MONASH University
Engineering

Unlocking Celestial Enigmas: AI Driven Exploration of the Universe

ENG4702: Final Year Project - Final Report

Author(s): Zach Drinkall (30596467), Katherine Hawkins (31561764), Muhammad Suleman (31991874), and Yide Tao (31491154)

Supervisor(s): Dr. Mehrtash Harandi

Date of Submission: 24/05/2024

Project type: Research

Executive Summary

Radio galaxies represent the most distant observable galaxies known to humanity, offering a unique window into the vast and dynamic evolution of the universe. Their study unveils crucial information about the structure of the cosmos and the intricate processes driving celestial phenomena. As modern radio telescopes generate data at unprecedented volumes, the traditional methods of manually identifying and classifying these galaxies have become increasingly untenable, hence driving a new demand for automatic data processing.

In our study, we aim to harness the capabilities of machine vision for the automatic classification, detection, and segmentation of radio galaxies. By implementing the YOLOv9 architecture on the RadioGalaxyNET dataset, we have surpassed the previous state-of-the-art benchmarks by over 15% in mean average precision, achieving a mAP50 of 81.7% on the testing dataset, showcasing the exceptional performance of YOLOv9 in processing complex astronomical data. Building on this success, we also explore both semantic and panoptic segmentation using the innovative Segment Anything Model (SAM), marking its first application as a foundational model for segmentation in astronomy. This dual approach has enabled us to set new benchmarks for segmentation tasks, significantly advancing our ability to precisely delineate and analyze the intricate features of radio galaxies, thus propelling the field of radio astronomy into a new era of precision and discovery.

Then, our research ventured into the frontier of computer vision, focusing on developing self-supervised learning methods—a methodology designed to allow machine learning tools to learn using new data without needing any human intervention. By employing the DINO pre-training technique, we sought to enhance the capabilities of pre-existing backbones that were originally trained on conventional datasets. Coupled with our innovative DARGN augmentation method, we have successfully developed a self-supervised ResNet-50 backbone. This advanced backbone was subsequently tested using a Faster R-CNN, where it demonstrated a notable performance improvement of 5-10% over backbones trained via traditional methods. This significant enhancement not only validates the effectiveness of self-supervised learning in this specific context but also establishes potentially new and innovative methods for developing the detection and segmentation backbones of radio galaxies.

In summary, our project marks a pioneering effort to implement advanced machine learning techniques tailored for astronomy. The insights from our work are set to drive further innovation and application of machine learning architectures for the field of astronomy, promoting their adaptation to a broader array of astronomical datasets and tasks. This combination of Astronomy + AI will enhance our understanding of complex cosmic phenomena and establish a new standard for integrating technology and science in exploring the boundaries of the universe.

Acknowledgement of Country

The authors of this report would like to acknowledge the Wurundjeri people of the Kulin Nation, the traditional owners and custodians of the land where this research was conducted. We pay our respects to Elders past, present and emerging. We recognise their connection to the Country and their role in caring for and maintaining the Country over thousands of years. Sovereignty was never ceded. It always was and always will be Aboriginal land.

Contents

1. Introduction

Radio-loud active galaxies or 'radio galaxies' are the most distant galaxies detectable due to large regions of radio-frequency emissions and luminous centres called active galactic nuclei [1]. These nuclei are thought to result from the accretion of matter by supermassive black holes in the centre of host galaxies [2]. The Fanaroff-Riley (FR) scheme classifies radio galaxies into two categories according to observed structure or morphology. An FR-I galaxy has two opposing jets that decrease in luminosity with increasing distance from the centre. An FR-II galaxy has two bright hotspots at the ends of opposing, separated lobes and a single jet.

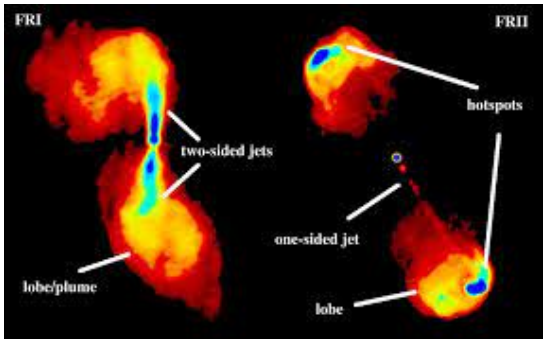


Figure 1 - Illustration of Fanaroff-Riley I (FR-I) and Fanaroff-Riley II (FR-II) classes [3].

Figure 2 - The radio galaxy "Alcyoneus" [4].

Recent studies have found sources with hybrid (FR-X) or unclassifiable (R) morphologies and more examples are needed to better understand the dichotomy of Fanaroff-Riley classes. Modern radio telescopes such as the Square Kilometer Array (SKA) seek to produce up to 11 exabytes of data per day with increasing depth of field and resolution [5]. Data on this scale can no longer be manually inspected and requires robust machine learning algorithms to process it.

Figure 3 - The Square Kilometer Array (SKA) [6].

Machine learning solutions for radio astronomy are non-trivial due to the scarcity of labelled data and the unique nature of images from radio astronomy. Two relevant machine learning techniques are (1) the detection and categorisation of galaxies with bounding boxes, and (2) segmentation to delineate galaxy boundaries against the background. The recent release of the 'RadioGalaxyNET' dataset [7] allows us to use these methods for the automated identification of the extended components of radio galaxies. This is

necessary to model galaxy evolution, measure the geometry and expansion rate of the universe, and build large informative catalogues of radio galaxies.

An emerging paradigm in computer vision concerns the pretraining of large foundation models on huge datasets for generalisable performance on downstream tasks across domains. These datasets are composed of natural images starkly dissimilar to radio imagery (*Figure 4*) and the usefulness of such foundation models for radio astronomy is not well understood. Alongside task-specific algorithms for the detection and segmentation of radio galaxies, we evaluate existing foundation models for these tasks as well as propose our own. This helps us comment on the general usefulness of pretraining for radio imagery.

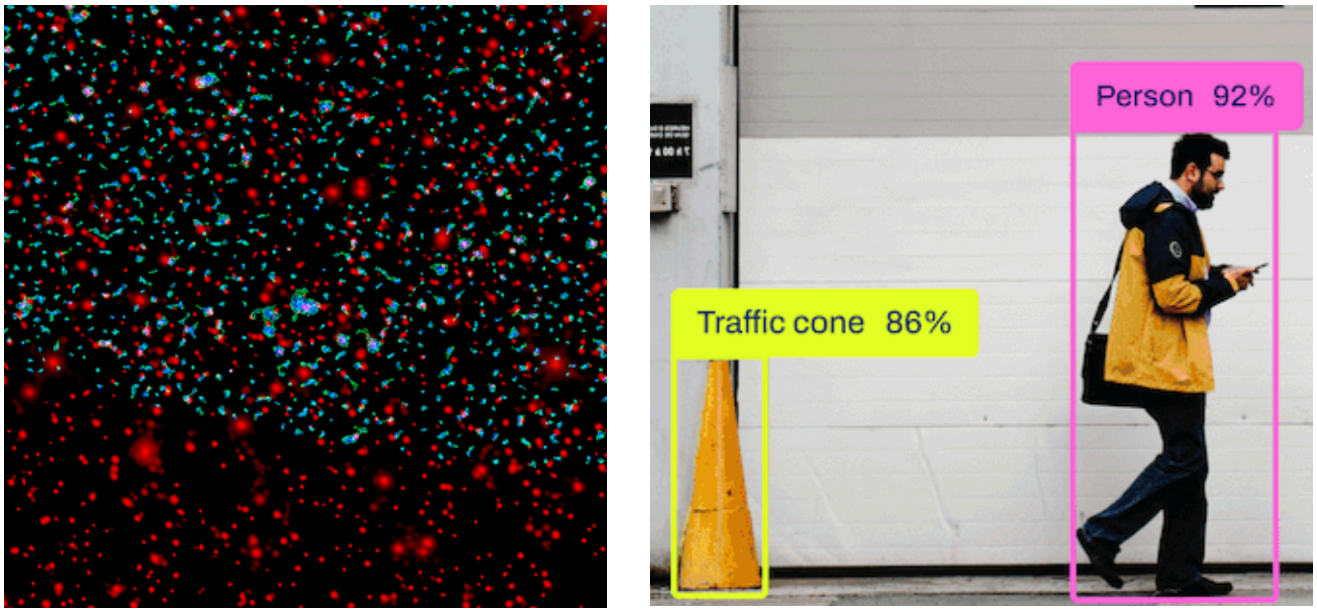


Figure 4 - An example of the RadioGalaxyNET dataset [7] (left) and a natural image typically used for object detection (right) [8].

2. Literature Review

2.1 Fanaroff-Riley (FR) Classification

Currently, radio galaxies are classified under the Fanaroff-Riley (FR) Classification system [9]. Two of these categories are Fanaroff-Riley Class I (FR-I) and Fanaroff-Riley Class II (FR-II). Whether a galaxy is FR-I or FR-II depends on the galaxy's *a* to *b* ratio, where *a* represents the distance between the two emission peaks of a galaxy and *b* is the length of the asymmetric jets emitted from its centre [10]. If *a/b* is determined to be less than 0.45, the galaxy is classed as FR-I, and if *a/b* is larger than 0.55, the galaxy is classed as FR-II [11]. Any galaxy with an *a/b* value between 0.45 and 0.55 is too close to the threshold to be accurately classified and is assigned to the third class, FR-X [10]. The final class is used where only one or no emission peaks can be detected outside of the central component, therefore, *a* is 0. These galaxies are classified as R or Resolved, as their FR class cannot be determined until higher-resolution data is collected.

2.2. Fundamental Architectures

2.2.1 Linear Layers

A linear layer refers to a set of output neurons that compute the weighted sum of all input neurons using a different set of weights per output neuron [13]. The Universal Approximation Theorem states that a network composed of cascaded linear layers can approximate any mathematical function given: (1) a nonlinear transformation between layers, and (2) the right set of weights [14].

