the comparator's 0.870. Precision, on the other hand, sits between 0.27 and 0.52 against a target of 0.905. These metrics suggest that alternative approaches may produce a better model compared to the exact implementation used by us, however, they say little of the best approach to developing the model.

This limits the performance claims that can be made about the models, noting however that the intended purpose of the models was to develop an understanding of good architectures and capabilities, not to develop the best model. Comparisons are made on precision, recall and F1 rather than the comparator's mAP50. Since mean average precision is unstable at 0.016% positive pixels, and our IoU threshold differs from theirs, the comparison is indicative rather than exact.

Existing systems have insurmountable limitations that cannot be addressed without a change in approach. AIS-based detection tops out near 25% recall because 75% of fishing vessels have no transponder (Paolo *et al.*, 2024), and current satellite-based systems operate on only the small amount of data downlinked.

Another limitation is that the hardware-equivalent latencies measured do not allow us to process 100% of imagery in real time. It is likely that imagery already scheduled for transmission, imagery with zero commercial value, and imagery taken at night have little to no value in processing. Without end-user requirements being set out, a confident statement about models being fast enough cannot be made.

5.3 Robustness

Cloudy scenes can be difficult to process; however, not processing, or processing very poorly, creates functional blindness in cloudy areas. To avoid satellite detection, illegal operators could simply fish when the clouds are out, when the system cannot see them.

Clouds hide the target, reducing the contrast between a vessel and the water around it. It also adds bright, textured structure at a range of scales, and because vessels occupy only 0.016% of pixels in training data, a reasonable pixel-wise false positive rate has a detrimental effect on instance precision. Furthermore, models tuned to clear water specify their decision boundary using clear-water statistics, clouds shift the distribution underneath that boundary, so the threshold ends up in the wrong place even where the features remain informative. Using clouded data in training can improve model performance, especially in more difficult scenes; however, it is quite impossible to look at an opaque medium and determine what's behind it.

Clouds do not stop these models from finding vessels; they do degrade confidence in the results, and if the end user cannot trust the data, then it is functionally useless. However, if it can still be used for decision-making, then it is still useful. Exactly what this threshold is would need to be determined by a user, something that is not publicly available and is lacking; this further exemplifies the need for user-driven success criteria to help guide design decisions.

5.4 Successful economic implementation

The system described runs many customer tasks over a single stream of imagery. Convincing satellite operators to put an expensive AI system on satellites and to allow users to use it for far less than their current business model (spending lots of money to make less return) is likely not a viable business proposition. However, with many users running models simultaneously, it becomes worth the satellite operators' time. Under those conditions, viability is determined by the cost vs benefit of adding an additional user. The backbone takes over 1000ms and uses up to 12.6 GB of memory, whereas the heads take ~2-3 ms and use between 8KB-1 MB of memory, suggesting that you could run hundreds of heads before you equal the cost of the backbone.

A universal backbone flattens the cost gradient of adding new users, with the sacrifice of higher baseline cost for processing. The expensive computation is run only once and is not specific to any task; each customer then reads those embeddings with their own small head. This is far cheaper than the alternative of one separate model per customer. This distinction is the reason to prefer DINOv3, but was not one of the original success criteria.

Another constraint considered is uplink. A trained head is only around 8 KB, which fits a single contact window; a ResNet-152 U-Net is roughly 196 MB, which at 512 kb/s is around 51 minutes of continuous uplink. The backbone is far larger than either, but it's paid for only once or possibly pre-flight; the head is the only component that has to be uploaded per customer use.

It is possible that the large size of the model could be a limiting factor if deploying to an active fleet of satellites, but our experiments show a minimal drop in performance when using quantised models with Int8 and even Int4 performing very similarly; this allows us to drastically reduce the size of a model and bypass this limitation.

During training, a head is small enough to run on modest hardware in a short time, so customer jobs take a couple of hours and can be handled by one moderately competent ML engineer rather than a model development team.

5.5 Agentic decision making

The system tested can be the observation layer of a larger agentic system but is not itself an agentic system.

If implemented within that system, the satellite can be allowed to autonomously make decisions that provide the best service to the satellite operator and its customers. There are three main steps in the process, which have a significant impact on the resources available to the satellite and may be advantageous to avoid. These are imaging the surface, processing the image, and transmitting the results.

An agentic system would be able to take in information about end-user needs, the satellite's state, and the environment it's capable of imaging. From this, it can decide what to do. For example, if the satellite is operating over an area where there is no commercial demand, or it is night-time, it may decide not to image at all. If it has imaged and the image contains heavy cloud cover, or a customer has played to have it downlinked anyway, it may not be worth processing the image. Depending on the results of processing, the satellite may decide to transmit the outputs of the processing, with small images attached of likely vessels, or perhaps even an entire image.

The other components of this pipeline have been shown in the literature to be viable, and the results from our experiments indicate that this approach would likely be successful. Given an end user and their requirements, it would be possible to discuss how likely they are to achieve those requirements and what the best approach is for them to meet their business needs.

Building an agentic system would require you to be able to make decisions, act, and change the inputs to the system. This is not possible without access to hardware or a very deep understanding of how the systems within the spacecraft operate.

5.6 Limitations and counterarguments

Despite the arguments put forward for our approach, there are limitations in the approach taken and the conclusions to be drawn.

The first limiting factor would be that the cloud simulation used is ultimately an imitation of clouds, and it therefore is indicative of real-world performance. Artificially generated clouds were used; this allowed us to have better ground truth of vessel locations in highly clouded areas, as the vessel can be seen clearly and we can accurately label, whereas clouded images would incur many errors by the human labeller. A similar problem with the ground truth data is that it is labelled by hand by only one person. This means that a single point of error can corrupt the dataset. One final problem is that the data comes from a single source; to make broad generalising claims, using multiple data sources from different providers would better substantiate our claims.

Our models for the different architectures use only one family; the justification for this is the fact that our objective is to find the best approach to the solution, not the objectively best model. However, different model families perform differently; it's possible that a different selection of families would result in a different conclusion if the selected architectures could scale the number of users with a similar or lower cost and/or better performance.

Additionally, when we are analysing the models, a probability threshold of 0.5 is used for all of our headline figures, even though performance changes as different thresholds are selected. Without an end-user requirement, any selected threshold would be equally valid; however, performance can be tuned, and user specifics may allow different conclusions.

The final limiting factor that needs to be considered here is that the benchmarks used to simulate performance are proxies of real-world performance. The scope of our research does not allow running on real-world flight hardware that would have additional constraints (programmes running alongside the model competing for resources, additional limitations around heat and power that hardware would be limited to, the effect that increased radiation has on increased frequency of bit-flips, and the additional engineering steps that satellite operators may place on software to the limit and reduce the effects of crashes).

5.7 Summary

In our discussion, we have found a compelling answer for the RQ which is to:

Process imagery onboard with one quantised backbone shared across users, reading its embeddings with many lightweight user-defined heads, and downlink or process only their outputs.

This makes it economically feasible: spreading one expensive computation across many users making using low-value imagery affordable to customers.

Chapter 6: Conclusion

This research examined how image processing for a vision-based agentic system can be effectively implemented to process large volumes of spacecraft imagery at the edge. Rather than answer this question by producing a single model, we identify the most effective approach to building a maximally effective system. We argue the best approach is to produce a self-supervised backbone trained on satellite imagery and quantised to fit spacecraft hardware, which produces patch embeddings that are read by many small, user-defined classification heads. Performing a single expensive computation allows the per-user cost to drop, enabling continuous computation of many use cases, without saturating downlink. The economic argument is that, with a large number of users, it lets users minimise cost and cover large areas, while allowing satellite operators to deploy a highly profitable, scalable solution.

This proposal was not solely economic; it is also supported by performance. The DINOv3 backbone outperformed all alternative approaches we tested for instance-level detection, achieving an F1 score of 0.577, compared with the best alternative of 0.532 (ResNet-152). The advantage came from higher precision and was greatest in adverse weather, an important factor in making the model more robust. Quantising down the backbone to int8 or int4 changed IoU by less than 0.01, which helps alleviate the stress of model size, especially when combined with the fact that the 7B-parameter backbone offered a negligible improvement over the much smaller ‘Large’ model while being 10 times slower.

The models developed were compared to the model by (Magalhães, Falcão and Barbosa, 2025); none of our models were able to achieve this model's instance precision of 0.905 or F1 of 0.887; only recall was able to reach its target. The objective, however, was never to produce a deployable system, but to determine the best path to build one.

6.1 Academic contribution

The contribution of this work is that it fundamentally changes what can be achieved for users seeking insights in imagery tasks where the value per square km of imagery is low. We explored this via the use case of vessel detection, where a user would need to look at a significant fraction of the Earth's surface to search for a handful of objects; the advantage of this system being able to run in parallel to regular operations allows the satellite operator to grant the user new insights at very little cost to itself.

6.2 Industrial value

There are two key points from our research:

1. Operators considering onboard processing should deploy a single quantised foundation backbone shared across customers rather than per-customer models, as this enables similar or better performance and orders-of-magnitude lower cost to bring on new users or use cases.
2. These findings are yet to be validated on real-world flight hardware, imagery from multiple providers, independently repeated labelling, or additional model families within each architecture.

6.3 Further research

Researchers considering continuing this work should be aware of the gap in published end-user requirements for detection confidence, acceptable latency, and operating thresholds; and should begin further research by establishing or deriving these criteria.

Future work on how to effectively build, train, and deploy the backbone needs to be considered. Specifically, we believe that a backbone that produces embeddings that are agnostic to clouds would be incredibly valuable and suggest future research into how this can be achieved; perhaps using our clouding technique. We also believe that federated training of a backbone on-board the spacecraft would yield better results, as it would allow the use of all satellite images for training, and not only downlinked images.

6.4 Final thoughts

EO spacecraft can already see far more of the planet than they can ever downlink; this gap limits what can be monitored from orbit. Pragmatically, the value of EO imagery is the insight it produces, not the imagery itself. The assumption that the imagery itself is the product comes from the historical requirement for humans, not computers, to extract insights. This research shows you can close this gap with a novel implementation of existing tools, enabling otherwise economically unreachable use cases such as vessel detection to become viable.

References

- Ahumada, J., Doerksen, K. and Zeller, S. (2021) ‘Automated fleet commissioning workflows at planet’. Available at: <https://doi.org/10.26077/vfkb-a117>.
- Born, M., Wolf, E., and Avadh Behari Bhatia (2016) *Principles of optics electromagnetic theory of propagation, interference and diffraction of light*. Cambridge Cambridge Univ. Press.
- Bouthillier, X. *et al.* (2021) ‘Accounting for Variance in Machine Learning Benchmarks’. arXiv. Available at: <https://doi.org/10.48550/arXiv.2103.03098>.
- Caron, M. *et al.* (2021) ‘Emerging Properties in Self-Supervised Vision Transformers’. arXiv. Available at: <https://doi.org/10.48550/arXiv.2104.14294>.
- Chien, S. *et al.* (2005) ‘Using Autonomy Flight Software to Improve Science Return on Earth Observing One’, *Journal of Aerospace Computing, Information, and Communication*, 2(4), pp. 196–216. Available at: <https://doi.org/10.2514/1.12923>.
- Devaraj, K. (2022) ‘Behind the network that images the whole world every day’. Available at: <https://blog.apnic.net/2022/07/27/behind-the-network-that-images-the-whole-world-every-day/> (Accessed: 1 April 2026).
- Dosovitskiy, A. *et al.* (2021) ‘An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale’. arXiv. Available at: <https://doi.org/10.48550/arXiv.2010.11929>.
- Efron, B. and Tibshirani, R.J. (1993) *An Introduction to the Bootstrap*. 0 edn. Chapman and Hall/CRC. Available at: <https://doi.org/10.1201/9780429246593>.
- European Space Agency (2026) ‘Planet pelican’. Available at: <https://www.eoportal.org/satellite-missions/planet-pelican> (Accessed: 1 April 2026).
- European Union Agency for the Space Programme (2022) *EUSPA EO and GNSS Market Report*. European Union. Available at: <https://doi.org/10.2878/94903>.
- Everingham, M. *et al.* (2010) ‘The Pascal Visual Object Classes (VOC) Challenge’, *International Journal of Computer Vision*, 88(2), pp. 303–338. Available at: <https://doi.org/10.1007/s11263-009-0275-4>.
- Furano, G. *et al.* (2020) ‘Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities’, *IEEE Aerospace and Electronic Systems Magazine*, 35(12), pp. 44–56. Available at: <https://doi.org/10.1109/MAES.2020.3008468>.
- Giuffrida, G. *et al.* (2020) ‘CloudScout: A deep neural network for on-board cloud detection on hyperspectral images’, *Remote Sensing*, 12(14), p. 2205. Available at: <https://doi.org/10.3390/rs12142205>.
- Giuffrida, G. *et al.* (2022) ‘The Φ -Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation’, *IEEE Transactions on Geoscience and Remote Sensing*, 60, pp. 1–14. Available at: <https://doi.org/10.1109/TGRS.2021.3125567>.

- He, K. *et al.* (2016) ‘Deep Residual Learning for Image Recognition’. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. Available at: <https://doi.org/10.1109/CVPR.2016.90>.
- Hendrycks, D. and Dietterich, T. (2019) ‘Benchmarking Neural Network Robustness to Common Corruptions and Perturbations’. *arXiv*. Available at: <https://doi.org/10.48550/arXiv.1903.12261>.
- Hinton, G., Vinyals, O. and Dean, J. (2015) ‘Distilling the Knowledge in a Neural Network’. *arXiv*. Available at: <https://doi.org/10.48550/arXiv.1503.02531>.
- Hoffmann, J. *et al.* (2022) ‘Training Compute-Optimal Large Language Models’, *arXiv preprint* [Preprint]. Available at: <https://doi.org/10.48550/arXiv.2203.15556>.
- Howard, A.G. *et al.* (2017) ‘MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications’, *arXiv* [Preprint]. Available at: <https://doi.org/10.48550/arXiv.1704.04861>.
- Huang, G. *et al.* (2017) ‘Densely Connected Convolutional Networks’. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708. Available at: <https://doi.org/10.1109/CVPR.2017.243>.
- Jet Propulsion Laboratory (2025) ‘How NASA is testing AI to make earth-observing satellites smarter’, *NASA*. NASA. Available at: <https://www.nasa.gov/science-research/earth-science/how-nasa-is-testing-ai-to-make-earth-observing-satellites-smarter/>.
- Jewett, R. (2025) ‘Planet unveils next-gen owl satellites with 1 meter resolution, nvidia GPUs’. Available at: <https://www.satellitetoday.com/imagery-and-sensing/2025/10/08/planet-unveils-next-gen-owl-satellites-with-1-meter-resolution-nvidia-gpus/> (Accessed: 1 April 2026).
- Kaplan, J. *et al.* (2020) ‘Scaling Laws for Neural Language Models’, *arXiv* [Preprint]. Available at: <https://doi.org/10.48550/arXiv.2001.08361>.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ‘ImageNet Classification with Deep Convolutional Neural Networks’. *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., pp. 1097–1105.
- Kroodsma, D.A. *et al.* (2018) ‘Tracking the global footprint of fisheries’, *Science*, 359(6378), pp. 904–908. Available at: <https://doi.org/10.1126/science.aao5646>.
- Lecun, Y. *et al.* (1998) ‘Gradient-based learning applied to document recognition’, *Proceedings of the IEEE*, 86(11), pp. 2278–2324. Available at: <https://doi.org/10.1109/5.726791>.
- Lin, T.-Y. *et al.* (2014) ‘Microsoft COCO: Common Objects in Context’, in D. Fleet *et al.* (eds) *Computer Vision – ECCV 2014*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 740–755. Available at: https://doi.org/10.1007/978-3-319-10602-1_48.

Magalhães, R., Falcão, A.P. and Barbosa, A. (2025) ‘Vessel detection leveraging satellite imagery and YOLO in maritime surveillance’, *Remote Sensing Applications: Society and Environment*, 40, p. 101730. Available at: <https://doi.org/10.1016/j.rsase.2025.101730>.

Marin, A. *et al.* (2021) ‘Phi-Sat-2: Onboard AI Apps for Earth Observation’, *ResearchGate*. Available at: https://www.researchgate.net/publication/358278832_Phi-Sat-2_Onboard_AI_Apps_for_Earth_Observation (Accessed: 16 March 2025).

Masterman, T. *et al.* (2024) ‘The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey’. *arXiv*. Available at: <https://doi.org/10.48550/arXiv.2404.11584>.

Mäyrä, J. *et al.* (2025) ‘Mapping recreational marine traffic from Sentinel-2 imagery using YOLO object detection models’, *Remote Sensing of Environment*, 326, p. 114791. Available at: <https://doi.org/10.1016/j.rse.2025.114791>.

Micikevicius, P. *et al.* (2018) ‘Mixed Precision Training’. *arXiv*. Available at: <https://doi.org/10.48550/arXiv.1710.03740>.

NVIDIA (2022) *NVIDIA Jetson Orin*. Available at: <https://www.nvidia.com/en-gb/autonomous-machines/embedded-systems/jetson-orin/> (Accessed: 1 April 2026).

OECD (2023) *The space economy in figures: responding to global challenges*. OECD Publishing. Available at: <https://doi.org/10.1787/fa5494aa-en>.

Ouellette, M. (2025) ‘Is NVIDIA’s jetson thor the robot brain we’ve been waiting for?’ Available at: <https://www.engineering.com/is-nvidias-jetson-thor-the-robot-brain-weve-been-waiting-for/> (Accessed: 1 April 2026).

Paolo, F. *et al.* (2022) ‘xView3-SAR: Detecting Dark Fishing Activity Using Synthetic Aperture Radar Imagery’. *arXiv.org*. Available at: <https://doi.org/10.48550/arXiv.2206.00897>.

Paolo, F. *et al.* (2024) ‘Satellite mapping reveals extensive industrial activity at sea’, *Nature*, 625(7993), pp. 85–91. Available at: <https://doi.org/10.1038/s41586-023-06825-8>.

Picard, D. (2021) ‘Torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision’. *arXiv*. Available at: <https://doi.org/10.48550/arXiv.2109.08203>.

Planet Labs PBC (2024a) ‘Planet education and research program’, *Planet Labs*. Available at: <https://www.planet.com/industries/education-and-research/>.

Planet Labs PBC (2024b) ‘Planet’s modular, extensible smallsat platform enables new missions including Pelican and Tanager’, *Planet Pulse*. Available at: <https://www.planet.com/pulse/modular-extensible-smallsat-platform/> (Accessed: 1 April 2026).

Planet Labs PBC (2024c) ‘PlanetScope product specifications’, *Planet Labs*. Planet Labs. Available at: <https://developers.planet.com/docs/data/planetscope/>.

Planet Labs PBC (2024d) ‘Tanager mission documentation’. Available at: <https://docs.planet.com/data/imagery/tanager/techspec/> (Accessed: 1 April 2026).

Planet Labs PBC (2026a) ‘Planet successfully runs AI in space’. Available at: <https://www.businesswire.com/news/home/20260407165913/en/Planet-Successfully-Runs-AI-in-Space> (Accessed: 9 April 2026).

Planet Labs PBC (2026b) ‘Planet to build world’s first GPU-Native AI engine for planetary intelligence with NVIDIA’. Available at: <https://www.businesswire.com/news/home/20260316377872/en/Planet-to-Build-Worlds-First-GPU-Native-AI-Engine-for-Planetary-Intelligence-with-NVIDIA> (Accessed: 1 April 2026).

SatNews (2026) “‘Software-Defined’ AI Compression Startup Secures \$3.4M to Unblock the Space-to-Ground Bottleneck”, *Satnews.com*, 12 February. Available at: <https://news.satnews.com/2026/02/12/software-defined-ai-compression-startup-secures-3-4m-to-unblock-the-space-to-ground-bottleneck/> (Accessed: 17 April 2026).

Shannon, C.E. (1948) ‘A Mathematical Theory of Communication’, *Bell System Technical Journal*, 27(4), pp. 623–656. Available at: <https://doi.org/10.1002/j.1538-7305.1948.tb00917.x>.

Simonyan, K. and Zisserman, A. (2014) ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’, *arXiv* [Preprint]. Available at: <https://doi.org/10.48550/arXiv.1409.1556>.

Sun, C. *et al.* (2017) ‘Revisiting Unreasonable Effectiveness of Data in Deep Learning Era’. *arXiv.org*. Available at: <https://doi.org/10.48550/arXiv.1707.02968>.

Tan, M. and Le, Q.V. (2019) ‘EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks’. *Proceedings of the 36th International Conference on Machine Learning (ICML)*, PMLR, pp. 6105–6114. Available at: <https://proceedings.mlr.press/v97/tan19a.html>.

Tarvainen, A. and Valpola, H. (2017) ‘Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results’. *arXiv*. Available at: <https://doi.org/10.48550/arXiv.1703.01780>.

Trigo, A. (2026) ‘Artefact-driven narrative literature review (ART-NLR): supporting evidence synthesis for design science research’, *Journal of Documentation*, 82(4), pp. 899–930. Available at: <https://doi.org/10.1108/jd-01-2026-0013>.

Vaswani, A. *et al.* (2017) ‘Attention Is All You Need’. *arXiv.org*. Available at: <https://doi.org/10.48550/arXiv.1706.03762>.

Wang, B. (2025) ‘Planet’s most advanced satellite – new owl satellites for 1 meter resolution imaging’. Available at: <https://www.nextbigfuture.com/2025/10/planets-most-advanced-satellite-new-owl-satellites-for-1-meter-resolution-imaging.html> (Accessed: 1 April 2026).

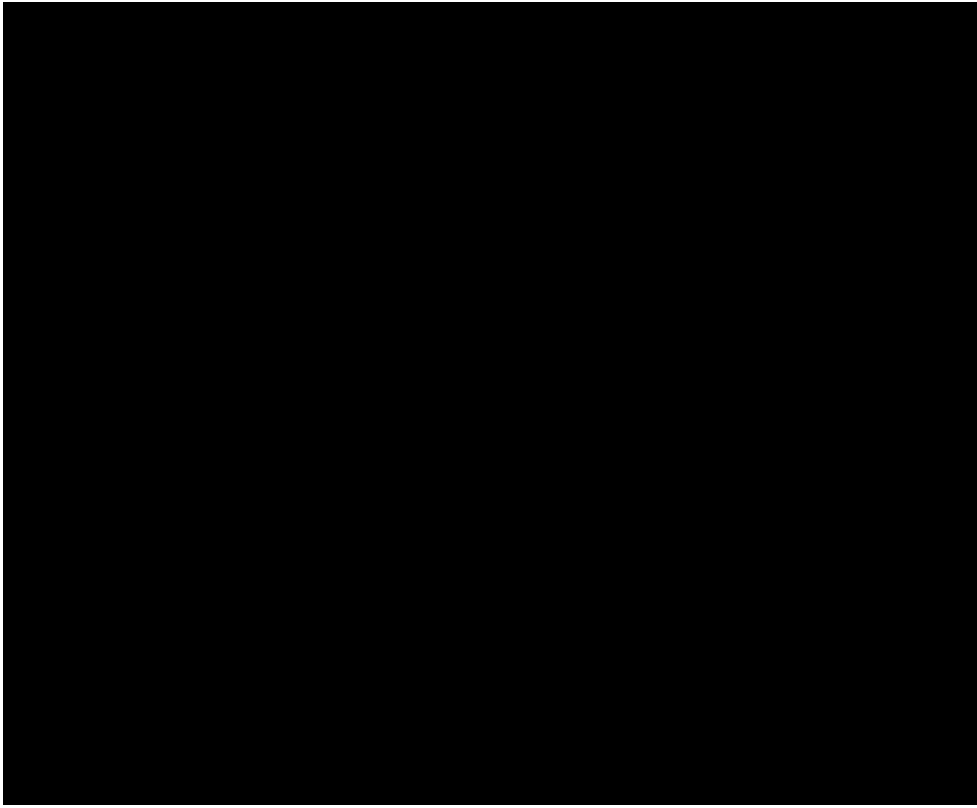
Wang, L. *et al.* (2024) ‘A survey on large language model based autonomous agents’, *Frontiers of Computer Science*, 18(6), p. 186345. Available at: <https://doi.org/10.1007/s11704-024-40231-1>.

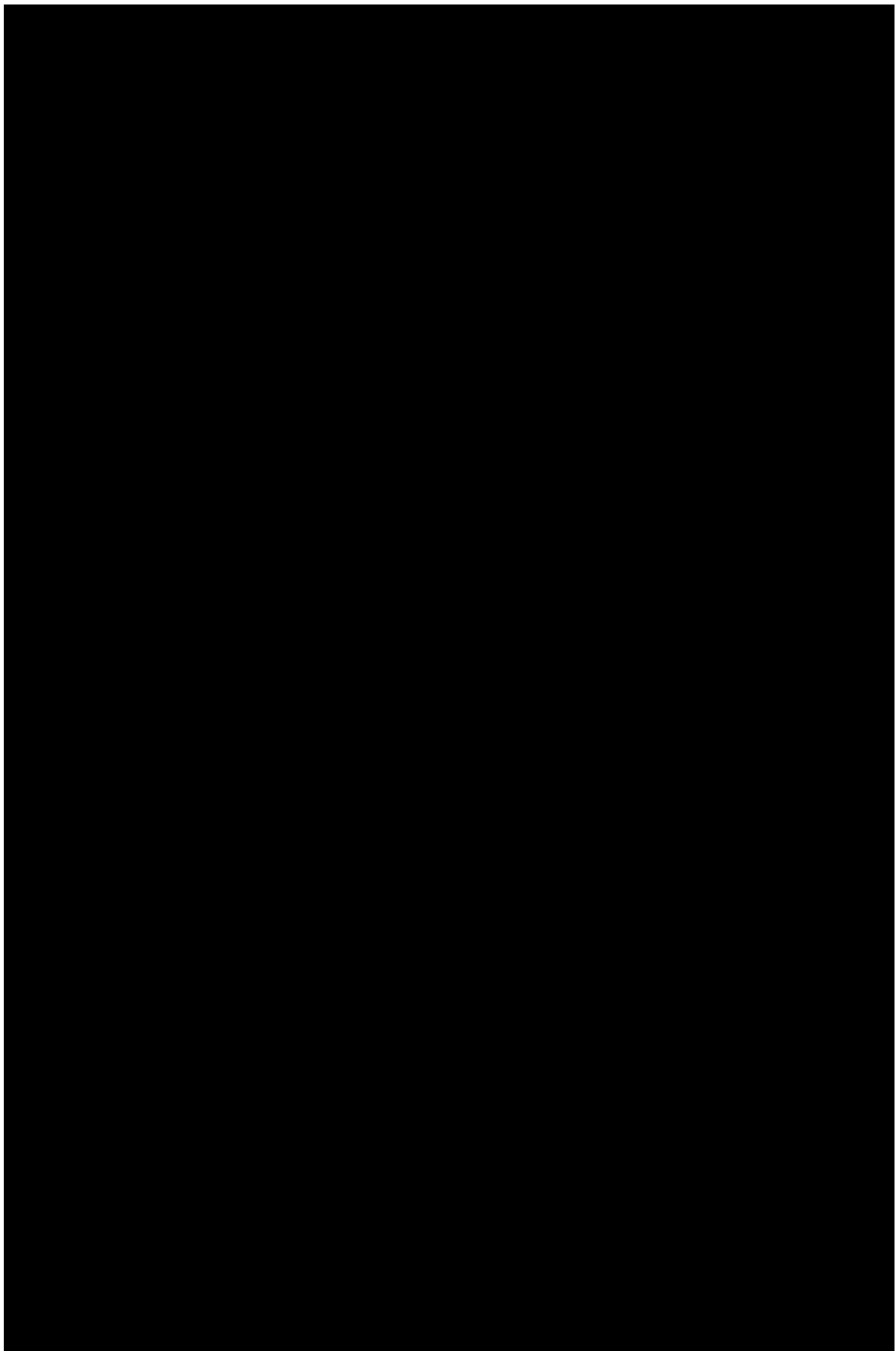
Xu, T.-B. and Liu, C.-L. (2019) ‘Data-Distortion Guided Self-Distillation for Deep Neural Networks’, *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), pp. 5565–5572. Available at: <https://doi.org/10.1609/aaai.v33i01.33015565>.

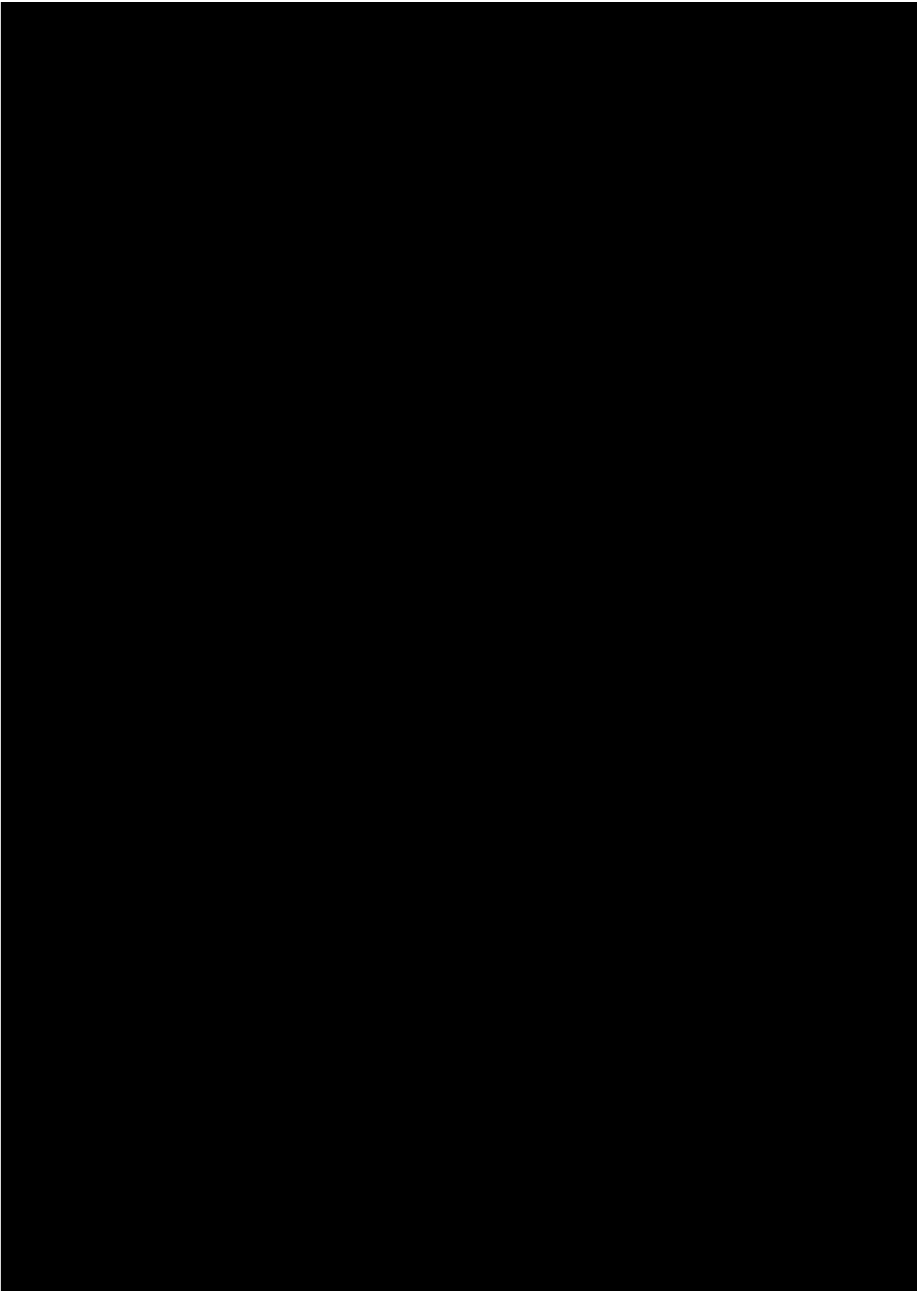
Zhai, X. *et al.* (2022) ‘Scaling Vision Transformers’. arXiv.org. Available at: <https://doi.org/10.48550/arXiv.2106.04560>.

Zhang, L. *et al.* (2019) ‘Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation’. arXiv. Available at: <https://doi.org/10.48550/arXiv.1905.08094>.

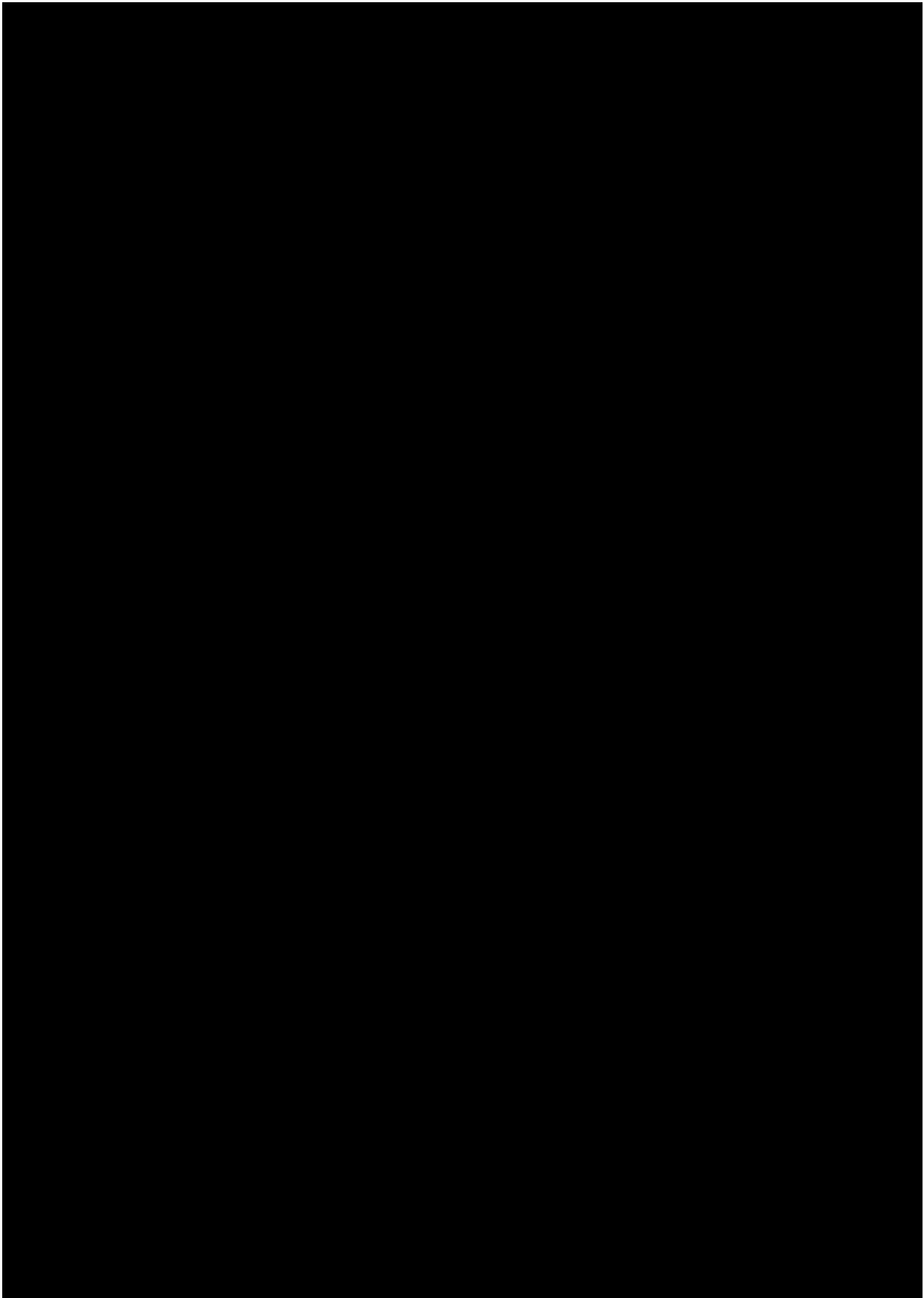
Appendix A: Certificates







Appendix B: Ethical approval waiver



Appendix C: Planet Labs education and research program terms of service (signed)

TERMS OF SERVICE FOR
EDUCATION AND RESEARCH PROGRAM
Revised May 10, 2023

Supersedes and replaces all prior versions.