2.2.2 Convolutions

A convolution uses a sliding kernel composed of learnable weights to compute the weighted sum of the input across small patches. Stacked convolutions with nonlinear transformations between layers can extract rich features from images in a hierarchical fashion similar to the visual cortex [15]. Convolutions are sometimes preferred for computer vision tasks as they encode the importance of neighbouring image regions as an inductive bias [16]. They also require fewer parameters than linear layers when processing images as a separate set of weights need not be learned for each pixel.

2.2.3 Attention

A Vision Transformer (ViT) is an emerging architecture inspired by the success of Language Models (LMs) such as the Generative Pretrained Transformer (GPT). It uses the 'attention' mechanism to capture relationships between patches in an image similar to words in a sentence. This mechanism involves computing the vector dot-product as a measure of semantic similarity [17]. A ViT does not encode the greater importance of neighbouring image regions as an inductive bias [18]. It can sometimes outperform convolutional networks as it learns long-range dependencies and global context within an image. This often requires a significantly larger dataset, deeper networks, and longer training times.

2.2.4 Foundation Models

An emerging paradigm in machine learning is the pretraining of 'foundation models' on large amounts of data for generalizable performance across downstream tasks. This technique is called 'transfer learning' and leverages similarities across all datasets to boost performance on smaller datasets [19]. As such, these models achieve superior performance on smaller datasets than task-specific networks at the expense of an

initial computational overhead. Popular architectures used in this paradigm include Residual Networks (ResNets) and ViTs.

2.3. Object Detection

Object detection refers to the localization and categorization of objects in an image with bounding boxes and labels.

2.3.1 Region-Based Convolutional Neural Networks (RCNNs)

Region-Based Convolutional Neural Networks (RCNNs) are a popular family of architectures for the detection task that use Region Proposal (RP) methods in conjunction with convolutions [20]. An RP identifies 'regions of interest' or regions in the image likely to contain objects. These regions are processed by a Convolutional Neural Network (CNN) to output a category and confidence score. This method enables the production of a variable number of output bounding boxes using a fixed CNN architecture [21]. Some classical methods to obtain regions of interest include the use of unsupervised clustering algorithms such as k-Means, superpixels, and heuristics regarding texture, colour, etc. The Faster-RCNN improves the performance of traditional RCNNs through a CNN called a Region Proposal Network (RPN) that identifies regions of interest [20]. This is significantly faster than non-parameterized methods.

2.3.2 You Only Look Once (YOLO)

YOLO (You Only Look Once) is a common object detection and classification model developed in 2015 by Joseph Redmon and Ali Farhadi [22]. YOLOv1 quickly gained popularity due to its incredible speed and accuracy, capable of generating bounding boxes in a single step. YOLO's speed and accuracy made it ideal for real-time detection such as in videos or real-time surveillance.

Since 2015, YOLO has been uploaded as open source software by the company, Ultralytics, where YOLO has been updated many times. While, officially, YOLOv8 is the most recent version available, an unofficial YOLOv9 has been developed and released by Wang et al. [23].

This YOLOv9 is an incredibly new model, published in February 2024, and is based upon the YOLOv5 model. YOLOv9 outperforms all previous train-from-scratch YOLO models, setting a new benchmark on the MS COCO dataset [24].

The updated YOLO model proposes and implements the concept of programmable gradient information (PGI). PGI is a novel method for overcoming what is known as an "information bottleneck principle", an issue that occurs in deep learning models whereby the more layers a model has, the less of the original information is preserved [25]. PGI seeks to create more reliable gradients using an extra pathway, known as an auxiliary branch, that can reverse its steps. This helps the network retain important details while training whilst also being cost-efficient and flexible, allowing the model to choose the best loss function for the specific task the model is trying to perform. Additionally, YOLOv9 [23] implements a generalised version of the ELAN (Efficient Layer Aggregation Network) architecture proposed by Wang et al. in an earlier paper [26] to achieve efficient parameter utilisation to reduce computational requirements, in a new architecture they call GELAN.

These innovations working in tandem result in an incredibly powerful and lightweight model. There are several different architectures for YOLOv9 which differ in accuracy and size, the two main models being

yolov9-c and yolov9-e. While yolov9-e performs slightly better on the MS COCO dataset than yolov9-c, it requires twice as many parameters [23].

Unfortunately, the YOLOv9 model from Wang et al. is incompatible with the current Ultralytics packages and requires custom scripts for training and testing. Therefore, Ultralytics adapted the architecture of Wang et al's YOLOv9 into yolov9c and yolov9e, slightly different architecture files that carry the same naming conventions as the originals. These models are capable of being trained, validated, tested, and fine-tuned using the open source Ultralytics functions, facilitating the training of models from scratch. Ultralytics adapted the work of Wang et al. further, developing YOLOv9 panoptic segmentation models known as yolov9c-seg and yolov9e-seg which are also compatible with Ultralytics packages. Despite this, almost all documentation states that YOLOv8 is still the most recent Ultralytics YOLO model.

2.3.3 Other Methods

Although we are primarily focusing on evaluating RCNNs and YOLOv9 for object detection, three other methods were studied to analyse their advantages and limitations for object detection. Firstly, SpineNet which has a scale-permuted backbone [27], meaning connections are made across different scales, allowing for simultaneous recognition and localisation. Secondly, Copy-Paste, which copies objects from one image to another to then create new training data [28]. Finally, TridentNet, which has an architecture of branches with the same transformation parameters, but different receptive fields [29].

Table 1 : Object Detection Model Comparisons

Model	Advantages	Drawbacks
YOLOv9	Performs detection in a single network pass, making it very fast. Achieves competitive accuracy results, particularly in comparison with its speed.	Lower level of accuracy compared to slower models. Can struggle to detect small objects, due to the down-sampling process.
RCNNs	Used as the foundation for many models, meaning it is well-established. Easy to use. Can use convolutional networks for classification.	A lot of time is spent on the training phase. Duplicated computations. Cannot be used in real time as one image takes up too much time. Does not involve an end-to-end training pipeline.
SpineNet [27]	Has the ability to be used for image classification and real time detection. Due to the scale-permuted backbone, it has great accuracy.	Training takes a lot of time.

Copy-Paste [28]	Can be easily integrated into instance segmentation. Great accuracy.	The model struggles to select realistic data due to the random nature of its data selection.
TridentNet [29]	Deals with scale variation. Produces large receptive fields, which allows for detection over a wide area.	Produces a very slow model.

2.4. Image Segmentation

Image segmentation seeks to delineate the boundaries of objects in an image. More specifically, semantic segmentation also identifies object categories but cannot distinguish between instances of the same category. Instance segmentation identifies object instances but is unaware of semantic categories. Panoptic segmentation identifies both instances and categories [30]. This work is concerned with semantic segmentation methods that learn from the 'supervision' of ground truth masks [31]. This particular field is evolving and less established than object detection. Many of the models studied are novel and there is opportunity for great growth in this area.

2.4.1 The U-Net

Fully-Convolutional Networks (FCNs) are a popular choice of architecture for the segmentation task as the convolution operation can preserve spatial dimensions to predict masks of the same size as the input [32]. Two challenges to the use of stacked convolutions are: (1) the need of large receptive fields for effective segmentation, and (2) the difficulty of identifying object boundaries from the high-level semantic features that stacked convolutional networks extract [33]. A receptive field refers to the portion of the input image that a convolutional kernel directly or indirectly operates on [34]. Segmentation often requires large receptive fields as identifying the object category of a pixel can be difficult without context from surrounding image regions. However, the receptive field of a stacked convolutional network is proportional to the number of convolutions and larger networks are slow and unstable to train.

The U-Net is a symmetric encoder-decoder architecture designed to address these challenges [35]. It uses a pooling operation to reduce the spatial dimensions of features between convolutional layers in the encoder. This artificially enlarges the receptive field of subsequent convolutions without the overhead of additional parameters. The predicted mask is constructed by iteratively upscaling the final features with the decoder. This prediction is informed by both high-level semantic features and low-level image features using a 'skip connection' that connects each layer of the encoder to the opposite layer of the decoder.

Some research investigates improvements to the convolutional blocks that compose the layers of a U-Net. Guan et al. propose to connect the input of each convolutional layer with the output of all previous layers [36]. Similarly, Zhou et al. connect each layer of the decoder with all previous layers of the encoder [37]. Such a 'dense connection' scheme allows the extraction of richer features at the expense of computational cost. Gu et al. propose an 'inception' block that extracts features at different scales using parallel branches with a varying number of convolutions [38]. He et al. propose the 'residual' block that combines the input of a convolutional layer with its output using skip connections [39]. Residual and inceptions blocks excel at

combining features at different scales. Skip connections are noted to improve the stability of training deep networks. They protect against vanishing and exploding gradients by providing an alternative pathway for gradient backflow.

2.4.2 Segment Anything Model (SAM)

The Segment Anything Model (SAM) is a powerful foundation model for segmentation that supports prompts such as text, bounding boxes, rough masks, and points for interactive segmentation [40]. It's composed of a large Masked Auto-Encoder (MAE) that extracts features from images and a lightweight decoder network that constructs predictions from features. An MAE is a ViT that learns rich features by reconstructing masked images. It has a large receptive field as it learns associations between all parts of an image at once without the computational overhead of stacked convolutions. SAM is pre-trained on 11M natural images and 1B masks for universal generalizability [40]. It is unusually fast given its size as the large encoder runs only once per image. It is also ambiguity-aware and returns multiple predictions corresponding to the parts and subparts of the segmented object. SAM has been widely applied to medical image segmentation; a visually similar domain to radio imagery that shares challenges such as blur and noise.

SAM can be adapted for automated segmentation by fine tuning the parameters of the lightweight decoder. The encoder may also need to be finetuned for images very dissimilar to the naturalistic scenes used during pre-training. This requires a prohibitive amount of computational resources given the size of the encoder. Low Rank Adaptation (LoRA) addresses this challenge by decomposing the matrix of learned parameters into two smaller matrices using low rank approximation [41]. The size of these matrices is determined by the rank parameter r. This decreases the number of trainable parameters while preserving key information and can be used to finetune large networks with limited resources. Zhang & Liu successfully finetuned the image encoder of SAM for the automated segmentation of medical images with promising results [42].

2.4.3 Loss Functions

The popular choice of loss function for segmentation is Cross Entropy (CE) loss which compares a vector of class probabilities to the ground truth for each pixel [43]. Two situations in which a different loss may be preferred are: (1) the segmentation of small objects, and (2) an imbalanced distribution of object categories. The segmentation of small objects is challenging as 'background' pixels occur with significantly greater frequency than 'object' pixels. An imbalanced 'background-foreground' distribution or an imbalanced object category distribution can lead to a biased network that overpredicts majority classes. Such a network can minimize loss without learning something generalizable. Weighted CE (WCE) loss combats imbalance by assigning greater weight to minority classes using an α-parameter [44]. Focal loss extends and improves WCE on low confidence predictions by downweighting high confidence predictions using the γ-parameter [45]. Dice loss is well-suited to small objects as it directly optimizes the overlap of predicted and true masks rather than pixel-level classifications [46]. However, it can result in unstable training from sensitive gradients. The standard method of combining loss functions is through a weighted sum.

2.4.4 Other Methods

Again, we have explored the advantages and drawbacks of other common methods, this time in the field of image segmentation. Firstly, Fully Convolutional Networks (FCN), which consists of solely locally connected layers that require less parameters than those with dense layering [47]. Secondly, Deeplab, which

up-samples the output of the last convolutional layer and then computes pixel-wise loss [48]. Finally, Recurrent Neural Networks (RNN), which uses previous outputs as its input [49].

Table 2: Image Segmentation Model Comparisons

Model	Advantages	Drawbacks
U-Net	Produces accurate segmentation maps. Excellent at handling multi-class datasets. Efficiently uses training data, as it can incorporate high-level and low-level features from input images and skip connections.	It is prone to overfitting. Due to its ability to skip connections, it requires a lot of parameters. Has high-computational cost. Can be sensitive to initialisation of parameters.
SAM	Flexible in terms of the datasets it can be used on. Requires no to minimal retraining. Has a large receptive field, better for large-scale images.	Issues can arise when scaling to higher-resolutions. Can struggle with images of low contrast.
Fully Convolutional Networks (FCN) [50]	Can be end-to-end trainable. Little restraint on input dataset size, predictions can be made on smaller datasets.	Direct predictions are relatively low-resolution, making blurred object boundaries.
Deeplab [51]	Great efficiency and accuracy. Simple, as it is composed of two well-established models.	Smaller receptive field that does not scale well to larger-scale images. Relatively slow model.
Recurrent Neural Networks (RNN) [52]	Efficient to train. Can process inputs of any length.	Slow computation. Gradients can explode.

2.5. Self-supervised Learning

As outlined in "Multimodal Scene Understanding" [53], self-supervised learning is a subset of machine learning methodologies that empowers deep neural networks to identify key features in a dataset without the need for manual labelling. This approach is commonly adopted in the pretraining process of modern foundation models to reduce labelling cost, mitigate inductive bias and enhance generalisability of predictions [54], exemplified by GPT-3.0 [55], which incorporates self-supervised learning in its pre-training process.

Common methods for self-supervised training on image tasks can be predominantly categorized into two main categories. The first category includes contrastive learning methods such as SimCLR [56] and BYOL [57], which focus on feature learning by exploring the similarity and dissimilarities between images. The second category involves knowledge distillation methods such as DINO [58] and KDFM [59], which aim to train a smaller, more efficient model by mimicking the behavior and predictions of a larger, pre-trained model.

Despite the superior performance of DINO relative to other self-supervision [60] methodologies, it is also recognized for its significant demand on training resources. According to the official paper, DINO underwent pre-training on 14 million images utilizing eight GPUs across a span of three days. This underscores the necessity for additional research into the realm of self-supervised fine-tuning, specifically on task-specific images, leveraging pre-existing model weights.

2.6. Existing Solutions

Gupta et al. evaluate numerous object detection algorithms for the RadioGalaxyNET dataset including DETR, Faster-RCNN, and the older YOLOv8 [7]. They observe that YOLOv8 and certain DETR variants outperform the Faster-RCNN. YOLOv8 displays an advantage over DETR for detecting small galaxies. This may be because the ViT in DETR computes relationships between large image patches that frequently miss the object of interest. A limitation of their work is that the choice of hyperparameters is not clearly justified and some networks appear undertrained. Furthermore, they initialize their networks randomly rather than with pretrained weights. However, recent research in cross-domain fine-tuning suggests that the use of pretrained weights from natural image datasets such as ImageNet can improve performance on downstream tasks [59][60]. Such an improvement has also been demonstrated in the Natural Language Processing (NLP) field [63]. Further research is needed to explore whether self-supervised pre-training can improve performance for radio imagery tasks.

Numerous studies investigate the feasibility of self-supervised pretraining to improve performance on scarcely labelled radio imagery datasets [64][65][66]. Slijepcevic et al. show that pre-training on a dataset of 100k radio galaxy images improves classification accuracy even across sky surveys. Their 18-layer Residual Network (ResNet) learns rich features corresponding to structural properties of the galaxy. They show that increasing the number of parameters in this network does not improve performance. However, it must be acknowledged that the classification of their dataset is not inherently complex (as shown in "Appendix A1"). Their results may not generalize to more complicated tasks such as detection and segmentation but are a useful baseline for other networks. Another limitation of their work is the use of the older Bootstrap Your Own Latent (BYOL) method for pre-training that has been outperformed by MAE and DINO.

Gupta et al. evaluate the use of weak labels for the segmentation and detection of radio galaxies in a dataset with identical properties to RadioGalaxyNET but fewer samples [67]. Their choice of architecture is a

50-layer ResNet. However, their method is too complex to generalize across datasets and they do not directly leverage self-supervised pretraining or transfer learning. Consequently, a significant literature gap exists for the development of a generalizable pretraining pipeline applicable to radio imagery.

Tang et al. utilized a 13-layer Deep Convolutional Neural Network (DCNN) that uses stacked convolutions to achieve an impressive 90% accuracy when categorizing galaxies [68]. Transfer learning enabled the DCNN to utilize pre-trained weights from the Faint Images of the Radio Sky at Twenty Centimetres (FIRST) dataset when working with NRAO VLA Sky Survey (NVSS) data. This approach resulted in a notable 5% accuracy improvement [68]. However, transferring pre-trained weights from NVSS to FIRST data led to an 8% decline in accuracy, highlighting the nuances of data compatibility in transfer learning scenarios[68].

Lin et al. [69] used a ViT to surpass CNNs in classifying galaxy morphology. ViTs have also demonstrated a 10% improved accuracy when classifying fainter, smaller galaxies with lower signal-to-noise ratios [69]. Their attention mechanism offers a solution to differentiating overlapping objects in complex scenarios where traditional methods struggle [69]. ViTs are limited by the quadratic time complexity of the attention mechanism and the need for larger datasets without an inductive bias [18]. In some instances, they may struggle to segment small objects as they operate on large patches that may miss the object of interest.

3. Aims and Objectives

3.1. Research Questions

1. How do state-of-the-art techniques for supervised classification and semantic segmentation perform on radio galaxy datasets given the scarcity of labelled data and the contrast between images obtained from radio astronomy and natural images found in conventional datasets?

2. Can self-supervised pre-training be used to improve state-of-the-art deep learning backbones for the specific task of radio galaxy detection and segmentation, thereby reducing the need for labelled data?

3.2. Aims

The aims of this project are as follows:

1. Develop models to classify, detect, and segment radio galaxies instances within the RadioGalaxyNET dataset by exploring and adapting state-of-the-art computer-vision methods and foundation models.

2. Explore the application of self-supervision to the task of segmenting radio galaxies by exploring and adapting existing self-supervision methods.

3. Promote machine learning education by documenting progress on an open access repository.

and lastly,

4. Outperform existing solutions for the detection and segmentation of radio galaxies within the RadioGalaxyNET dataset.

3.3. Objectives

These aims will be achieved through the completion of the following objectives or milestones:

1. To understand the physical and technical characteristics of RadioGalaxyNET dataset by reviewing the technical documentation. These characteristics include image size, format, resolution, channels, etc. and can significantly influence the processing techniques used.

2. To compare architectures for the detection and segmentation of radio galaxies using standardized metrics that include accuracy, precision, recall, and intersection over union (IoU), as explained in *4. Methods*.

3. To develop and evaluate a simple baseline approach for the supervised detection of radio galaxies using accuracy, precision, and recall, and explore the reasons behind the poor performance of the existing benchmarks.

4. To develop and evaluate a model for the supervised segmentation of radio galaxies using accuracy, precision, recall, and intersection over union (IoU), and investigate the difficulties associated with the creation of such a model.

5. To develop and evaluate an approach for the weakly-supervised semantic segmentation of radio galaxies using accuracy, precision, recall, and intersection over union (IoU).

6. To optimise the detection and segmentation models to outperform the existing solutions to the problem.

7. To create an open access repository on GitHub that documents all code and progress throughout the project in PEP 8 format.

4. Methodology and Methods

4.1 Methodology

In order to achieve the aims set out in this report, a neural network was developed to detect and segment radio galaxies. The methodology of this project is hence classed as quantitative as it involves the development of models that will be analysed with quantitative metrics to determine accuracy. To carry out this process, the main steps involved will include data collection and preparation, research on models, training, validation, testing, and optimisation.