WELCOME TO YOUR PLANET!

PLANET LABS PBC (“PLANET”) IS PLEASED TO MAKE CERTAIN PLANET CONTENT AVAILABLE AS PART OF ITS EDUCATION AND RESEARCH PROGRAM (THE “PROGRAM”) VIA THE PLATFORM (DEFINED BELOW). BY ACCESSING THE PLATFORM OR CONTENT (DEFINED BELOW), THE USER (“LICENSEE”) IS EXPRESSLY AGREEING TO BE BOUND BY: (1) THESE

TERMS OF SERVICE; (2) THE PLANET [TERMS OF USE](https://www.planet.com/terms-of-use/) (<https://www.planet.com/terms-of-use/>);

AND (3) PLANET’S [PRIVACY POLICY](https://www.planet.com/privacy/) (<https://www.planet.com/privacy/>) (COLLECTIVELY, THE “AGREEMENT”). PLANET IS WILLING TO GRANT THE RIGHTS AS OUTLINED HEREIN AND MAKE THE PLATFORM AND CONTENT AVAILABLE TO LICENSEE ONLY UPON THE CONDITION THAT LICENSEE ACCEPTS THE TERMS OF THE AGREEMENT. WRITTEN APPROVAL OF THE AGREEMENT IS NOT A PREREQUISITE TO THE VALIDITY OR ENFORCEABILITY OF THE AGREEMENT AND NO SOLICITATION OF ANY SUCH WRITTEN APPROVAL BY OR ON BEHALF OF PLANET SHALL BE CONSTRUED AS AN INFERENCE TO THE CONTRARY.

1. DEFINITIONS.

Content means data generated from satellite imagery made available to Licensee pursuant to these Terms of Service. “Content” does not include Third Party Content, which is subject to its own terms as set forth herein.

Derivative Product means textual reports, graphical reports, maps or pages that are derived or developed from the Content and that do not contain any source image data from the Content and from which the Content cannot be extracted, but which may include Permitted Imagery.

Licensed Materials means the Content and the Platform.

Noncommercial means use of Content for purposes that do not include uses from which Licensee will derive profits (or other consideration) or otherwise obtain a financial advantage by charging a third party in exchange for products or services that involve Licensee’s use of the Content or Derivative Product.

Permitted Imagery means any representation of Content that does not preserve the underlying imagery data and values, including red green blue, red-edge, and near infrared values. Examples of Permitted Imagery include screenshots, jpegs, annotations, and printed copies.

Platform means the application programming interface and graphical user interface that is made available to Licensee under these Terms of Service, which enables Licensee to access Content, and which Platform includes tools such as searching, viewing, and downloading Content and Third Party Content.

Third Party Content means the data of certain third parties which Planet may make available to Licensee (e.g. Landsat).

THE PROGRAM

The Education and Research Program sponsors students, researchers, and educators with access to the Licensed Materials for education and research purposes only. The Program is intended

to fulfill Planet's founding intent: *to use space to help life on Earth*. As a member of the Program, Licensee hereby acknowledges and agrees that Licensee is not and will not be compensated directly or indirectly, by any third party for Licensee's participation in this Program or for any research or resulting work product. Licensee agrees to uphold the highest ethical standards in connection with Licensee's use of the Licensed Materials. Licensee hereby confirms that Licensee's ability to publish research results using the Content is not subject to approval by any government.

LICENSEE's PLANET ACCOUNT

Licensee will need to establish a Planet account in order to access the Licensed Materials (a "Planet Account"). Planet Accounts are only available to persons not barred from receiving services under applicable laws, including but not limited to those of the United States. Each Planet Account is unique to its holder and shall not be shared. At all times, Licensee agrees to treat Licensee's password as confidential information using at least a reasonable standard of care. Licensee acknowledges and agrees that Licensee is responsible for the activity that occurs under Licensee's Planet Account. If Licensee suspect or discover that Licensee's password or access has been compromised, Licensee shall contact Planet from the following location: <https://support.planet.com/hc/en-us/requests/new/>. Nothing in this Section 3 shall be deemed to limit the Planet Terms of Use.

LIMITED LICENSE

4.1 Licensed Materials. Subject to the terms and conditions of this Agreement, Planet hereby grants to Licensee a limited, nontransferable, nonexclusive, non-sublicensable, non-assignable, revocable license to: (i) use, access, and view Content through the Platform during the Term; (ii) reproduce, store, display, download, and print Content, and create Derivative Products in perpetuity; and (iii) publish articles related to or based upon the Content in perpetuity; provided however, any usage must be solely for (x) Noncommercial education or scientific research purposes, and (y) publication in academic or scientific research journals. Licensee agrees that all such publications must include an attribution that clearly and conspicuously identifies "Planet Labs" as the source of the Content on which the publication was based.

4.2 Evaluation Data. From time to time, Planet may make certain experimental data products available to Licensee hereunder (collectively "Evaluation Data") Planet hereby grants to Licensee a non-exclusive, nontransferable, non-sublicensable license to such Evaluation Data, but solely for Licensee's own internal testing and evaluation purposes. Evaluation Data may contain bugs that may cause system or other failure and data loss. Any further use of Evaluation Data Licensee may request in writing that Planet expand the license rights to Evaluation Data such that the terms of Section 4.1 apply. Planet will consider any such request and, if agreed in Planet's sole discretion, shall provide such approval in writing.

4.3 Restrictions. Licensee may not use the Licensed Materials for any purpose except as expressly set forth in this Agreement. By way of example, and without limiting the generality of the preceding sentence, Licensee will not: (a) alter, remove, or obscure any proprietary notices, watermarks or legends included or embedded in the Licensed Materials; (b) use the Licensed Materials in violation of applicable laws or regulations; (c) adapt, alter, publicly display, publicly perform, translate, create derivative works of, or otherwise modify the Licensed Materials except as expressly authorized under this Agreement; (d) sublicense, lease, rent, loan, transfer or distribute the Licensed Materials to any third party; (e) reverse engineer, decompile, disassemble or otherwise attempt to derive the source code for the Platform; (f) allow third parties to access or use the Licensed Materials, including without limitation in any application service Licensee environment, service bureau, or time-sharing arrangements; or (g) use any aspect of the Licensed Materials as training or validation of any machine-learning model designed to replicate the Licensed Materials.

ATTRIBUTION

Licensee agrees that all publications permitted hereunder must include an attribution that clearly and conspicuously identifies “Planet Labs.” as the source of the Content on which the publication was based.

With respect to publication of Derivative Products in general, Licensee will cite Planet as a source using the following guidance (subject to formatting requirements of any particular publication where Derivative Products may appear): “Planet Team (20xx). Planet Application Program Interface: In Space for Life on Earth. San Francisco, CA. <https://api.planet.com>.” (where xx denotes the current year).

With respect to publication of Derivative Products that include Permitted Imagery, Licensee will at all times clearly and prominently display the following Planet copyright notice on or adjacent to the applicable Content: “Imagery © 20xx Planet Labs” (where xx denotes the current year of the Content).

RESERVATION OF RIGHTS

Except for the limited license granted under Section 4 above, as between Licensee and Planet, Planet retains all right, title and interest, including all intellectual property rights, in and to the Platform, the Content, and all other Planet property. All rights not expressly granted in this Agreement are hereby expressly reserved by Planet.

THIRD PARTY CONTENT

Any Third-Party Content available on or accessible via the Platform is subject to separate terms and conditions included in the “Terms” (or similar) section of the Platform site and as listed below. Any Third Party Content made available to Licensee is provided as a convenience and at no charge. As such, Licensee acknowledge and agree that Planet makes no representations or commitments as to any such Third Party Content and assumes no liability in connection with any such Third Party Content.

Current Third Party Content available includes the following and is subject to the applicable terms set forth below:

Landsat: Landsat data may be available to view and download, and subject to the following terms: <https://developers.planet.com/docs/data/landsat-8/>

Sentinel: Sentinel data may be available to view and download, and subject to the following terms: <https://developers.planet.com/docs/data/sentinel211c/>

TERM AND TERMINATION

This Agreement is effective for so long as Licensee uses and accesses the Content and/or the Platform, provided however that Planet may at any time add, change, or remove features or functions of the Platform. Planet will provide notice to the Licensee in the event these Terms of Service, Planet’s Privacy Policy, or Planet’s Terms of Use are modified. Licensee may stop using our Content at any time.

DISCLAIMER

THE PLATFORM, THE CONTENT, AND THIRD PARTY CONTENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTIES OF ANY KIND, AND PLANET HEREBY EXPRESSLY DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT, ERROR-FREE OR UNINTERRUPTED SERVICE, PERFORMANCE, OR SECURITY. LICENSEE ACKNOWLEDGES AND AGREES THAT LICENSEE HAS NOT RELIED ON ANY

REPRESENTATIONS OR WARRANTIES IN CONNECTION HERewith.

LIMITATION OF LIABILITY

IN NO EVENT SHALL PLANET BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES, INCLUDING WITHOUT LIMITATION LOSS OF DATA, GOODWILL, BUSINESS, REVENUE, PROFITS, OR

OTHER LOSS (INCLUDING WITHOUT LIMITATION SUBSTITUTION OF SERVICE) ARISING OUT OF OR RELATED TO THIS AGREEMENT, THE CONTENT OR THE PLATFORM, EVEN IF PLANET WAS PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF WHETHER SUCH LIABILITY SOUNDS IN CONTRACT, NEGLIGENCE, TORT, OR OTHER THEORY OF LIABILITY. IN NO EVENT SHALL PLANET'S TOTAL CUMULATIVE LIABILITY FOR ANY CLAIM UNDER THIS AGREEMENT EXCEED U.S. \$100.00 USD.

COMPLIANCE WITH LAWS, REGULATORY, DISASTER RELIEF, ETHICS

11.1 Compliance with Laws. Including but not limited to with respect to the disposition of the Content, Licensee shall comply fully with all laws and regulations, to the extent applicable to Licensee or the Content. Such regulations include, without limitation, the Foreign Corrupt Practices Act and other anti-corruption laws and regulations, economic sanctions, and export controls administered by the U.S. Department of the Treasury, the U.S. Department of Commerce, the U.S. Department of State, and other governments and governmental entities (collectively, "Trade Control Laws"). Without limiting the foregoing, Licensee shall ensure that neither the Content nor any part or derivation thereof is: (a) provided to or the subject of any transaction or dealing, directly or indirectly, with or related to an Sanctioned Jurisdiction or Restricted Party (as such terms are defined below), except as licensed or otherwise authorized under Trade Control Laws; (b) exported, reexported, transferred, retransferred or otherwise shipped, directly or indirectly, in violation of any applicable Trade Control Laws; or (c) used for nuclear end-uses, rocket systems, unmanned air vehicles, chemical or biological weapons, maritime nuclear propulsion, weapons of mass destruction, or other restricted end-uses except as licensed or otherwise authorized under applicable Trade Control Laws. "Sanctioned Jurisdiction" means a country, region, territory, or government with respect to which the U.S. government imposes economic sanctions (e.g., Crimea region of Ukraine, Cuba, Iran, North Korea, Sudan, Syria, and Venezuela). "Restricted Party" means an individual or entity included on any of the restricted party lists maintained by the U.S. Government (e.g., Specially Designated Nationals List, Foreign Sanctions Evader List, Sectoral Sanctions Identification List, Denied Persons List, Unverified List, Entity List, or List of Statutorily Debarred Parties).

11.2 Ethics. Licensee will read Planet's Code of Ethics (<https://www.planet.com/ethics/> or successor URL) and will not use the Licensed Materials in contravention thereof. Without limiting Licensee's obligation to comply with applicable international law, Planet may suspend or terminate Licensee's access to the Licensed Materials if Planet has a reason to believe that Licensee has used the Licensed Materials for the intentional targeting of civilians or in connection with the indiscriminate use of weapons.

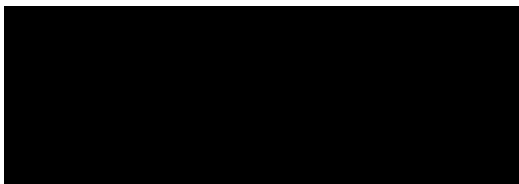
SURVIVAL

Sections 1, 6, 7, 9, 10, 11, 12, and 13 of this Agreement shall survive the expiration or earlier termination of this Agreement.

GENERAL MATTERS

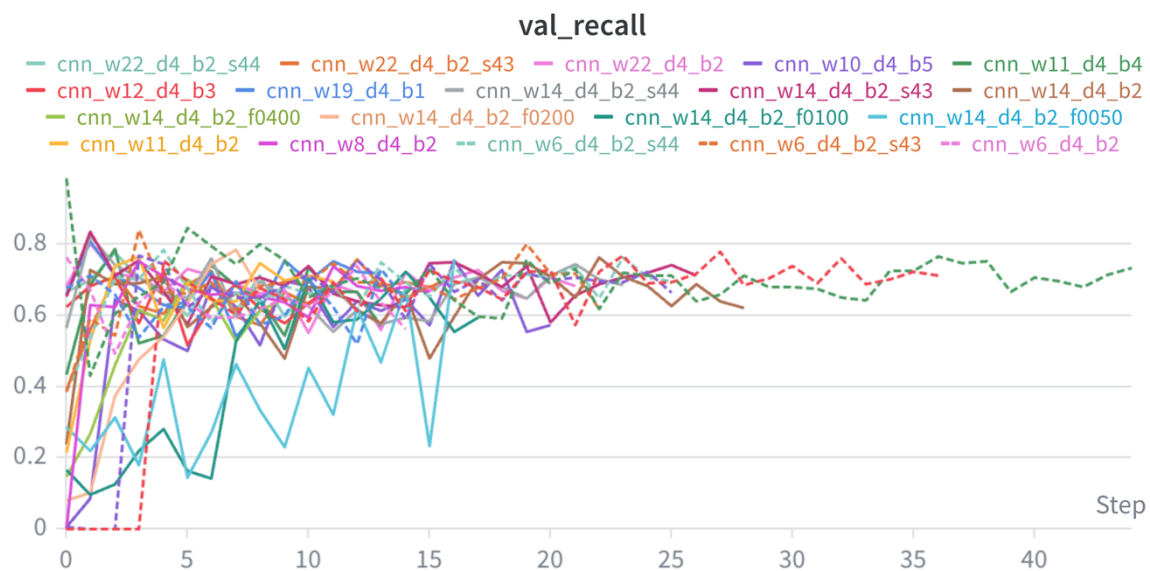
This Agreement represents the entire and exclusive agreement between Licensee and Planet and replaces any prior agreement or understanding regarding the subject matter hereof. If any term(s) of this Agreement are found to be unenforceable, the remainder will not be affected. If Licensee fails to comply with any one or more of the terms of this Agreement, any delay in Planet's response does not mean that Planet gives up any rights it may have and it may take action in the future. This Agreement will be governed by the laws of the State of California, irrespective of its choice of law provisions. All actions or proceedings arising under or related to this Agreement shall be brought in a federal or state court located in San Francisco, California, and each party hereby agrees to irrevocably submit to the jurisdiction and venue of these courts.