4.1.1 Software and Hardware

We use the PyTorch [70] library in Python to train most networks due to its ubiquity in machine learning research and extensive existing implementations of various methods. YOLOv9 is only available through the proprietary Ultralytics library. Our implementation of SAM and DINO are obtained from the Hugging Face and FastAI libraries respectively. We make use of distributed training on the MASSIVE M3 High Performance Computer (HPC) [71] to train large networks such as SAM. This allows us to use multiple GPUs concurrently such as the NVIDIA Tesla V100, T4, and A100. All code is stored on GitHub for easy accessibility to future researchers [71].

Figure 5 - M3 Massive Supercomputer Logo.

Fast.ai [72] is a deep learning library that simplifies neural network development on PyTorch, while its extension, the Self Supervised library by Kerem Turgutlu [73], adds self-supervised learning methods like BYOL [57], SimCLR [56], DINO [58] and SwAV [74] for single GPU operation. These methods are essential for leveraging unlabeled data, particularly after the deprecation of the torch.distributed.launch module in PyTorch 1.10 [75], which affected the source builds for DINO. The Self Supervised package thus provides the necessary infrastructure to adapt to these changes and continue using advanced self-supervised techniques effectively.

4.1.2 Experimental Framework

The specific procedure for training any neural network consists of iteratively updating the parameters of the network across small batches of the dataset. This requires computing a loss between the predictions of the network and the true labels [76]. Each parameter is updated using the gradient of the loss with respect to the parameter. The magnitude of parameter updates is controlled by the ‘learning rate’ and the number of times the network iterates across the whole dataset is termed ‘total epochs’. We report more details for each method below. We train our networks on a smaller subset of the dataset called the ‘train set’ and evaluate it on a separate ‘validation set’ every epoch. The network parameters are saved at the epoch of

the lowest validation loss to avoid memorization of the train set. We seek to optimize performance on the validation set by adjusting hyperparameters such as learning rate, epochs, etc. Finally, we report performance on a held out 'test set' to combat any bias introduced by the optimization of validation metrics. We employ random rotations and flips as a data augmentation strategy to supplement our small dataset and prevent the model from memorizing samples. These transformations avoid damaging the content of the image and encode rotational invariance as a useful inductive bias. Although there are no hard rules regarding the choice of hyperparameters, we use heuristics alongside trial and error. YOLOv9 is additionally optimized using the algorithms provided by Ultralytics. No optimizer is used for other methods such as SAM due to the large amount of training time and resources this requires.

4.1.3 Metrics

We evaluate detection and segmentation networks using Intersection over Union (IoU) and Average Precision (AP). IoU as shown in *Equation (1)*, is computed by dividing the area of the overlap between predictions and labels by the area of the union of predictions and labels. An AP is obtained by classifying predictions as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) with respect to the labels after thresholding prediction IoU. These are used to compute Precision and Recall across the set of predictions above a confidence threshold. Precision favors a network that underpredicts the number of objects while Recall favors a network that overpredicts the number of objects. AP is a holistic measure of both Precision (*Equation (2))* and Recall (*Equation (3)*) obtained by varying the confidence threshold and computing the area of the resulting Precision-Recall curve. To combat class imbalance, a mean IoU (mIoU) and mean AP (mAP) are obtained by separately computing IoU and AP for each object category and averaging the resulting values. Both metrics are bounded between 0 and 1 and larger values correspond to better predictions. Sometimes, F1 score *(Equation (4))* is also utilized to ensure a more accurate reflection of accuracy over uneven class distribution, it is the weighted average of precision and recall for each class.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

4.1.4 Data

Our dataset is 'RadioGalaxyNET' curated by Gupta et al. [7]. It contains 2800 samples of 3-channel 450x450 images with 4155 total instances of 2800 unique radio galaxies. These images have been taken by the CSIRO's Australian Square Kilometre Array Pathfinder (ASKAP) telescope, located at the Inyarrimanha Ilgari Bundara, the CSIRO Murchison Radio-astronomy Observatory on Wajarri Yamaji Country in Western Australia.

Two channels correspond to radio-frequency measurements, while the third corresponds to infrared. The radio galaxies are categorized as 13% FR-I, 48% FR-II, 14% FR-X (hybrid), and 25% R (unresolved or unclassifiable). The dataset is divided into train, validation, and test splits in a ratio of 0.7:0.15:0.15 respectively. Each set preserves the distribution of categories across the whole dataset. Some alternative

datasets are MiraBest [77] and CRUMB [78] which contain approximately 2000 images and only galaxy category labels. Galaxy Zoo [79] is a larger but deprecated dataset containing 6536 images and bounding boxes alongside categories. We elect to use RadioGalaxyNET for the following reasons: (1) the presence of ground truth segmentation masks and bounding boxes allows supervised detection and segmentation for the first time, (2) the 3-channel structure of images is directly compatible with popular computer vision methods, (3) the simple processing of the dataset allows generalizability of results across sky surveys.

4.2 Methods

4.2.1 YOLOv9

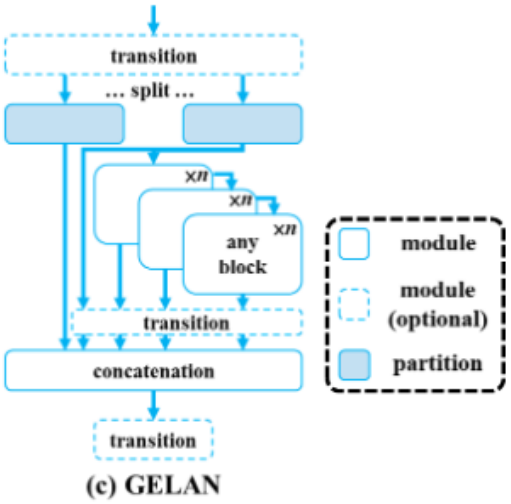


Figure 6 - Visual Representation of Yolov9 Architecture [23].

As shown in the literature review, YOLOv9 is an incredibly powerful and lightweight architecture, which would be very useful for the task of four class detection and segmentation models. Furthermore, YOLOv9's implementation of GELAN, PGI, and an emphasis on reversible functions results in a lower loss of information. As the images we intend to use are quite basic (mostly black) and the objects we wish to detect appear very small in the image, a CNN based model that is designed to preserve as much information as possible is an excellent choice, as it allows us to increase the depth of the model i.e. allows for parameters for learning.

Additionally, we explore the use of a plethora of advanced image augmentation methods such as left-right and up-down flips, rotations, scaling, shear, perspectives, mixups (the combination of two images), mosaics (the combination of multiple images), and copy-paste (copying and pasting objects from one image into another). These augmentation techniques make the dataset more robust, preventing overfitting and aiding in training. In the case of YOLOv9, these augmentation techniques are treated as hyperparameters, the extent of which each occurs controlled by a variable.

4.2.1.1 YOLOv9 Detection

We choose to implement the YOLOv9 detection model - both the unofficial version from Wang et al. and the version from Ultralytics. For both models, we select the c version as, while the e version has a slightly higher overall accuracy on MS COCO, it is much larger than the c version. Thus, we trade off accuracy for efficiency.

In order to implement the yolov9-c detection model from Wang et al. it was necessary to implement the architecture and scripts available from their GitHub repository [80]. As these scripts weren't broadly transferable and incredibly new, it took a significant amount of debugging to get the training files working on the RadioGalaxyNET dataset.

The model was trained and the metrics of the model were analysed. These included precision/recall curves, confusion matrices, visualised results, and overall accuracy. The analysis of the training process would then inform the changes made to the hyperparameters, which were tuned by hand. Six different models were generated from scratch with this procedure. Once the changing of hyperparameters was no longer improving the model, the Ultralytics YOLOv9 model was used.

The setup of the Ultralytics YOLOv9 was far easier. The Ultralytics packages included training, validation, testing, and fine-tuning functions compatible with the new yolov9c. The prebuilt commands were then used to fine-tune a pre-trained detection model, before opting to train a model from scratch. The metrics of the from-scratch model were then analysed in the same fashion, before fine tuning. The fitness curve of the fine-tuning procedure was analysed and the best hyperparameters were determined. These hyperparameters were then used to further train the model, and the final result was fine-tuned again. This process was repeated twice and stopped once the accuracy had plateaued between models.

4.2.1.1 YOLOv9 Segmentation

We also chose to implement the YOLOv9 segmentation model, however, only the architecture from Ultralytics is available to the public. Once again, we select the c version.

The YOLOv9 segmentation model was implemented similarly to the Ultralytics detection model. The prebuilt Ultralytics functions were used to train a segmentation model from scratch. This model was then analysed and fine-tuned, with the best model generated by the fine tuning model being fine tuned further. This process was repeated three times before it was determined that the model was no longer improving.

4.2.2 U-Net

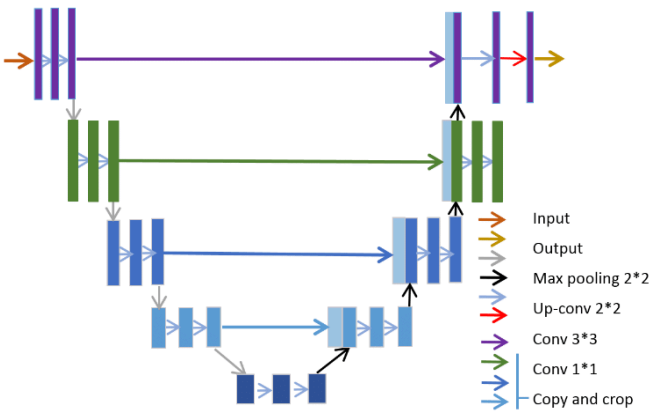


Figure 7 - Visual Representation of U-NeT Architecture [81].

We evaluate a U-Net for the semantic segmentation task given its ubiquity in research and lightweight, flexible architecture. Our U-Net is constructed out of ConvNeXt network blocks that mimic the skip connections of a ResNet but incorporate elements from ViTs. This is inspired by the success of the ResNet architecture for radio imagery and by the emerging success of ViTs across domains. ConvNeXt uses the

convolution operation which is well-suited to radio imagery as the segmentation of small galaxies only requires localized context. Both the decoder and the encoder are composed of five stages and the total number of parameters in the network is ~5.8M. This makes it easy to train for longer durations with limited resources. We minimize the weighted sum of Dice and CE loss given the presence of small objects in the dataset. Our hyperparameters are reported below (*Table 3*).

Table 3: Hyperparameters of U-Net Model

Hyperparameter	**Value**
Batch Size	10
Epochs	185
Learning Rate	0.00023090921666541305
Optimizer	AdamW
Weight Decay	0.015346716328972766
Dice Loss Weight	0.015346716328972766
Cross Entropy (CE) Loss Weight	0.98465328367

4.2.3 SAM

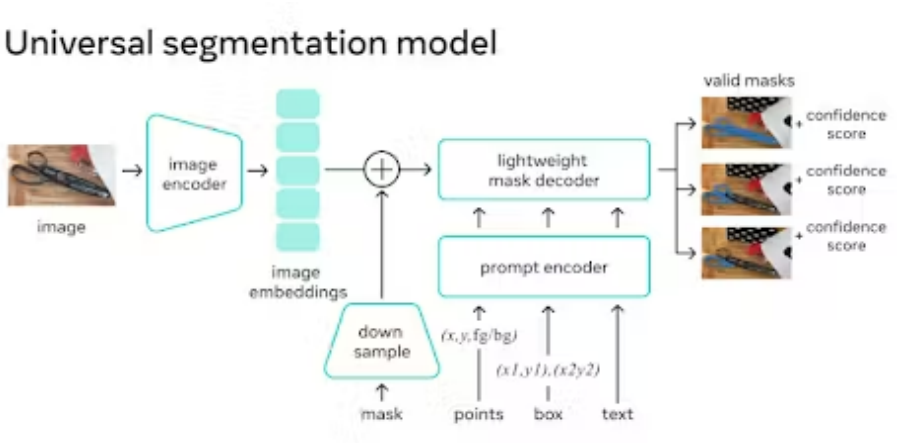


Figure 8 - Visual Representation of SAM Architecture [40].

Similar to previous works in medical image segmentation, we evaluate SAM [40] for the semantic segmentation task by: (1) fine tuning the lightweight decoder, and (2) fine tuning the Low Rank Approximation (LoRA) of the encoder weights as well as the decoder. This helps us evaluate the usefulness of LoRA. We note that SAM is a large model with 91M total parameters. The results we obtain are also indicative of the performance of large models for radio imagery which is not well studied in the literature. Moreover, ViTs have revolutionized machine learning tasks across domains and SAM is claimed to be a universally applicable model. Testing this claim helps us evaluate the applicability of pretraining for radio imagery. We minimize the weighted sum of Dice and CE loss given the presence of small objects in the

dataset. We use early stopping to stop training after the mIoU on the validation set does not improve for a certain number of epochs. We also reduce the learning rate by a scaling factor using a learning rate scheduler after a certain number of epochs without mIoU improvement. This helps us save training time. Our hyperparameters are reported below.

The LoRA of a weight matrix is obtained by computing its Singular Value Decomposition (SVD) as the product of three matrices. The middle matrix is a diagonal matrix whose entries are referred to as 'singular values' and occur in descending order of importance for the reconstruction of the matrix. The rank of a matrix is found by the number of nonzero entries. A 'low rank approximation' of the matrix is obtained by setting the last 'r' singular values to 0. This is equivalent to removing 'r' rows and columns from the remaining matrices and hence decreases the number of trainable parameters.

Similar to Zhang et al. [82], we downscale the 'patch embeddings' and 'positional embeddings' of the SAM image encoder to support 450 x 450 images. We use the 'bilinear interpolation' [83] algorithm which performs sequential linear interpolation in two directions given four closest points to the location of interpolation.

Table 4: Hyperparameters of SAM Model.

Hyperparameter	**Value**
Batch Size	2
Epochs	185
Early Stopping Patience	10
Learning Rate Scheduler Patience	5
Learning Rate Scheduler Scaling Factor	0.1
Learning Rate	0.00023090921666541305
Optimizer	AdamW
Weight Decay	0.015346716328972766
Dice Loss Weight	0.4187426704271407
Cross Entropy (CE) Loss Weight	0.58125732957

4.2.4. Self-supervised learning

To explore whether self-supervised pre-training can enhance deep learning backbones for Radio Galaxy classification, detection, and segmentation—thereby increasing prediction accuracy and reducing the need for labeled data—the state-of-the-art self-supervised method DINO (DIstillation with NO labels) is employed. Developed by Meta AI, DINO is a robust framework that can extract semantic segmentation information from images without requiring labeled data. This simplicity and versatility enable DINO to be applied across various architectures, including ResNet 50[84] and Vision Transformers (ViT) [85], where it has demonstrated significant improvements in few-shot learning on the ImageNet dataset. Researches also

showed that DINO outperforms other self-supervised segmentation techniques for vision tasks in specialised disciplines such as medical imaging [60].

4.2.4.1. DINO

At the heart of DINO is a teacher-student architecture similar to traditional knowledge distillation methods, where both models align their outputs through self-distillation. This mechanism not only fosters learning in the absence of labels but also ensures that the model can generalize across a wide range of visual tasks.

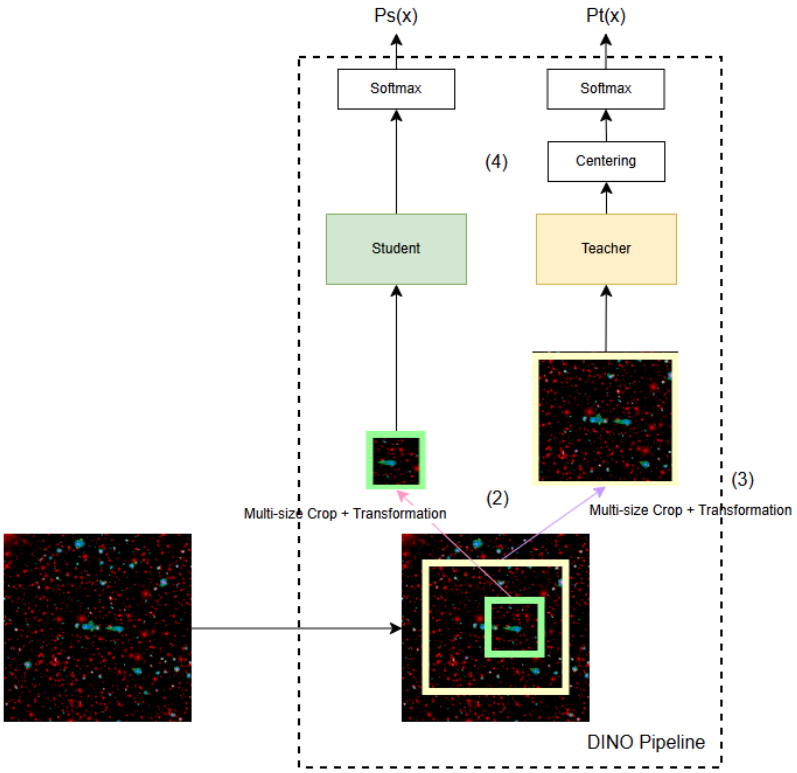


Figure 9 - Visual Representation of DINO Architecture.

1. Model Setup:

Initialize two models with identical architectures. The teacher model is updated using an exponential moving average of the student model's parameters to ensure stability and continuity in learning. This is depicted in the formula below:

$$\theta_t \leftarrow \lambda\theta_t + (1 - \lambda)\theta_s \quad (5)$$

2. Multisized crop:

In the preprocessing pipeline of DINO, it will crop the image into various crops. Crops smaller than 50% of the image size are termed "local crops," while those larger than 50% are "global crops." The teacher

network processes only global crops, while the student network receives both local and global crops. A visualization of this step is demonstrated in (2) in the *Figure 9* above.

3. Random Augmentation:

Apply a predefined set of augmentations to enhance the model's generalizability and robustness. For this project, the following transformations are used:

- Colour Jitter: Saturation = 0.5, Probability = 0.1
- Blur: Probability = 0.1
- Random Flip: Probability = 0.5
- Random Rotation: Up to 360 degrees, uniformly distributed
- Centre Crop: 40% of the image size

These transformations are the same transformations used by the creators of the first Radio Galaxy foundational model using BYOL[64] .

4. Regularization Process

Sharpening (Student Model): Pass the student model's outputs through a temperature-weighted softmax function to regularize and sharpen the probability distribution. For this project a constant temperature parameter $\tau = 0.04$ is used, the consideration behind selecting this less aggressive value is due to the number of images available for the pre-training process is limited, having a smaller temperature parameter can avoid model collapse.

Centering + Sharpening (Teacher Model): Before applying the sharpening, the teacher model's output distribution uses a centering step to prevent the probabilities from being too flat or peaked, this is updated using the *Equation (5)*:

$$c \leftarrow c_{old} + (1 - m)\frac{1}{B}\sum_{i=1}^{B} g_t(x_i) \qquad (6)$$

Where, c_{new} and c_{old} are correspondingly the new and old centering parameters, m is the momentum, B is the batch size, and $g_t(x_i)$ is the teacher's output for the i-th image in the Batch. A visualization of this step is demonstrated in step (4) in the *Figure 9* above.

5. Model Update

Ultimately after obtaining the cross-entropy loss formulated across the final output of the student and teach model, a conventional backpropagation will take place for the the student weights using the loss obtained, while the teacher model will be updated using an exponential moving average on the student weight as shown in the equation in step 1.