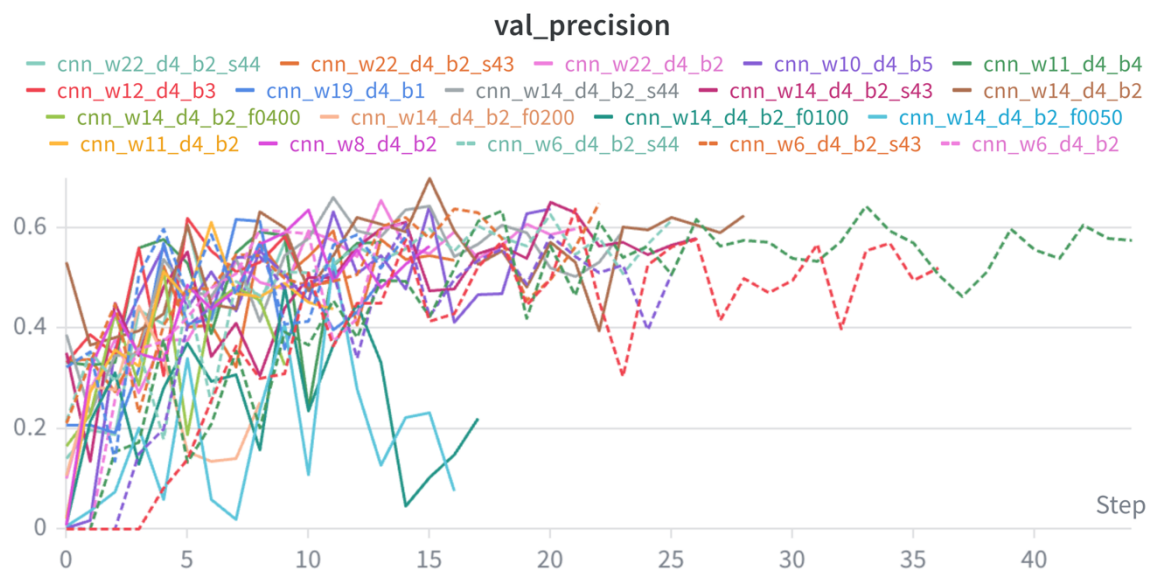
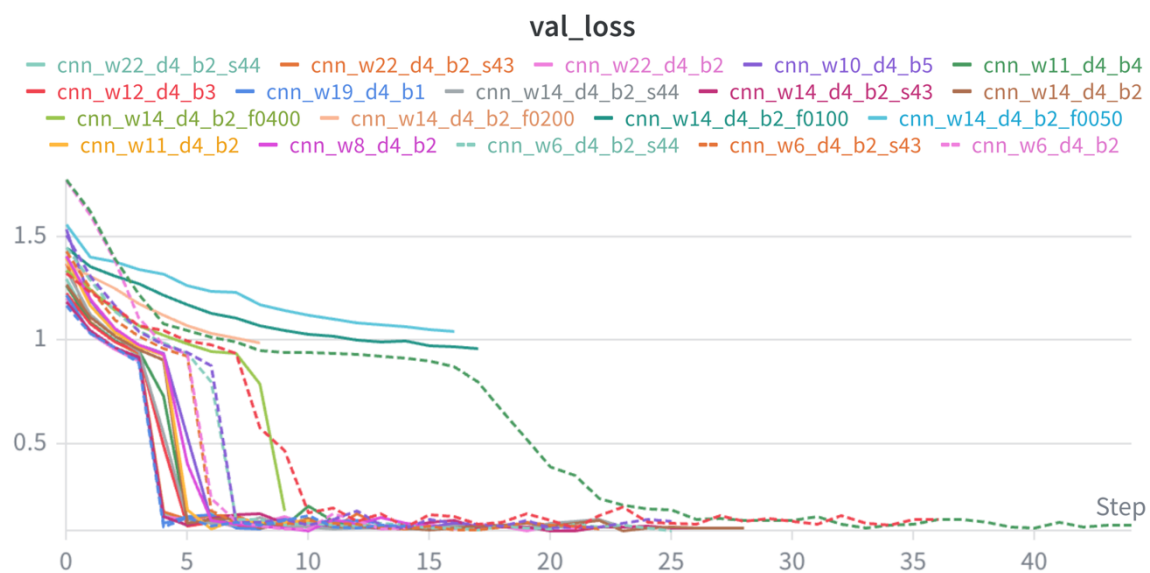
I HAVE READ AND AGREE TO THESE TERMS OF SERVICE FOR THE EDUCATION AND RESEARCH PROGRAM.

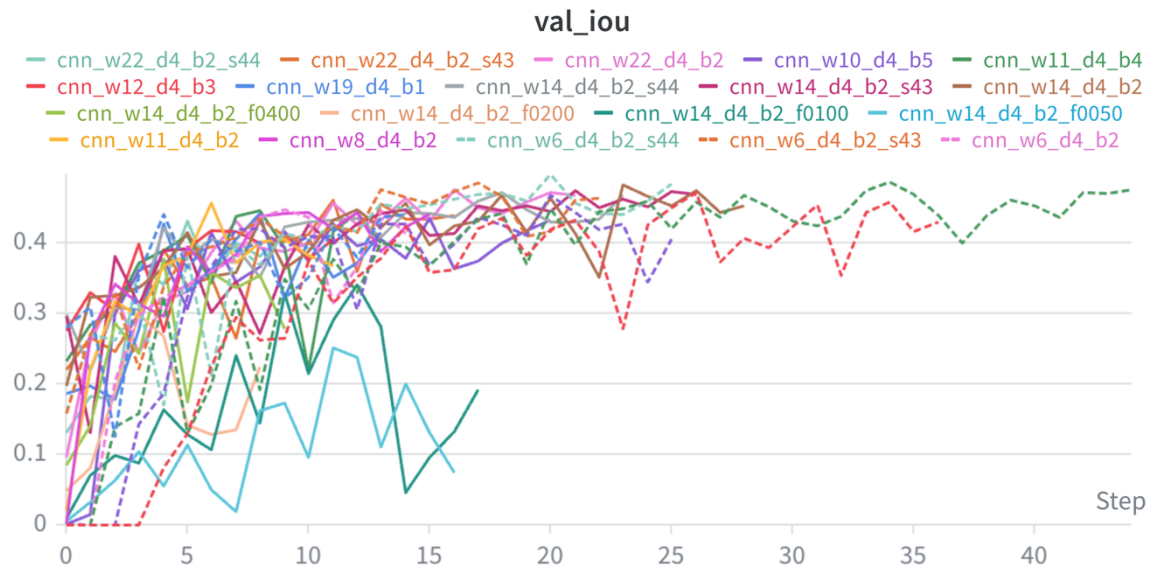
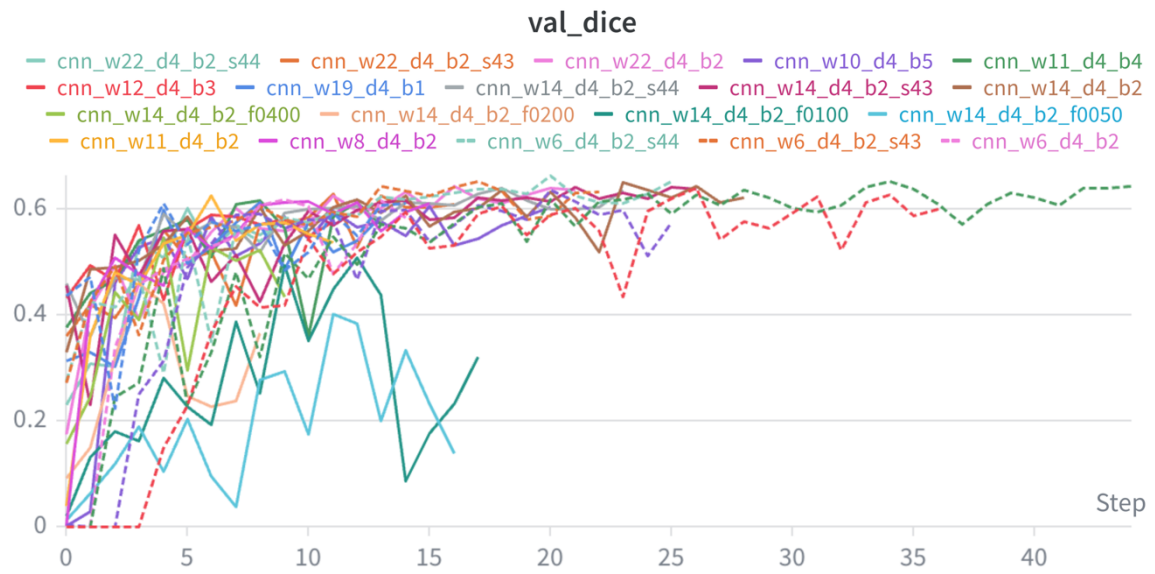


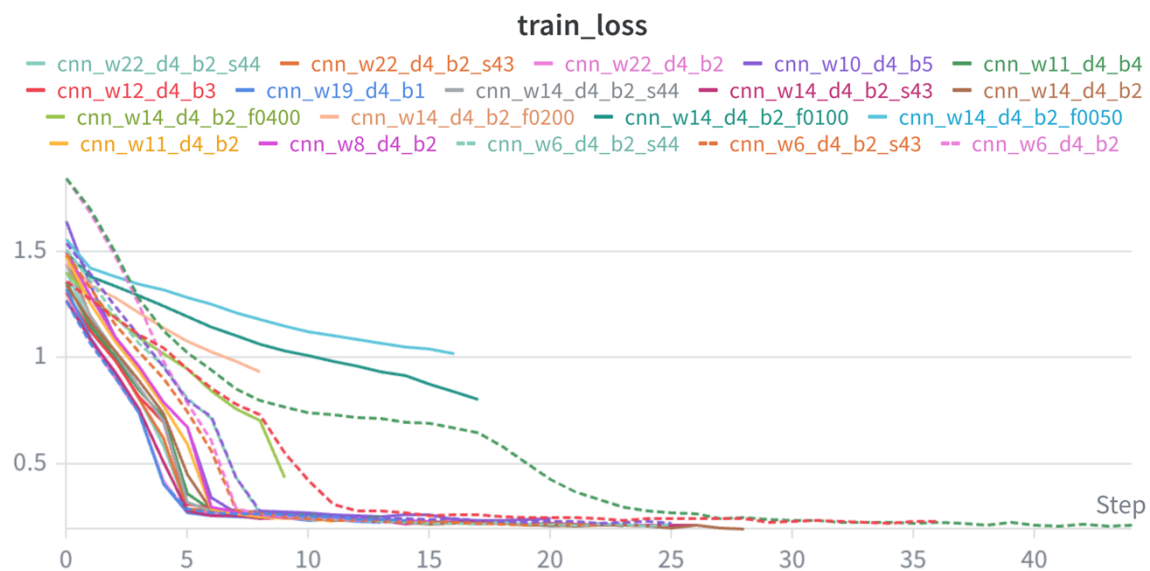
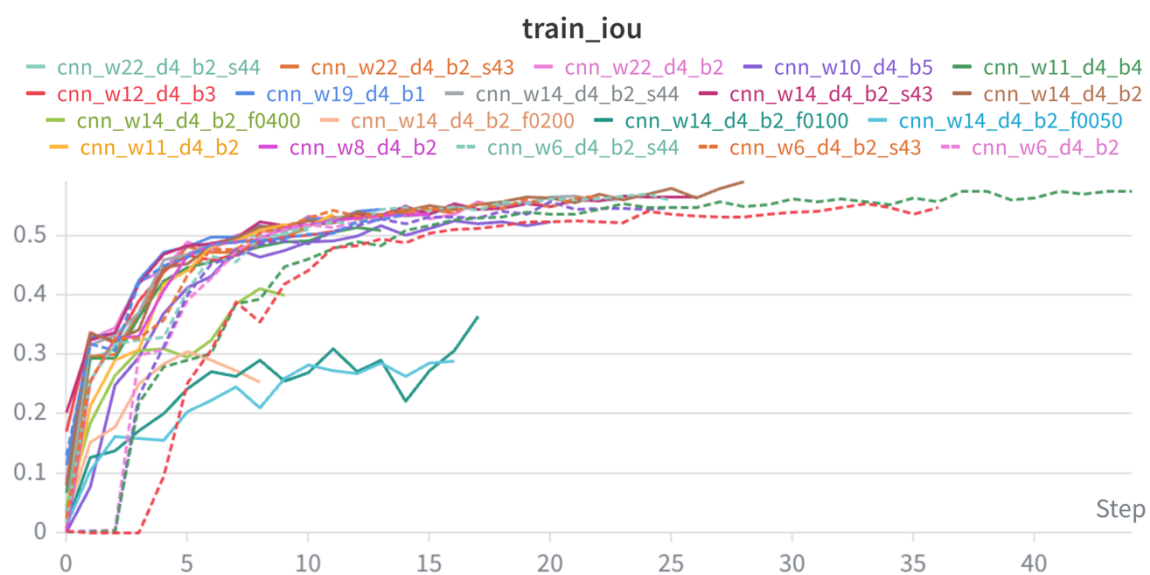
Appendix D: Hyperparameter sweeps