4.2.4.2. Self supervised fine tuning and DARGN

In the literature review, it is noted that training a DINO ViT from scratch necessitates an extensive array of images and resources, which are not available for this project [58]. The original DINO is trained on large datasets such as ImageNet [86] which contains around 14 million images, creating a dataset of similar magnitude for Radio galaxy images is not likely for this particular project. Consequently, a significant focus of this project's self-supervised section is to investigate whether it is feasible to utilize the available smaller datasets, namely CRUMB and RGN, to pre-train a model through DINO.

While DINO was originally designed for Vision Transformers (ViT), the training of ViTs is notably resource-intensive, and their accuracy significantly depends on the dataset's size [87]. In contrast, ResNet-50 has demonstrated robust generalizability and effective feature extraction capabilities on similar datasets, as evidenced by experiments conducted by the Data61 team [7]. Consequently, ResNet-50 has been selected as the default backbone for pre-training with DINO in this project.

However, the training set of RGN remains insufficiently large. Utilizing only 50% of the training images (approximately 980 images) from RGN as the training set and employing the remaining 50% as a validation set for DINO will not be large enough for the model to learn useful representations, as demonstrated in the *Figure 10* below. Despite the model overfitting on the training dataset, the model fails to generalize on the validation data.

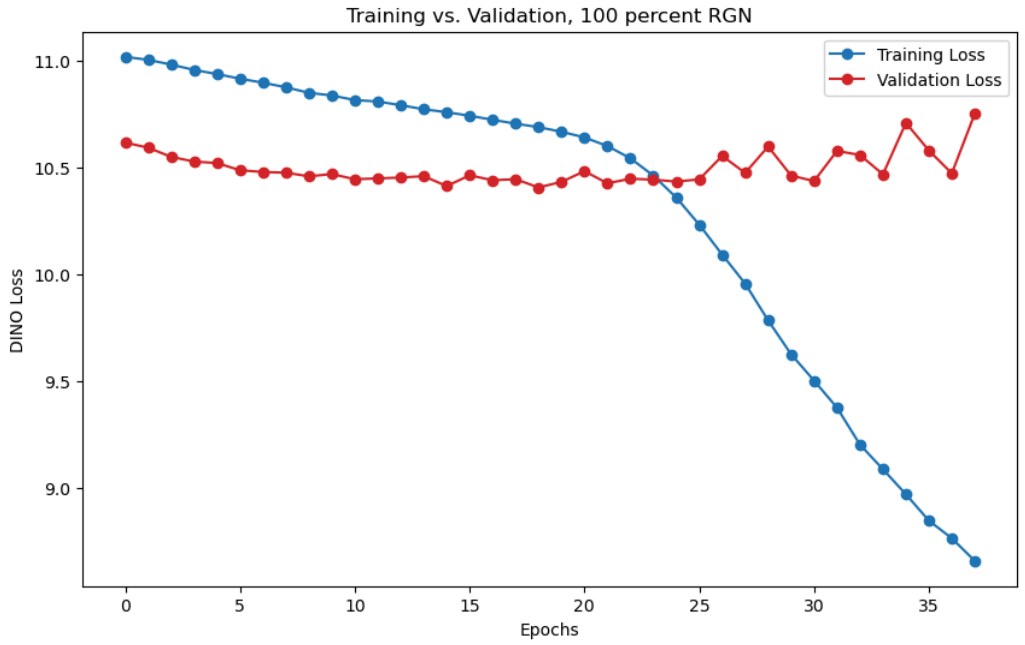


Figure 10 - Validation, Training Loss, DINO Training on 100% RadioGalaxyNET.

To resolve this issue, two methods are proposed:

1. *Method 1:* incorporate pre-trained ImageNet weights for the student and teacher models used in DINO, doing so will utilize DINO mechanism as a form transfer learning, as opposed to training model from the ground up, this method hopes to reduce the training data cost by utilizes

pre-existing features learnt through general images at the same time attune the pre-trained model to understand the general features of radio images.

The approach of integrating transfer learning into the pretraining phase contrasts sharply with the conventional methodology of fine-tuning a pretrained model on labelled data[86]. The rationale behind incorporating this additional step of transfer learning in the pretraining process is to foster the development of more generalizable backbones. These backbones are intended to be applicable to a broad spectrum of radio galaxy datasets, rather than being narrowly tailored to a specific dataset.

2. *Method 2:* DARGN (Dual-channel Augmented Radio Galaxy Net) is a novel augmentation strategy for single channel Radio Galaxy images developed to assist with the self-supervised pre-training of models utilizing two channel Radio Galaxy images similar to the ones used in RadioGalaxyNET dataset. In our task, this method transforms the CRUMB dataset [78] into enhanced instances more similar to the celestial objects found on RadioGalaxyNET. DARGN draws inspiration from established augmentation strategies, aims to improve the model's adaptability and learning from varied data representations.

    a. Random Erasing [88]: This technique involves the stochastic deletion of regions within the input image, a process designed to enhance the model's capability to discern and interpret all salient features of the targeted object.

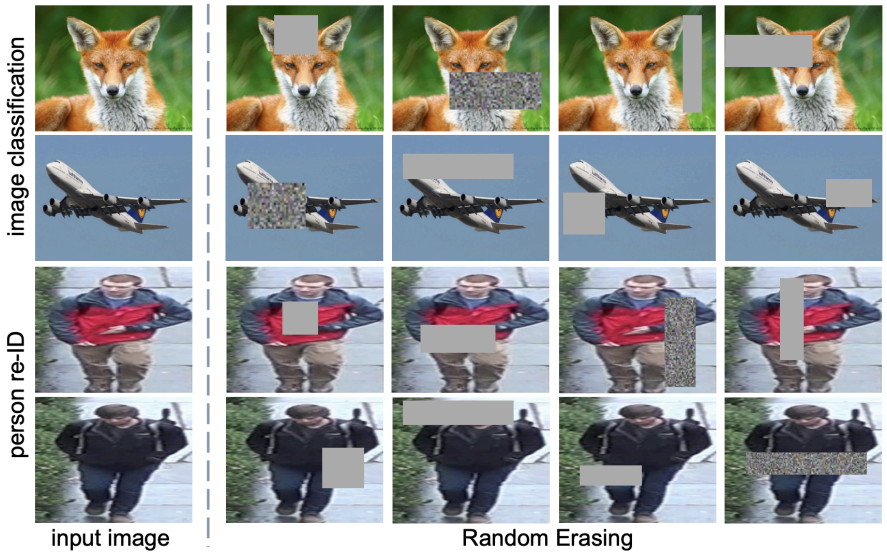


Figure 11 - Visual Representation of Random Erasing [88]..

    b. MixGen [89]: This approach entails combining multiple input images and labels into a single instance and corresponding fabricating a broader spectrum of scenarios, thereby enriching the diversity and robustness of the training data.

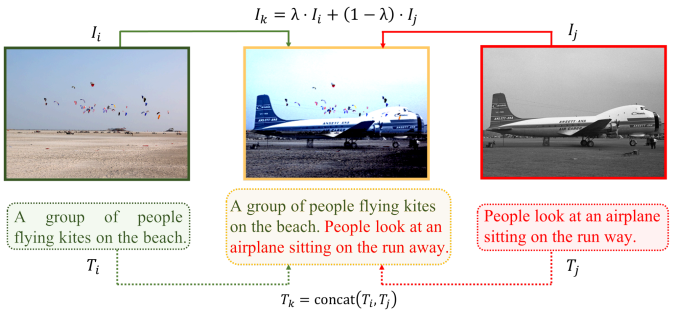
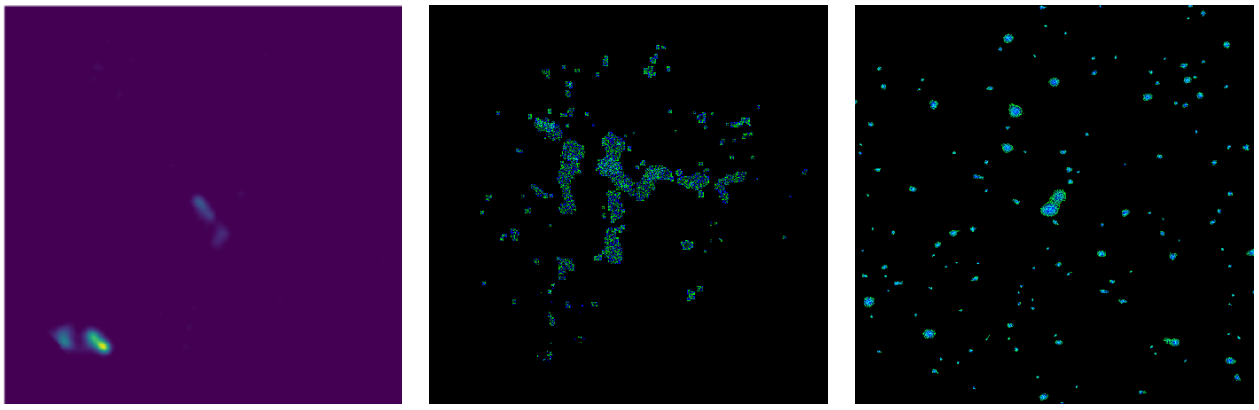


Figure 12 - Visual Representation of MixGen [89].