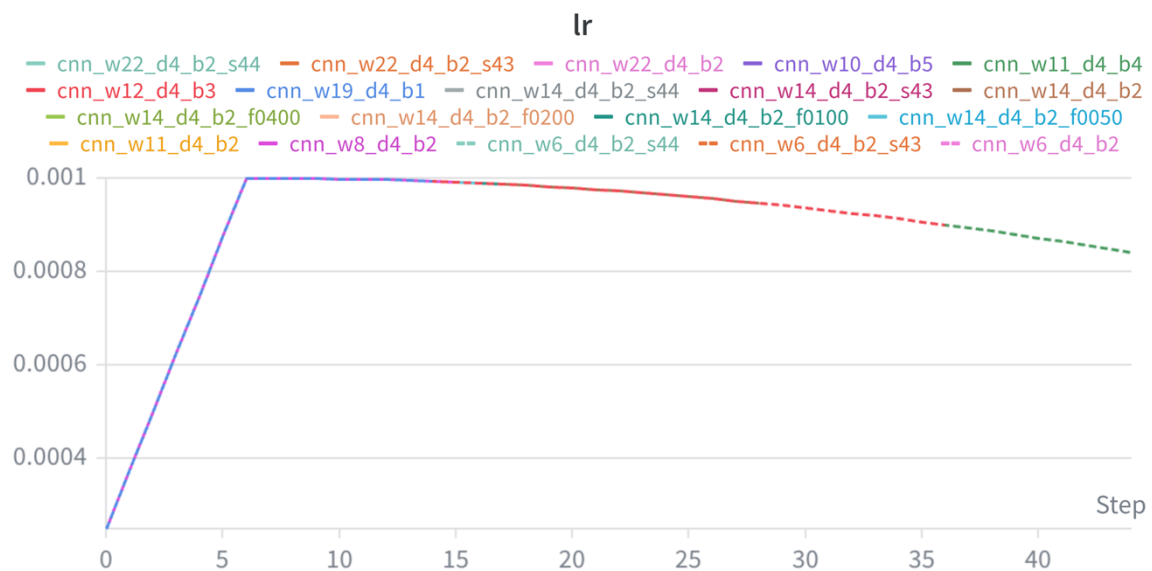
Appendix D.1 Foundation CNN



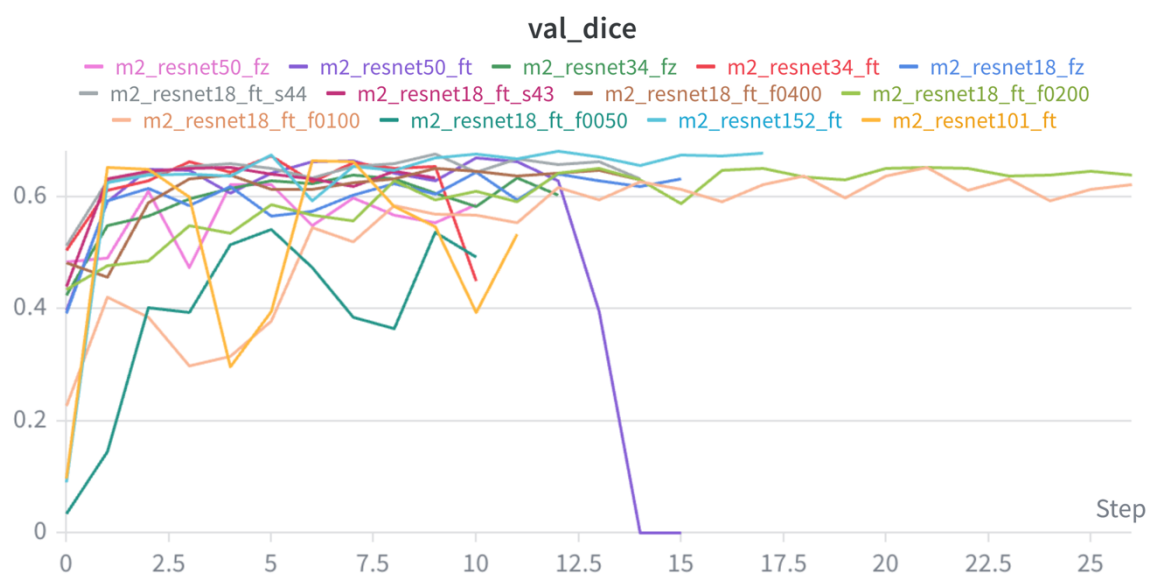


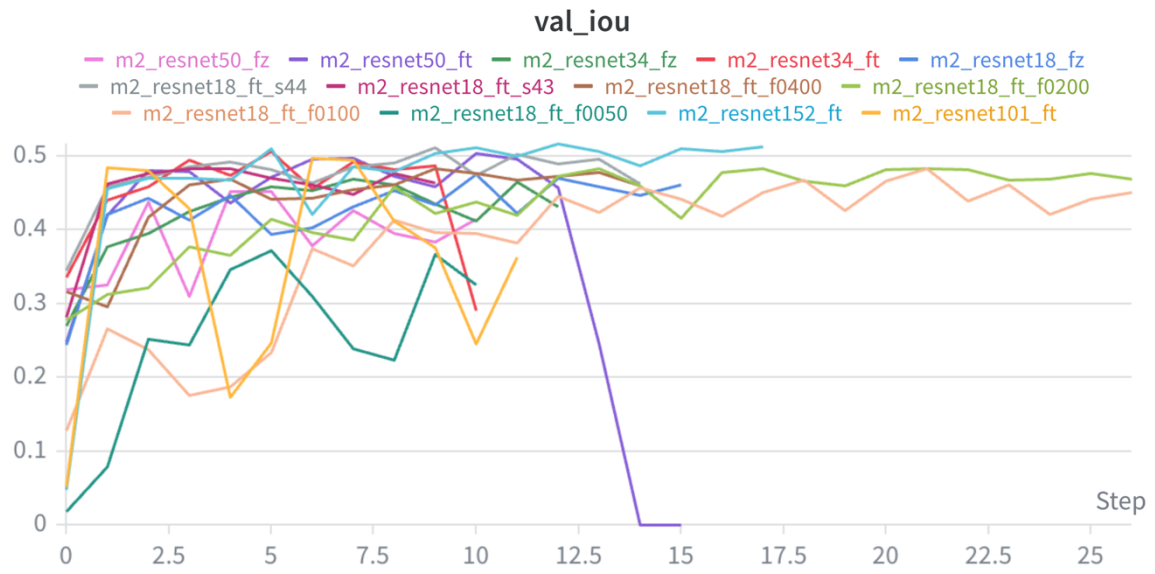
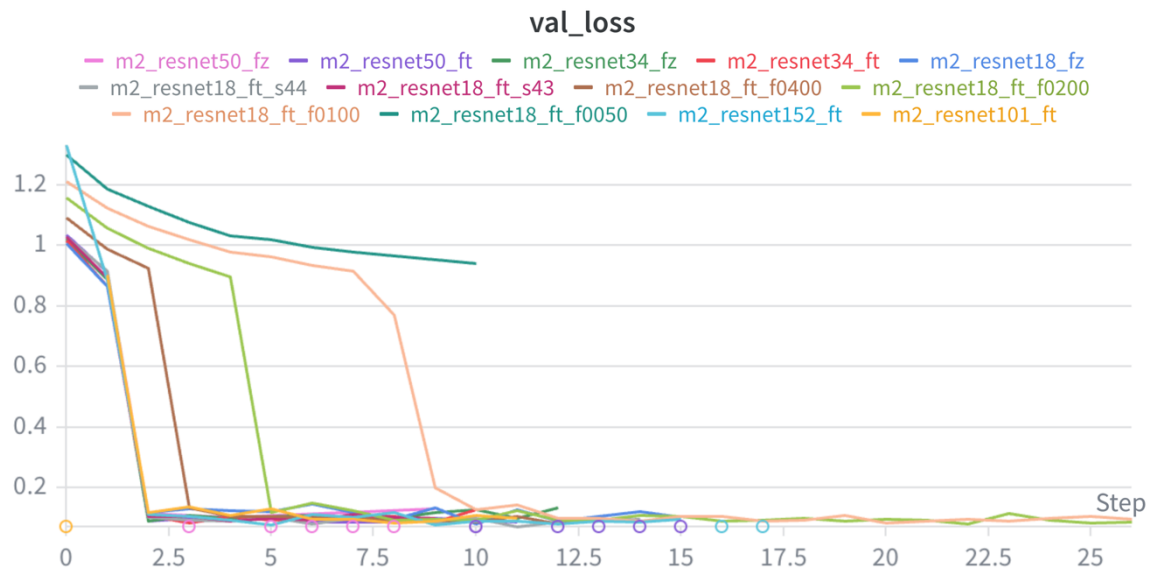


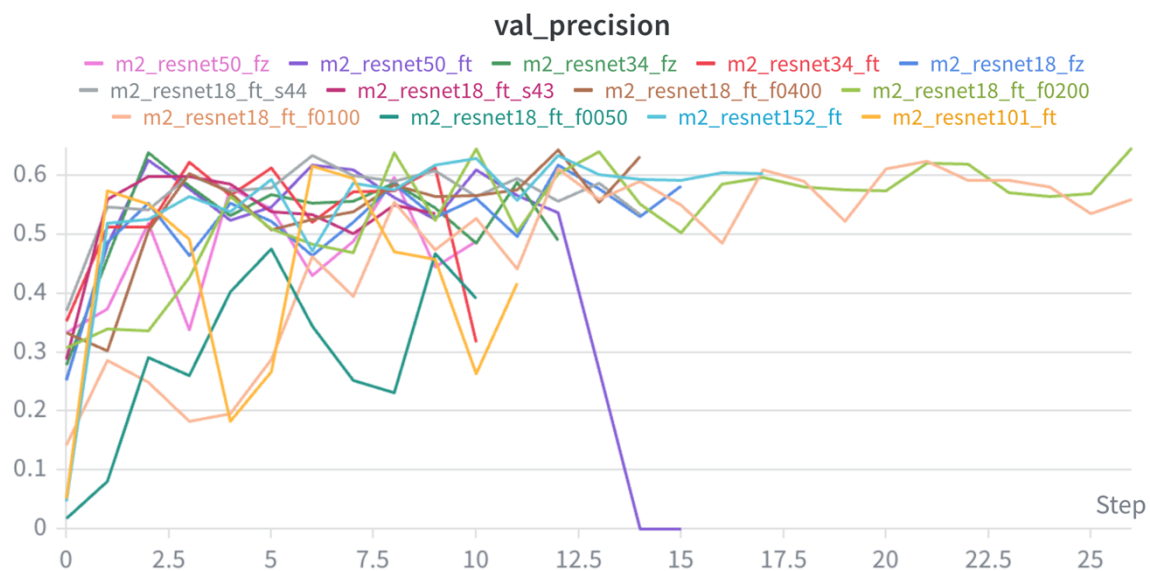
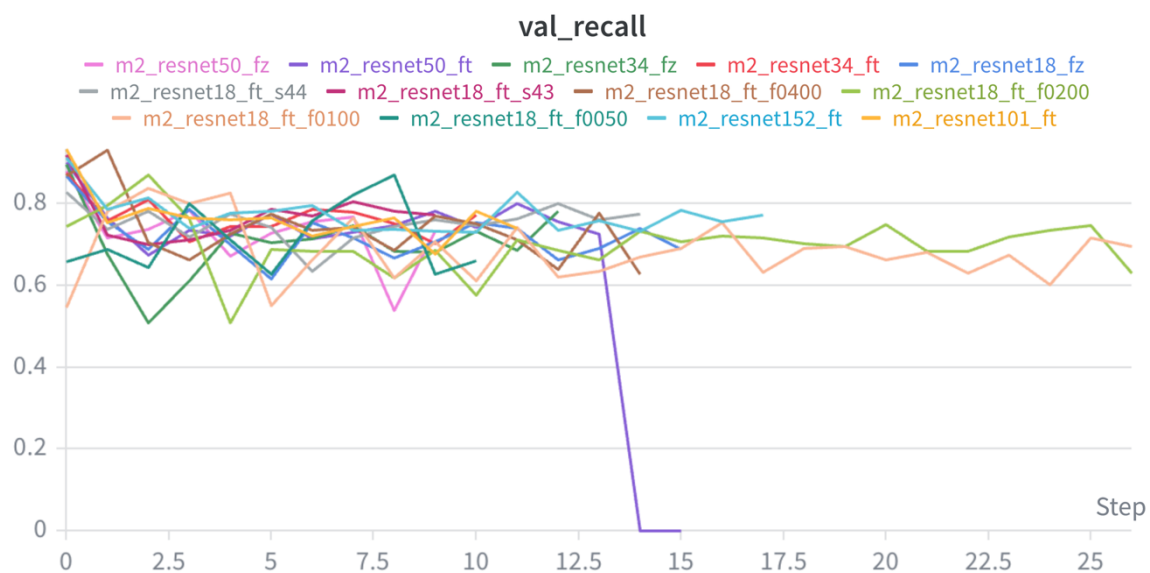


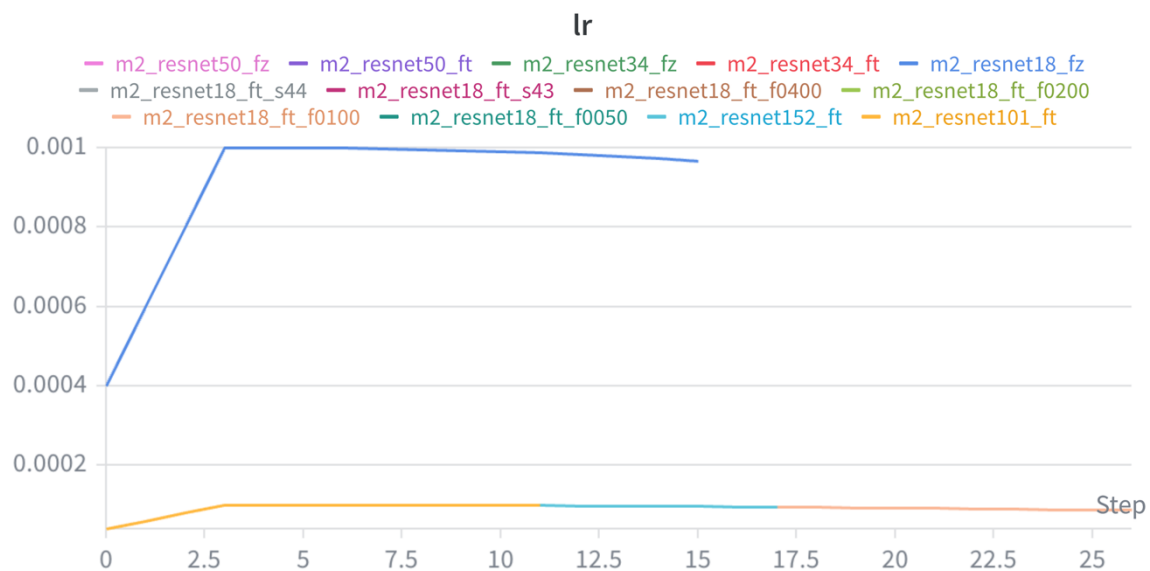
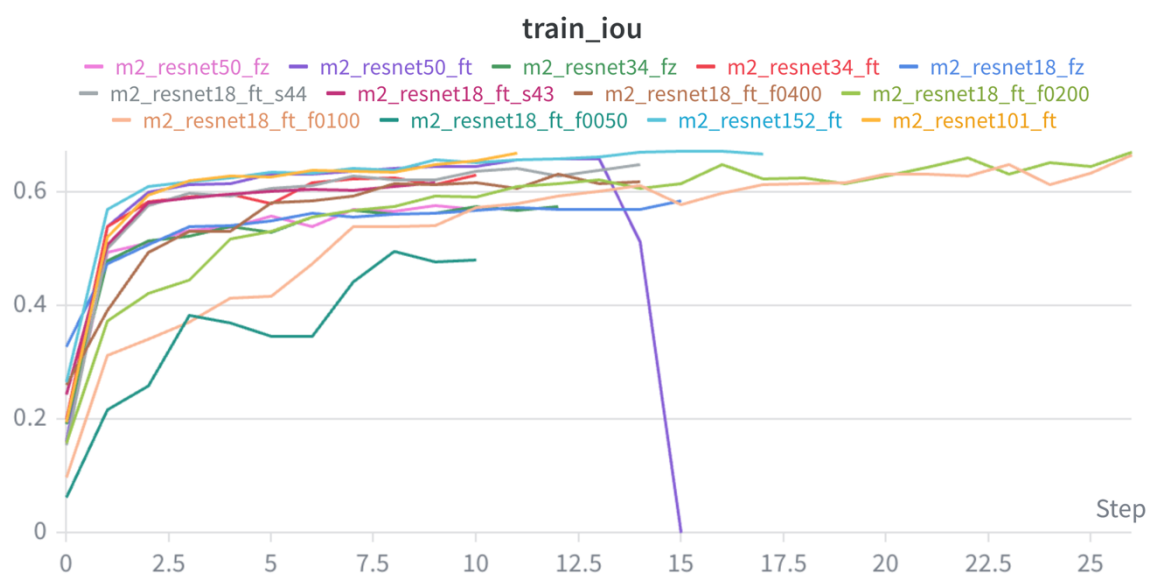


Appendix D.2 Fine-tuned CNN

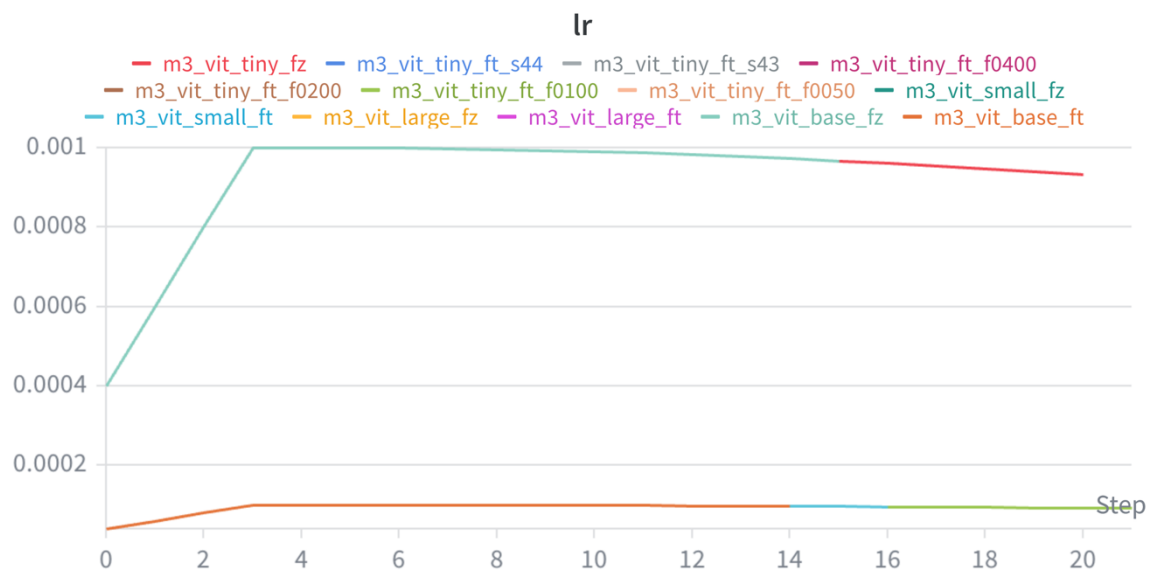
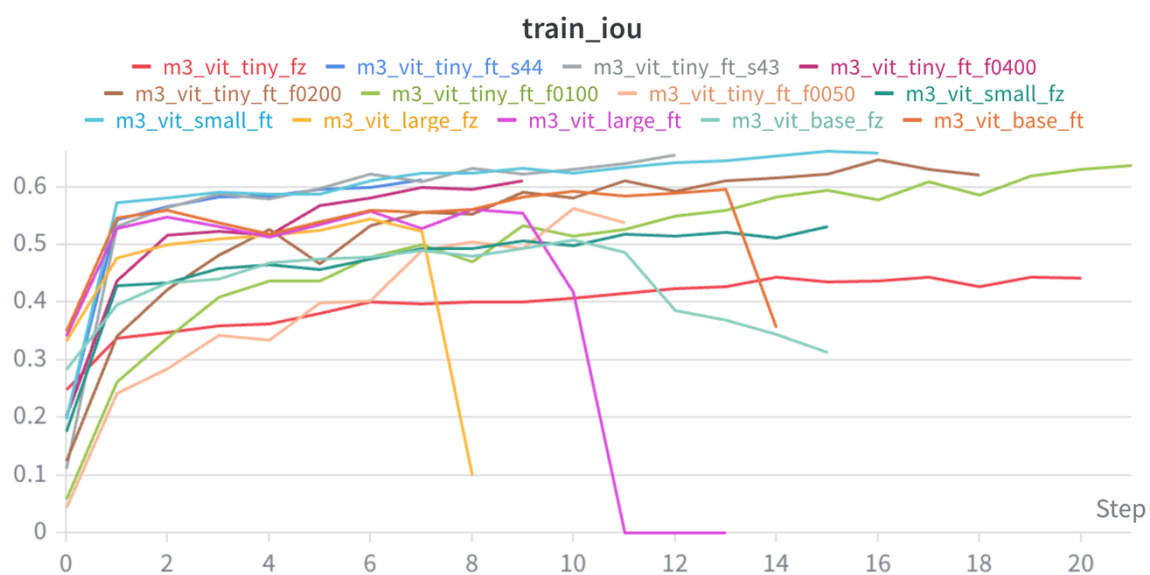