Ultimately we are able to create 15000 augmented training data using the 1800 instances from the CRUMB dataset, the detailed steps of augmentation is as follows:

1. Preprocess Images: Crop, rotate, and normalize each galaxy image from the CRUMB dataset, converting them into tensors. Note that the original images are single-channel grayscale with dimensions of 150x150 pixels.
2. Create Large Canvas: Generate a 450 x 450 pixel blank RGB canvas to host the augmented galaxy images from the CRUMB dataset, not MiraBest as previously mentioned.
3. Random Placement: Place between 1 and 7 randomly transformed instances of CRUMB galaxy images onto the large canvas, ensuring varied positions and scales.
4. Separate Channels: Distribute the non-zero pixel values from the single-channel CRUMB images randomly between the green and blue channels of the large canvas for enhanced visual diversity.
5. Save the Augmented Image: Convert the processed image into a standard format and save it, preserving the augmented variations for further machine learning model training.



CRUMB Images | MiraBest Images after DARGN (2c) | Sample Images from RGN (2c)

Figure 13 - Visual Comparison of RadioGalaxyNET, CRUMB, augmented CRUMB using DARGN.

Note: Since there is no established method to simulate the Inferred Channel of the RadioGalaxyNET (which corresponds to the red channel in images) and the main objective of this project is not key point detection. The red channel will be excluded from all pre training and testing in the self-supervised research section of this report.

After obtaining the augmented dataset, we will create the training and validation datasets for DINO. The training dataset will comprise 50% of the data from RadioGalaxyNET and the 15,000 images generated using DARGN. In contrast, the validation dataset will include the remaining 50% of the training images from RadioGalaxyNET. A data sampler will be employed to ensure that each batch contains a mix of data, with a proportion 'p' generated using DARGN and '1-p' sourced from RadioGalaxyNET. This mixed data will then be fed into DINO to undergo the self-supervised learning pipeline described earlier. The weight with lowest validation loss will then be used as the optimal backbone which is to be tested in the Faster RCNN.

4.2.4.3. Testing on Faster RCNN

We evaluate the performance of the pretrained ResNet50 backbone by integrating it within the Faster R-CNN framework [90], a robust object detection model that combines a Region Proposal Network (RPN) with Fast R-CNN for precise object localization and classification. Since the focus of this project is on the backbone pre-trained using DINO, we will not go in detail about the theory and application behind the Faster-RCNN architecture. For all the following experiments involving “DINO + DARGN” backbone and Faster RCNN, we will be using the backbone formulated through the “60% DARGN, 40% RadioGalaxyNet” example.

To determine the efficacy of the backbone developed using the DINO self-supervised learning approach, we first compare it against the traditional ImageNet[86] backbone trained through supervised method. This comparison is conducted by freezing the backbone layers and training only the predictor of the Faster RCNN, which includes a series of fully connected layers. The performance metric used in this evaluation is the mean Average Precision at 50% intersection over union (mAP50) for detection.

Subsequently, we explore the impact of a fully trainable, enhanced backbone on overall model performance on the testing split of the RadioGalaxyNET dataset. By incorporating a segmentation head alongside the detection module in Faster R-CNN, we are able to compute the mean Intersection Over Union (mIOU) and mAP metrics. This dual metric evaluation helps in assessing the comprehensive capability of the improved backbone in both detection and segmentation tasks.

The integration of various backbones into Faster-RCNN will use the torchvision subpackage of PyTorch, with code and tests available on GitHub. To maintain consistency in training and ensure accuracy differences aren't due to varying hyperparameters, we fine-tune a base Faster-RCNN model. The derived hyperparameters from this process are applied across all scenarios in the results.

Table 5: Optimization parameters.

Optimizer	**Learning Rate**	**Betas**	**Batch size**	**LR Scheduler Step Size**	**LR Scheduler Gamma**
ADAM	0.0015	(0.9, 0.999)	32	3	0.9

5. Results and Discussion

5.1. Final results and discussion

5.1.1 Detection Models

5.1.1.1 YOLOv9

Six different models were trained using the YOLOv9 architecture from Wang et al[23], the hyperparameters and mAP50 results from each model can be observed in "Appendix 1B". The first model generated used the recommended hyperparameters provided in the GitHub repository and achieved an mAP50 of 0.783, an incredible first result. The second model adjusted the batch size and saw no major differences. Next, the optimiser was changed from SGD to ADAM and whilst the mAP50 of the model improved, the overall number of true positives decreased. The fourth model explored the augmentation of the data, implementing a random up/down flip as well as the random left/right flip that all the previous models had. This seemed to be one too many random augmentations to the dataset and the precision of the model decreased.

The manual fine tuning of these models were performed concurrently with the automated fine-tuning of the Ultralytics model, thus, the hyperparameters of models 5 and 6 were informed by some of the best results from that model being trained. Model 5 was the overall best performer out of the six models with an mAP50 of 0.817, an incredibly high value. The training graph of the model can be seen in *Figure 14*.

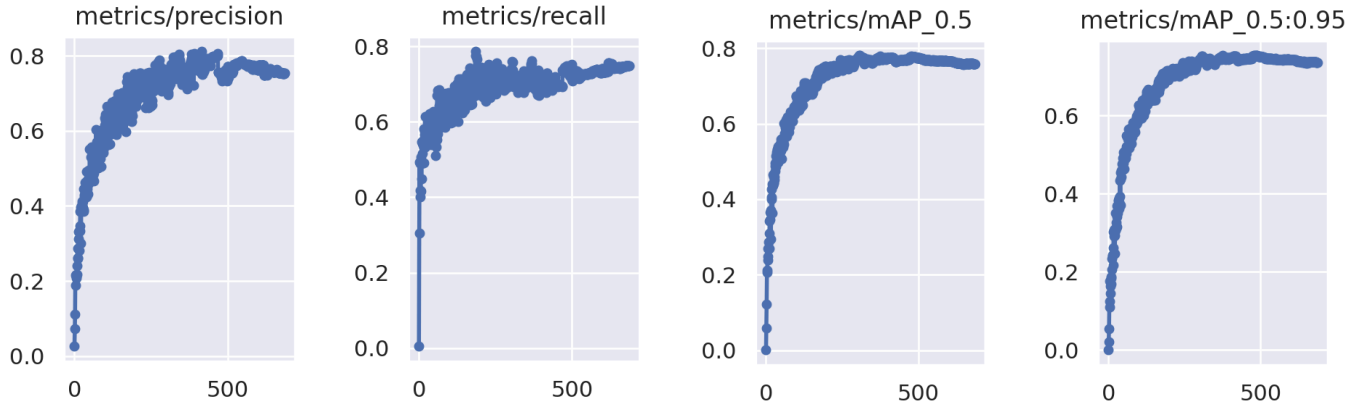


Figure 14 - The training metrics of YOLOv9 model 5.

This model was trained for over 500 epochs and, as expected, we can see the accuracy and recall increase, then plateau, then slightly decrease. The slight decrease is a result of overfitting, a phenomenon whereby the model is no longer learning the features of the training set, but instead simply memorising the images. To avoid this, we implement checkpoints that save the weights of the epoch with the best results, meaning that if overfitting occurs, it will not affect the final result.

Next we examine the confusion matrix of model 5 (*Figure 15*).

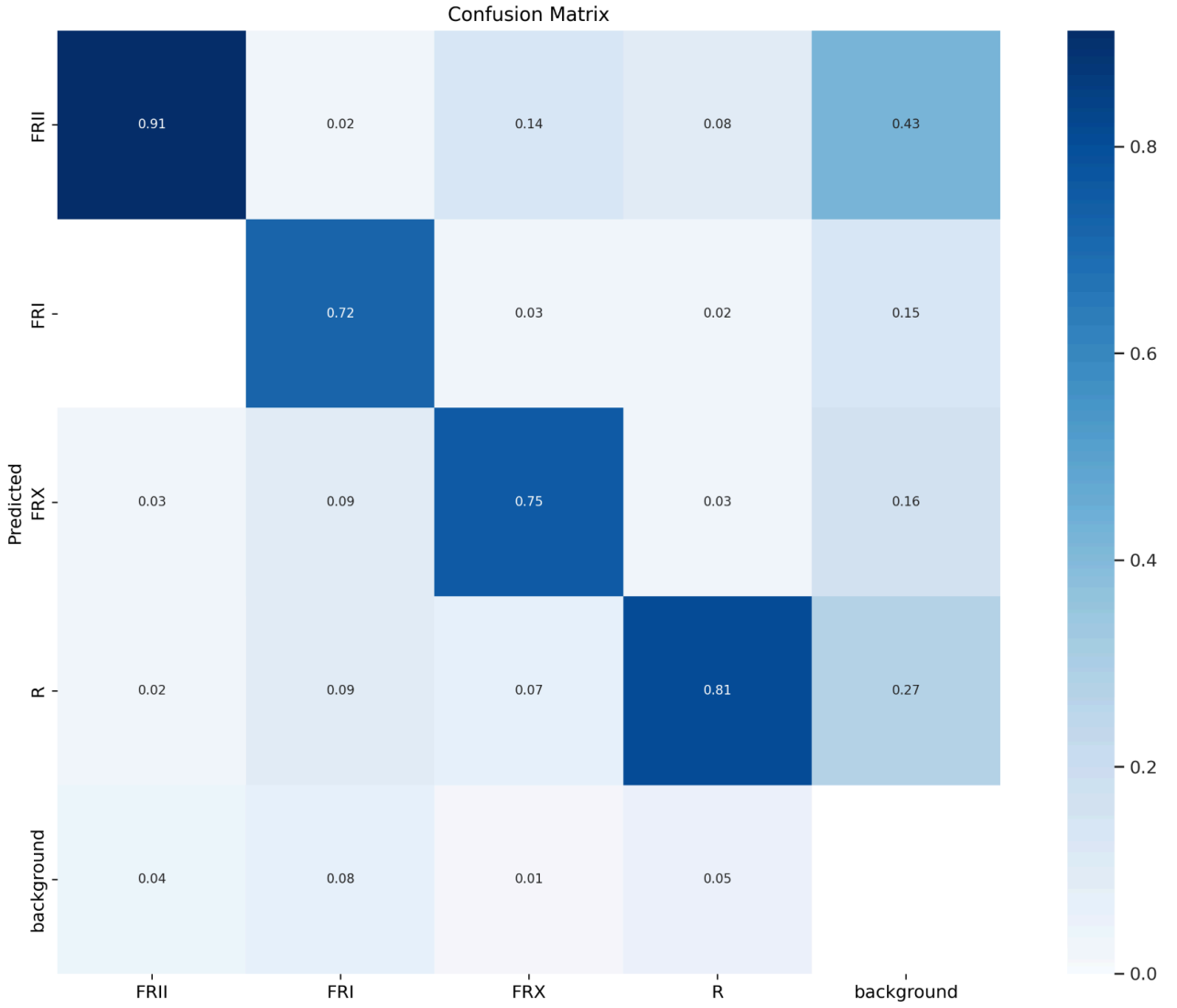


Figure 15 - The confusion matrix of model 5 - test set.