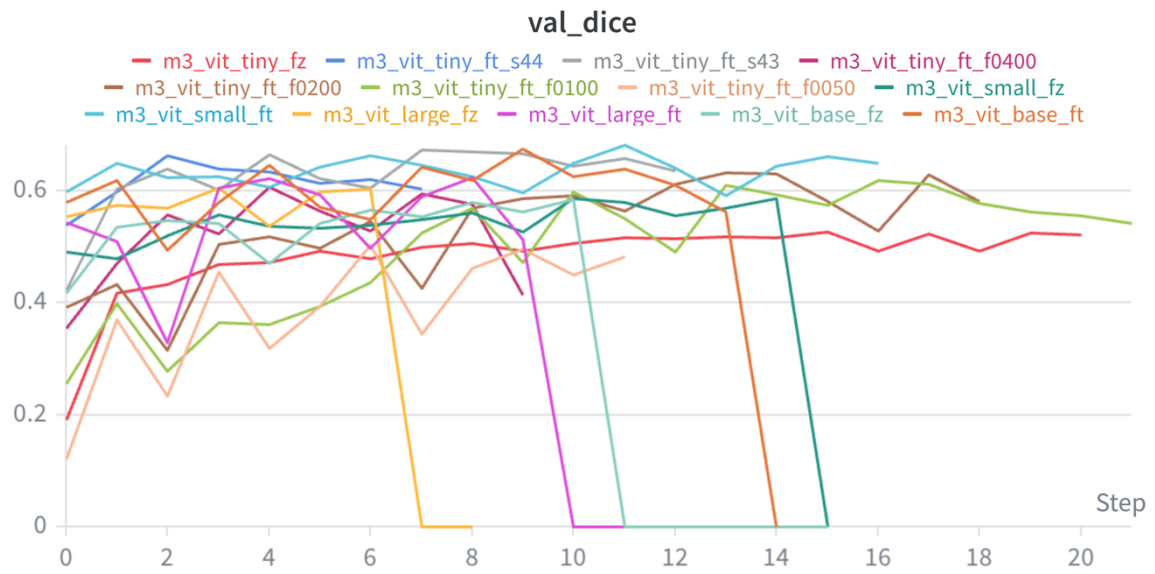


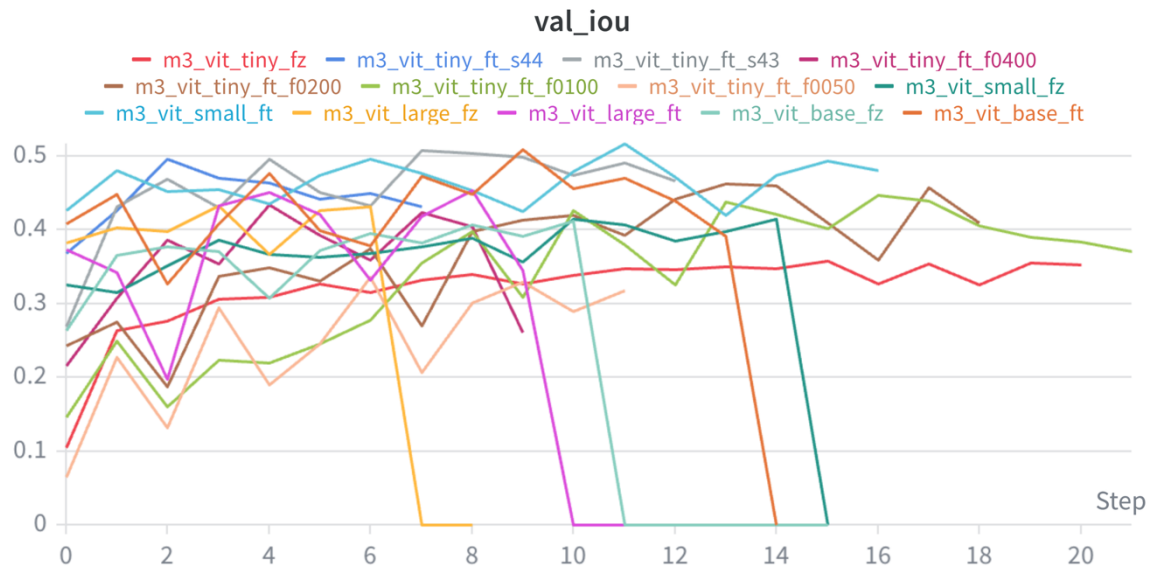
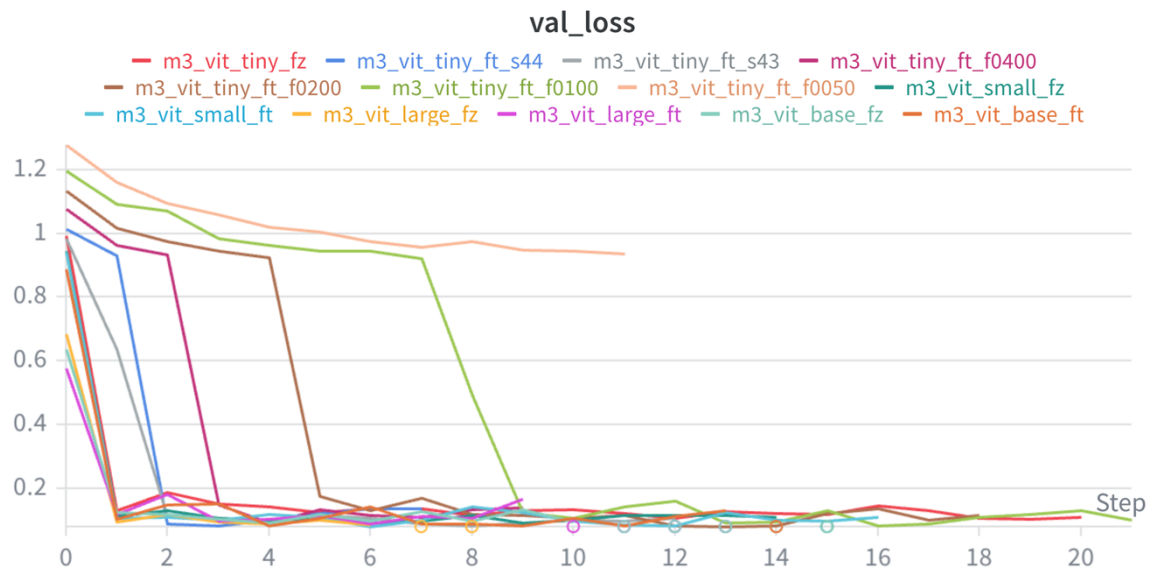


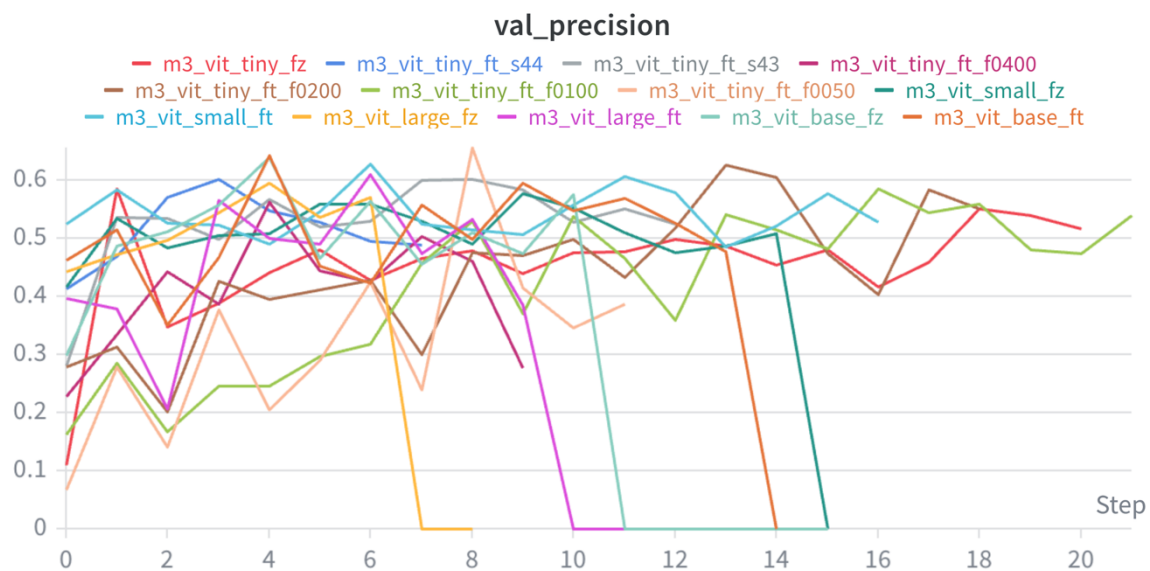
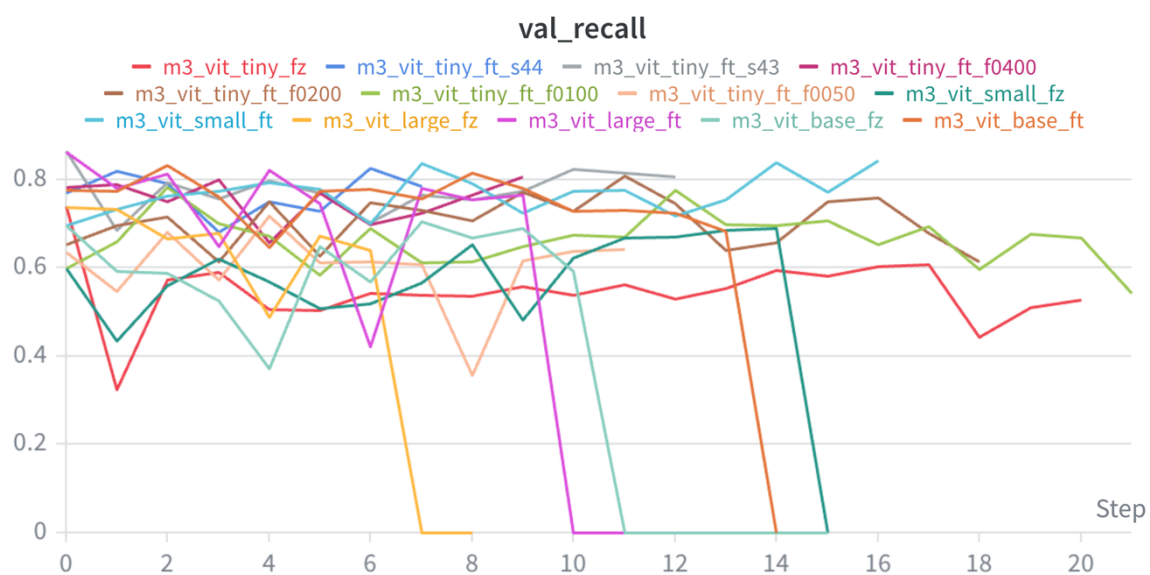


Appendix D.3 Fine-tuned ViT

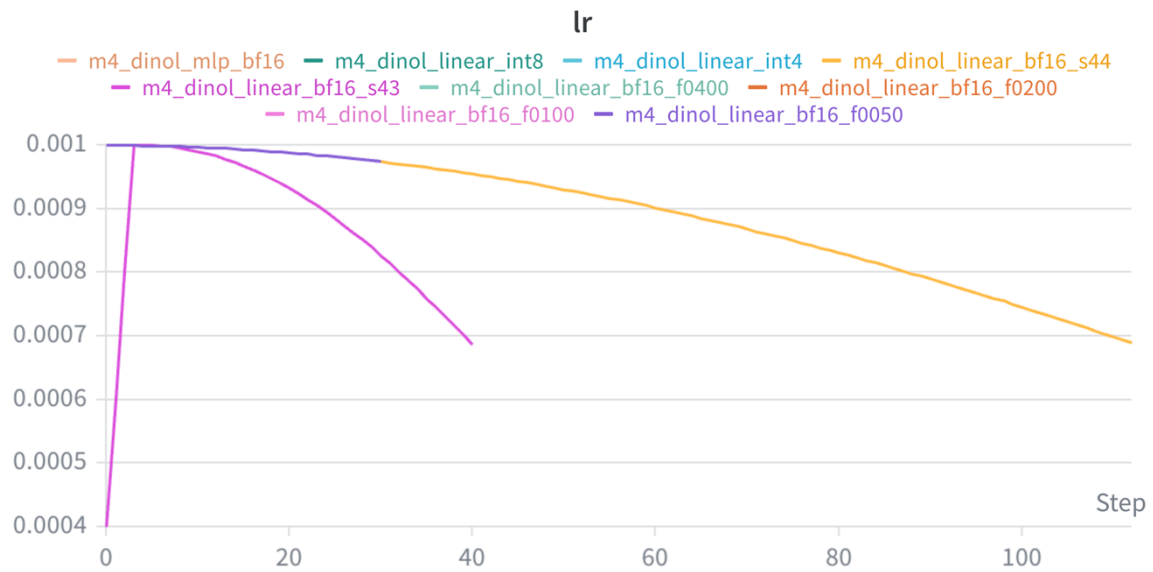
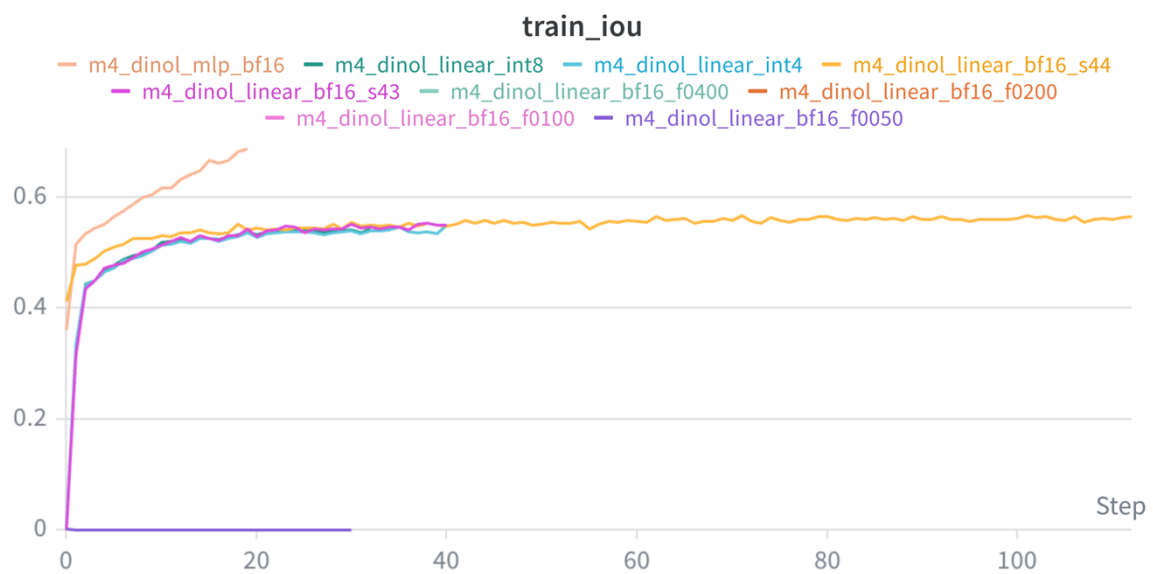


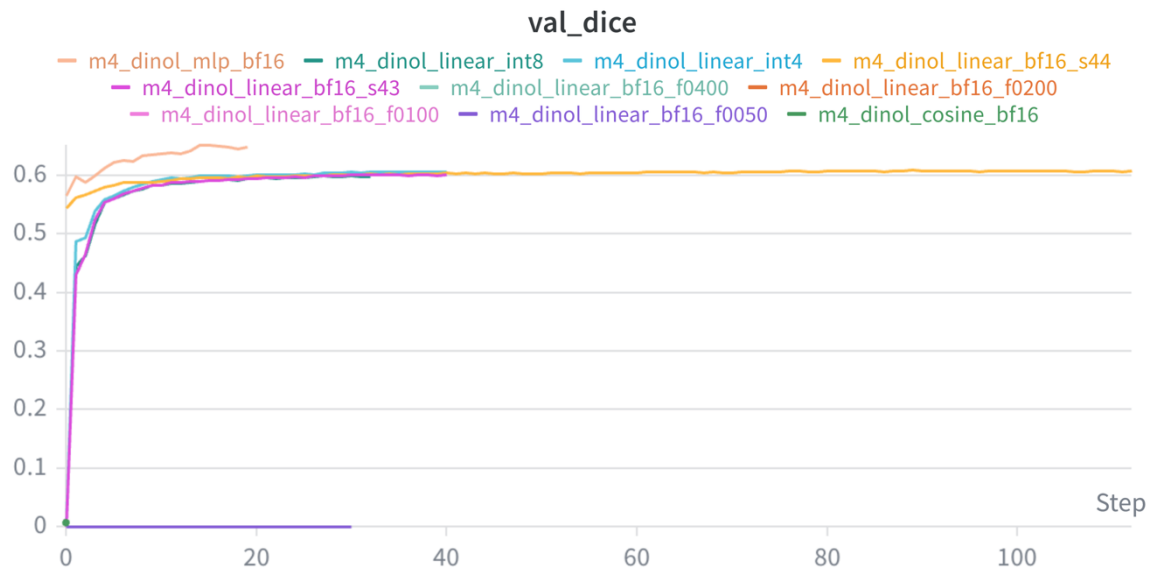


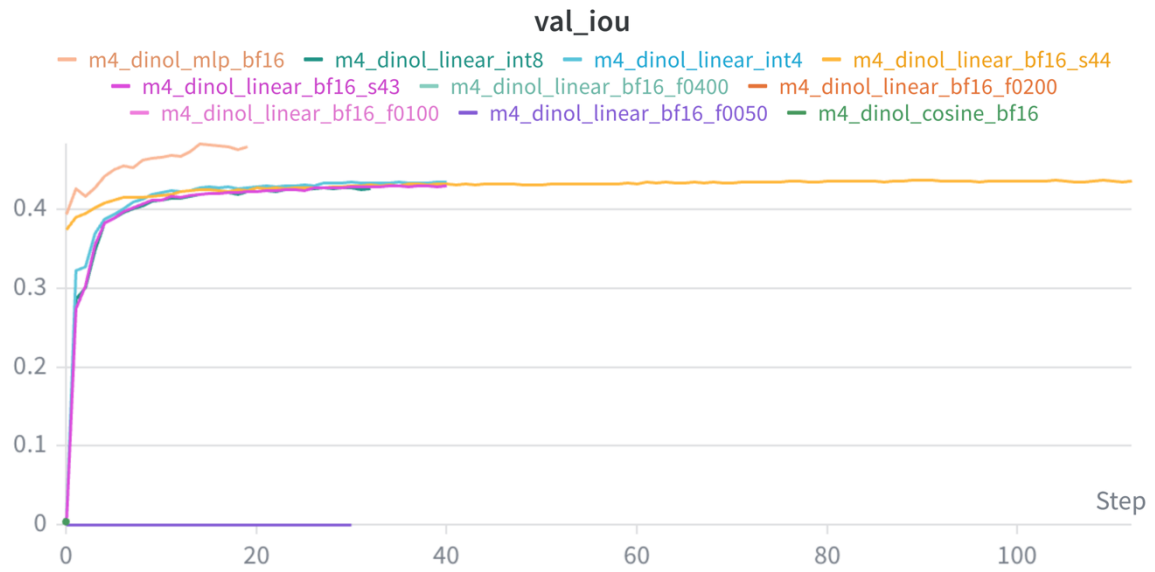
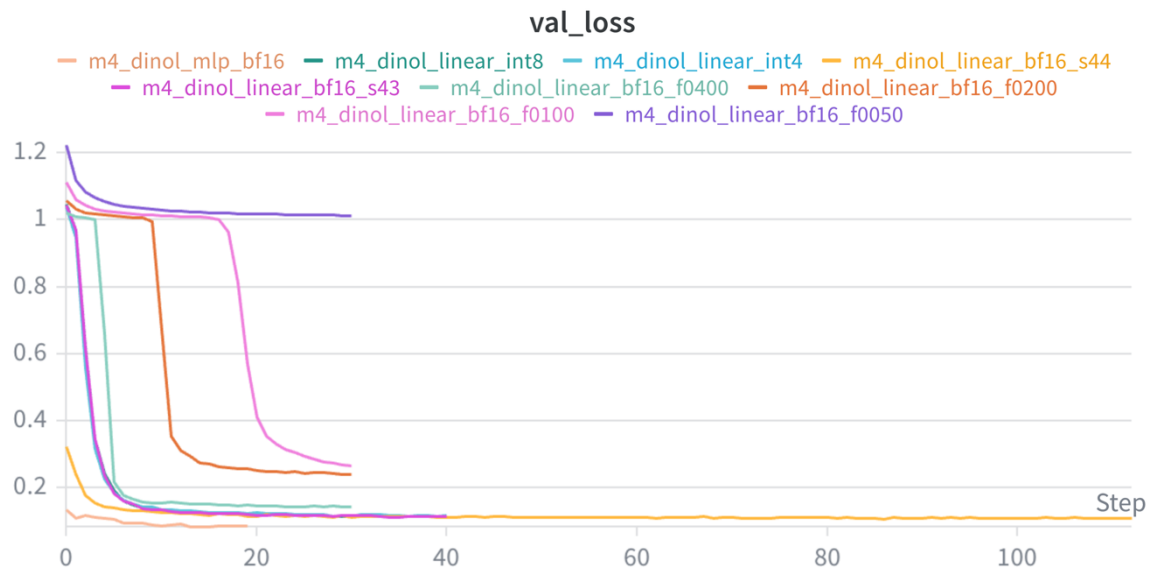


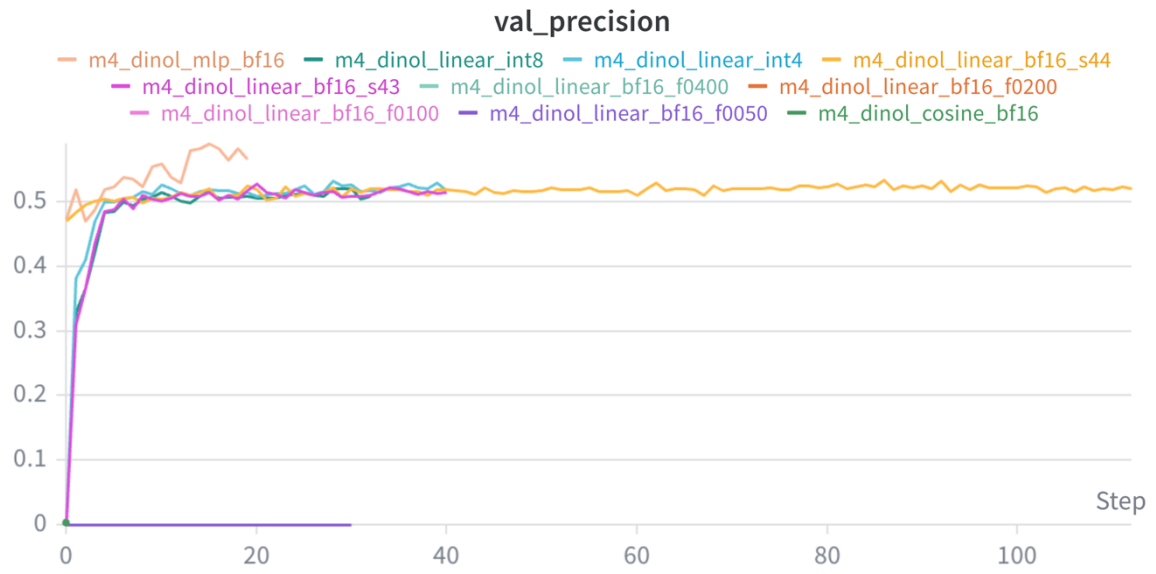
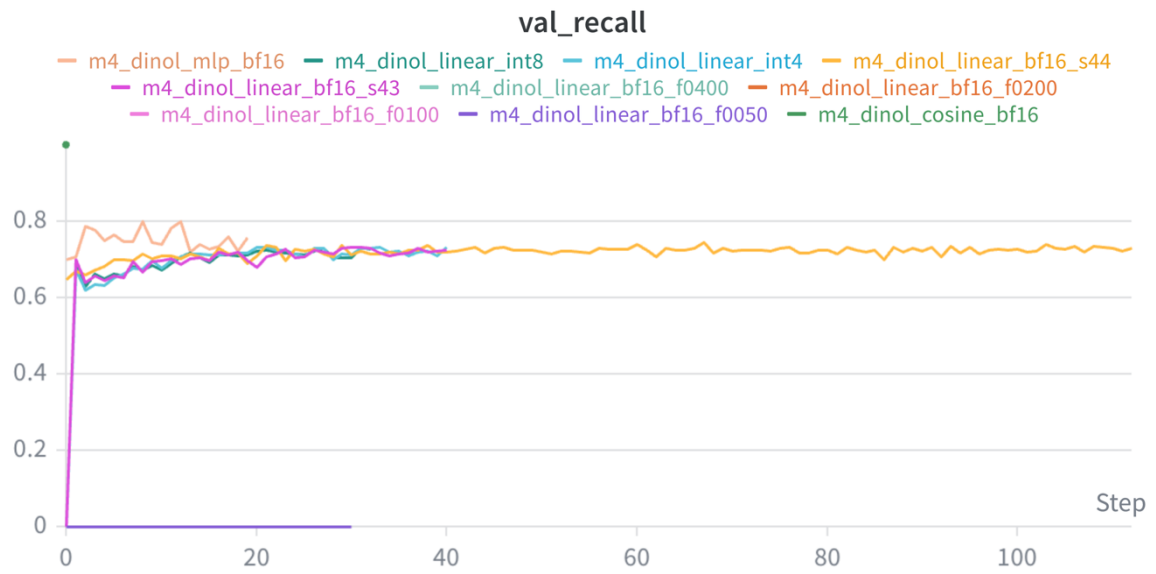


Appendix D.4 Dinov3-L

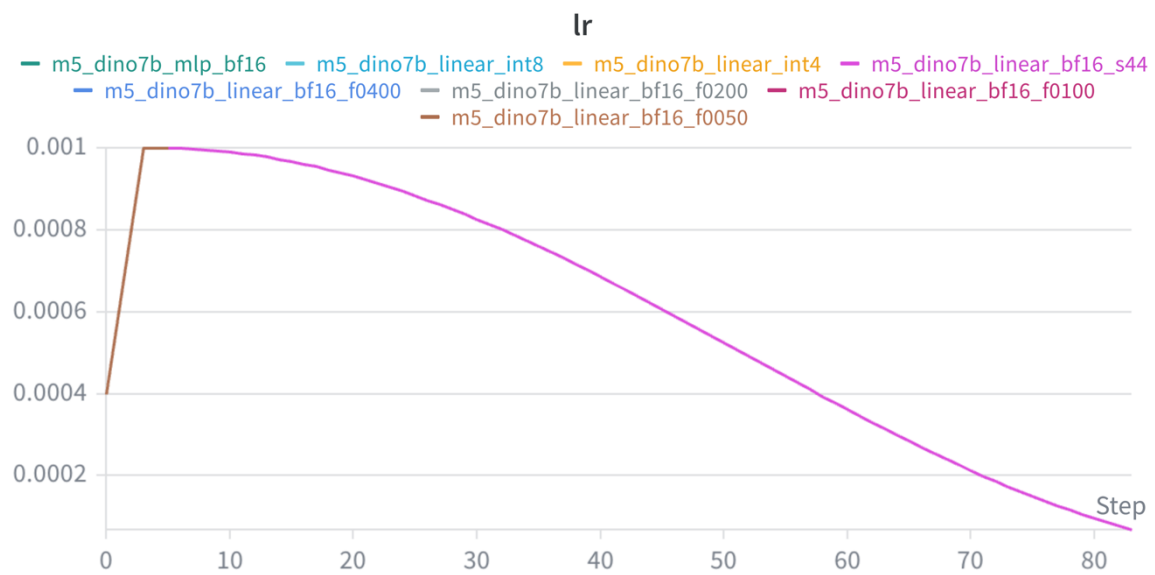
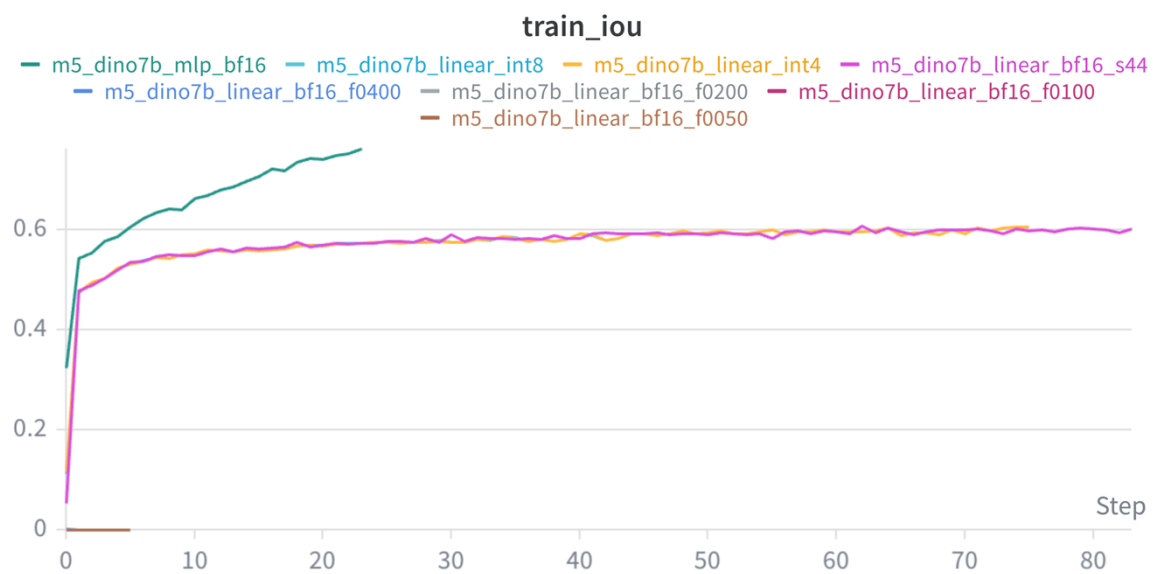


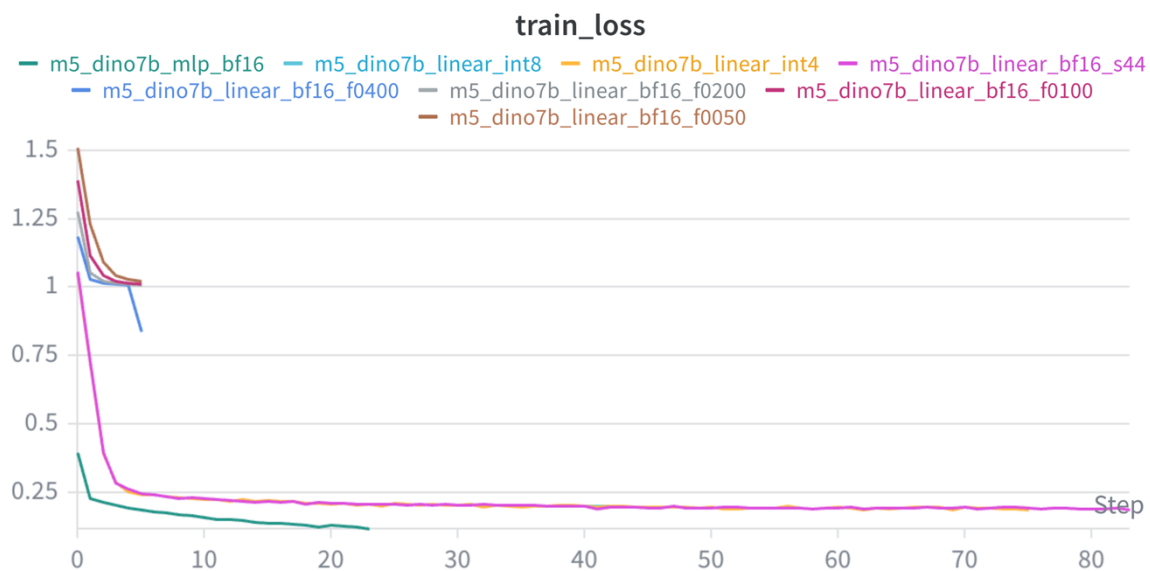


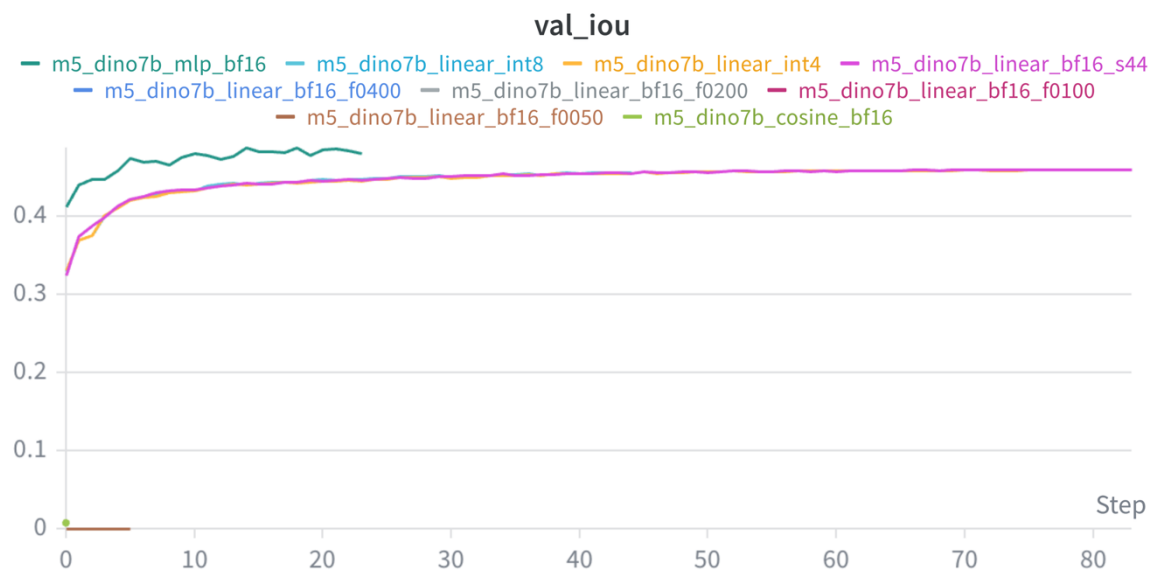
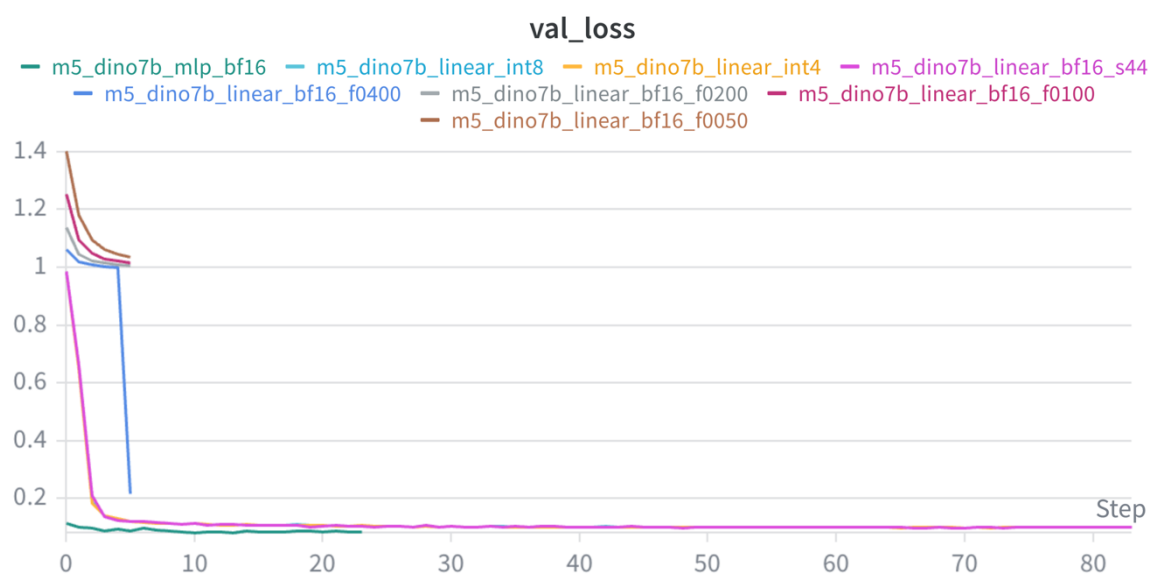


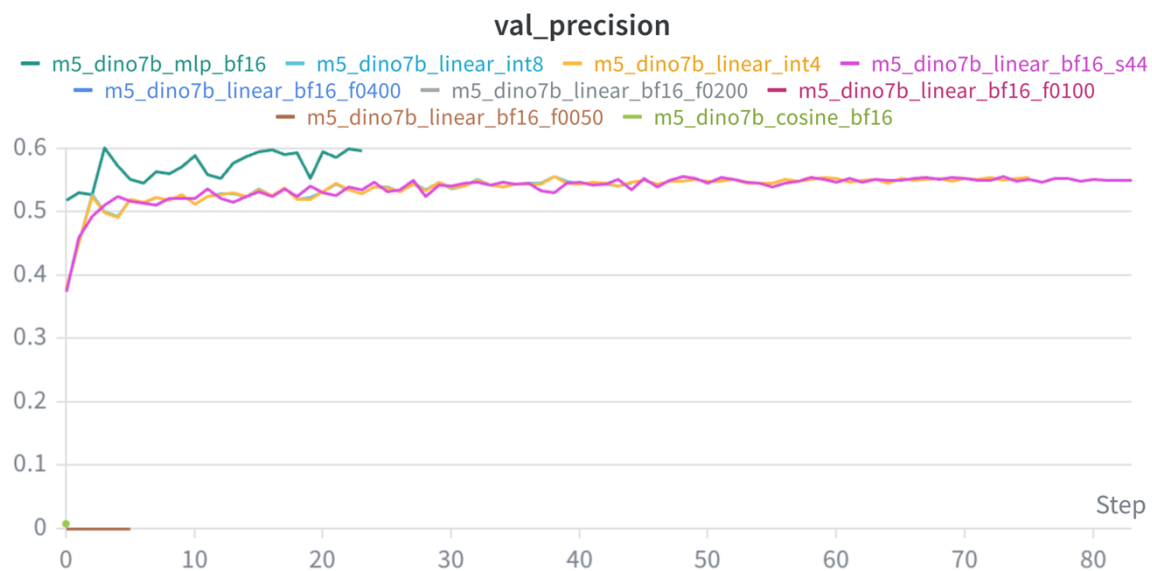
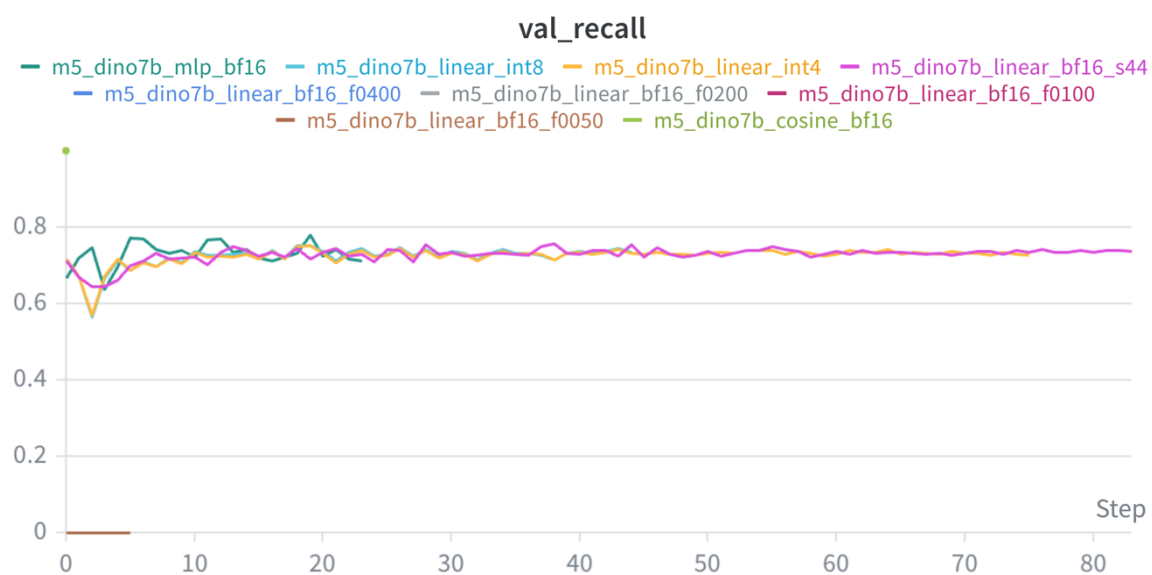


Appendix D.5 Dino v3- 7B









Appendix E: Hyperparameter definitions

Appendix E.1 CNN

Kernel size represents the size of individual kernels; it must be large enough to find the features you want to analyse, but small enough not to overly smooth them. A larger kernel requires more training data to work effectively. You can use a dilated convolution to identify larger features with the same kernel size.

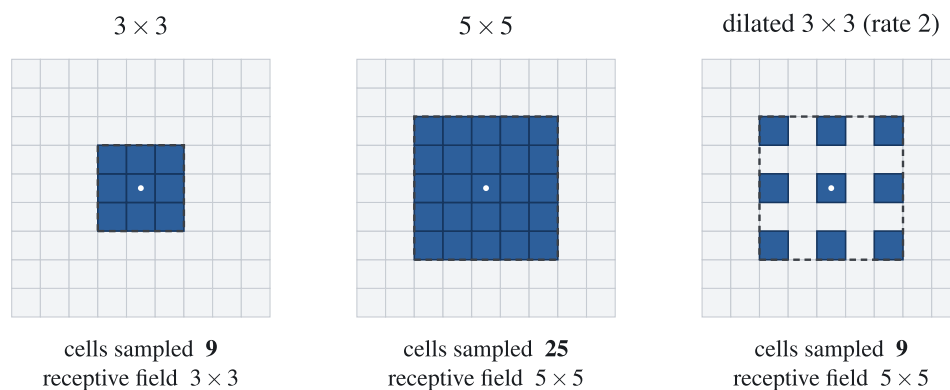
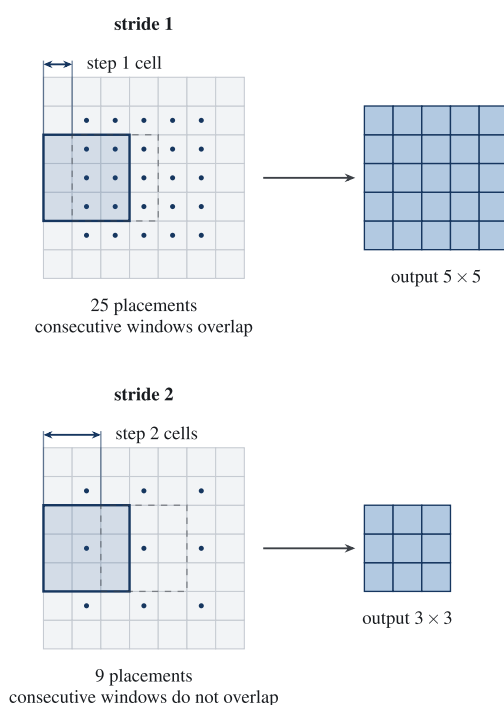


Figure E.1 kernel size and dilation diagram

Step size is the step distance a kernel is moved during convolution; a smaller step size produces a larger, more detailed output but is more expensive to compute; a larger step size may miss important features.



Each dot marks one kernel placement, so the dot lattice and the output grid always have the same shape.

Figure E.2 Step size diagram

Multiple kernels can be learned in parallel; the number of filters is the number of kernels that are used; more filters extract a greater number of features, and more complex ones.

Depth is the number of layers used; more layers enable the extraction of more abstract features and patterns.

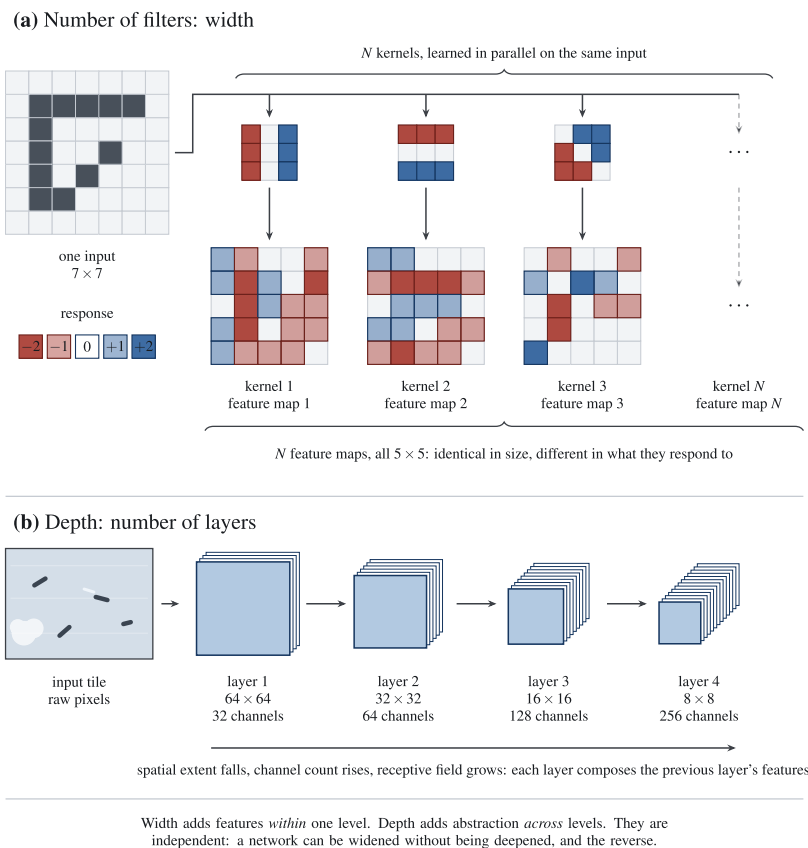


Figure E.3 width vs depth in CNN

Padding is the data added around the border of an input. Because you cannot have a null input to a kernel, the input output must always be smaller than the input. Padding is the process of adding extra data at the edges of an image to allow the model to process all parts of an image.

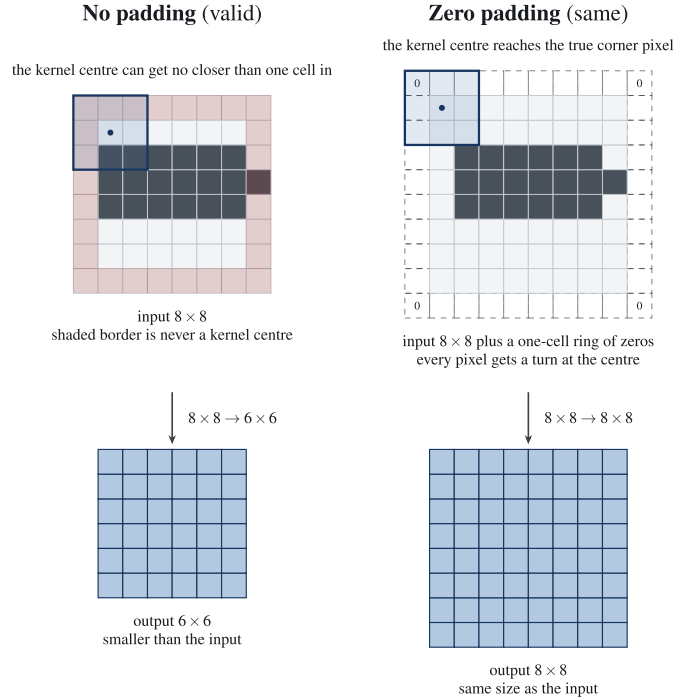


Figure E.4 Padding in CNNs

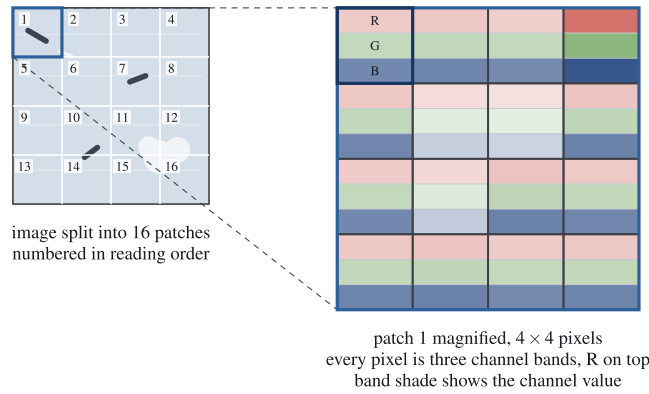
Appendix E.2 Transformers

The maximum number of tokens in the input to a model is determined by the model's context window size. In vision transformers, the total input size (vector dimensions * context window) equals the uncompressed image dimensions ($h * w * \text{colour channels}$); the size of token vectors is dependent on the embedding system. The depth (number of layers) of a transformer can be adjusted; a deeper model can model higher-complexity relationships, at the cost of training and inference.

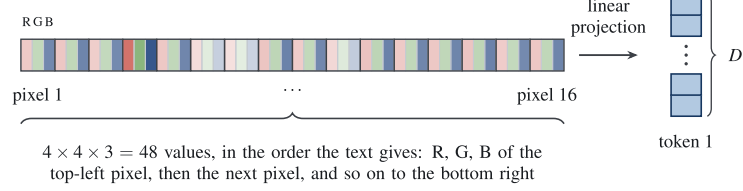
Multiple 'heads' can simultaneously model separate relationships through separate attention mechanisms; too few heads may underperform, as each head then has to model several relationships at once; too many may cause individual heads to perform poorly.

Transformers accept lists of vectors as inputs, so the image needs to be converted into a list of vectors; this can be done through 'patch embedding' (Dosovitskiy *et al.*, 2021). Patch embedding converts an image into a sequence of fixed-size patches, which are flattened and projected into a vector space (the values in the vector is composed of the red value for the top left pixel then the green value for the top left pixel, then the blue of the top left, then the red, then green, then blue of the next pixel, continuing for all the pixels in the patch, concluding with the red, green blue value of the bottom right pixel). This transforms the image into a token sequence, enabling Vision Transformers to process visual data using attention mechanisms.

1 tile the image



2 flatten, R then G then B, pixel by pixel



3 16 tokens into the transformer

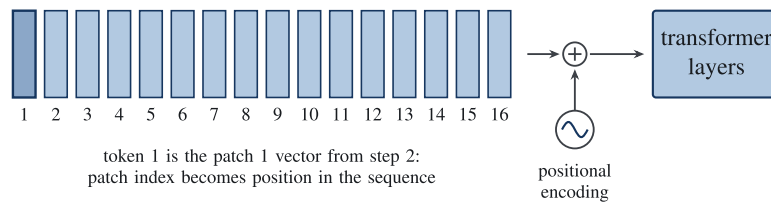


Figure E.5 Patch embedding steps

Appendix F: GAIT

Generative AI Transparency.

This is (to my recollection) an exhaustive list of AI tools used, and the types of uses of these tools. No content generated by AI has been presented as my own work, and all claims and thoughts are fully my own. With the exception of spelling and grammar correction, tool outputs are not used in this work.

Appendix F.1 Claude

(<https://claude.ai/>)

Claude is the primary LLM I have been using since January. It has been extremely useful for research and for automating slow, tedious tasks. This is obviously a non-exhaustive list of uses, but all my use falls under the umbrella of it helping me, not replacing me, listening to my instructions, and never making any decisions (I guess it's technically making decisions about what to tell me, what to show me, or how to explain something).

Examples of tasks given to Claude.ai:

- “Could you explain SNR in the Shannon-Hartley Theorem, I kind of understand but I don't really know where all the numbers come from, do you mind giving me the units of each number and its definition as you explain it.”
- “I have recently changed my reference manager from Mybib to Zotero, I have found at least 1 reference that was copied incorrectly. Could you go through and 1: check all citations are now done via Zotero references and not hand written, and 2: open every reference one by one and check everything was copied correctly. Air on the side of caution if there is something that you are only 99% certain of, please let me know so I can check it manually.”
- “I have had feedback that my glossary is probably missing some things that should be in there, can you give me a list of terms that are not in my glossary but should be in my latest draft (28th v9)”

Appendix F.2 Google AI Overview

(<https://www.google.com/>)

While I have not gone out of my way to use Google AI Overview, I do use Google, and as such am forced to interact with it.

Last search: "conclusion of my conclusion should be called? academic"

Appendix F.3 Grammarly

(Grammarly for Word - <https://www.grammarly.com/microsoft-word>)

Grammarly is a grammar and spelling checker. It now has AI integration, and at least some features are powered by LLMs, so for transparency, it's included here. At most, it can reorder sentence structure to make a sentence more grammatically correct and readable; it may drop a word or swap in a synonym, but it doesn't add any new information. Despite the fact that it has at least read every sentence in this dissertation, I think its effect is quite minimal and entirely positive.

It's also used as a litmus test for poor explanation: if the 'improved' rearranged sentence changes meaning, then that's an indicator to me that my original explanation was unclear and needs rewording.

Example of the use of Grammarly

- “Grammarly is a grammar and spelling checker, It does however now have AI integration, and at least some ferures are powered by LLM's, so in the intrest of transparency its included here.”

To

“Grammarly is a grammar and spelling checker. It now has AI integration, and at least some features are powered by LLMs, so for transparency, it's included here.”

Appendix G: Word count

- Word counts include: Quotations, tables, figures, footnotes, endnotes, in-text citations, main project title, titles, abstracts & summaries.
- Word counts exclude: Supplementary information such as tables of contents, tables of figures, tables of tables, the declaration page, the acknowledgements page, the reference list, or bibliographies. Material submitted in an annex or appendix is also outside the word count. However, all materials are subject to rules about academic integrity.

Word count: 16,279