Here, it can be seen how an imbalanced dataset affects prediction rate. FR-II was the most abundant galaxy type within the RadioGalaxyNET dataset, resulting in the highest rate of true positives, 91%. The next abundant in the dataset was R, which resulted in 81% true positive detections. Finally, the two underrepresented classes FR-I and FR-X resulted in the lowest true positive rates, 72% and 75%, respectively. This idea is reinforced by the background predictions. Background galaxies were incorrectly identified as FR-II galaxies at the highest rate, followed by R, then FR-X and FR-I. In order to correct this imbalance, more radio galaxy data would be required. While RadioGalaxyNET is the largest dataset we've examined, it is still quite small in comparison to the natural image datasets available. Therefore, any imbalance in the dataset is going to be exaggerated by these models that are designed to learn on tens of thousands of images. Additionally, it is important to note that radio galaxies are rarely being incorrectly classified as other galaxies - most errors stemming from galaxies being incorrectly identified or missed entirely.

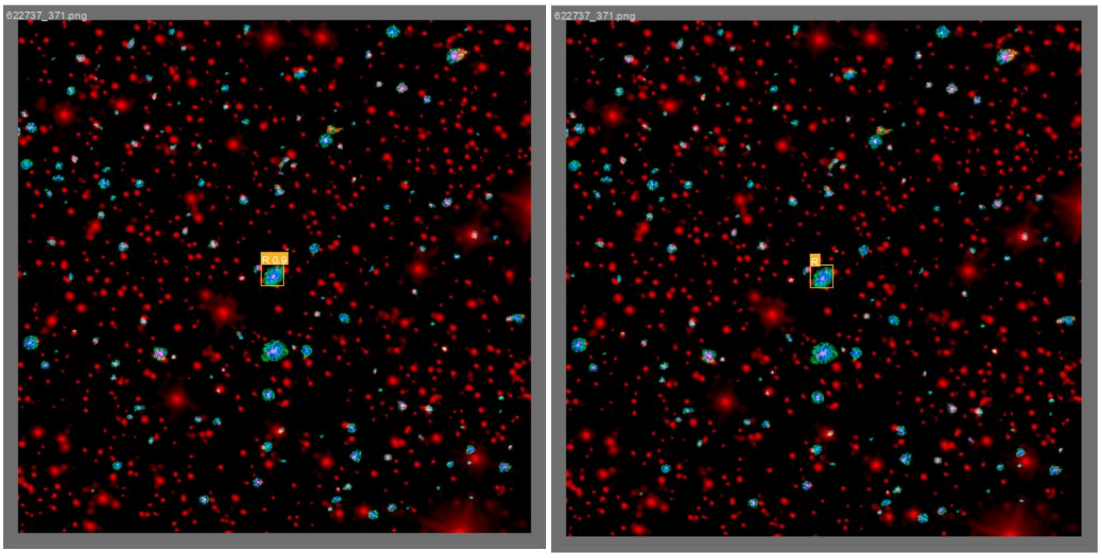


Figure 16 - A correct prediction from model 5 with a certainty of 0.9 (left) and the ground truth (right).

In *Figure 16*, we can see model 5 correctly predicts the galaxy at the centre of the image to be an R class radio galaxy with a certainty of 90%. As designed, the model places a bounding box around the galaxy.

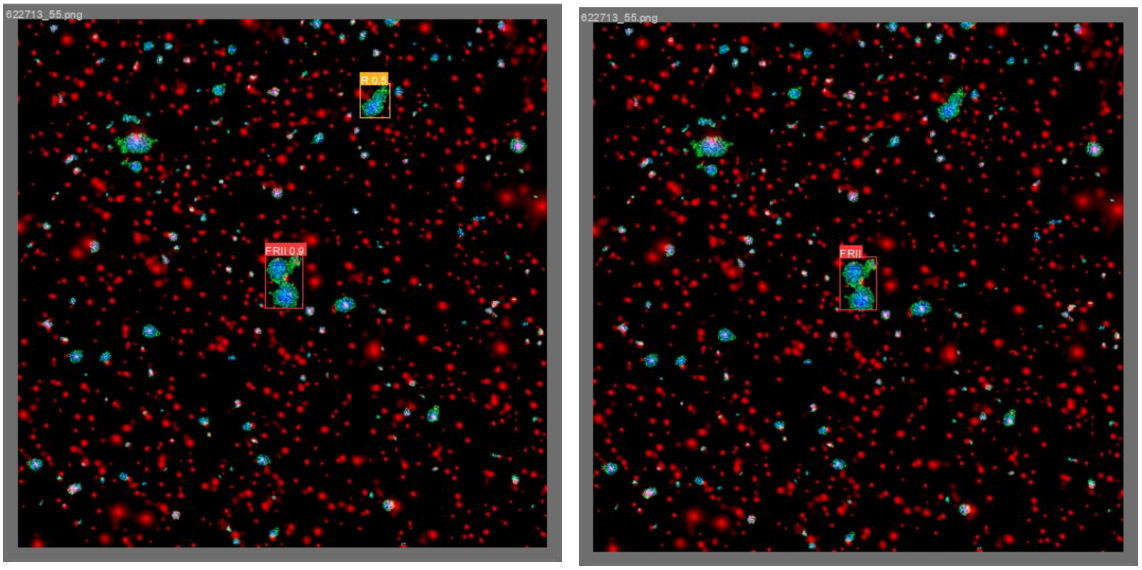


Figure 17 - A correct prediction from model 5 with a certainty of 0.9 with an additional incorrect prediction (left) and the ground truth (right).

In *Figure 17*, we see the FR-II galaxy in the centre of the image is picked up by the detection model, classified correctly with a certainty of 0.9, and highlighted with a bounding box which, to the eye, is almost perfectly aligned with the ground truth box. Unfortunately, the model has also interpreted something in the background (likely another non-radio galaxy) as an R class galaxy, however, its confidence is only 50%. This demonstrates the way in which parts of the background are being incorrectly classified and as many objects within these images look visually similar to radio galaxies, the task is made more difficult.

The final case can be seen below in *Figure 18*. This is the case where a galaxy is missed by the model. In this case, the model misses the FR-II galaxy in the centre of the image, but correctly identifies the R galaxy in the top left corner with a certainty of 90%. In this case, the galaxy was likely missed by the model because it is an FR-II galaxy that appears quite large in the image. The implications of this is that the two lobes are quite distant from the centre of the galaxy, resulting in the large amount of black space between sections, making the model think it is looking at three separate radio sources as opposed to the one.

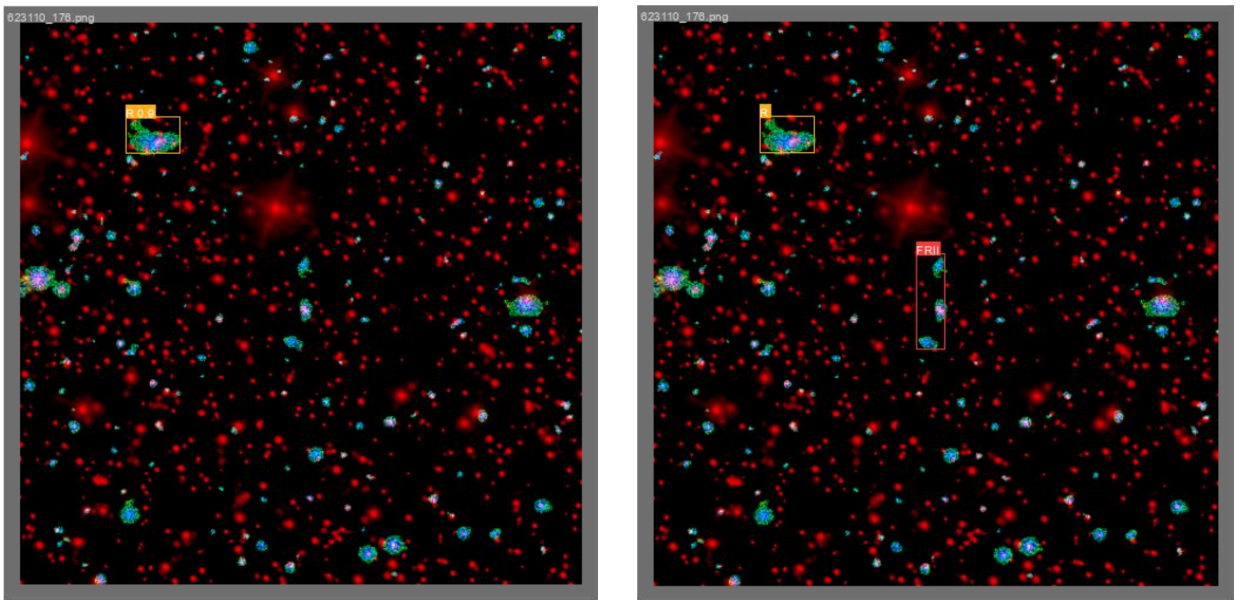


Figure 18 - A correct prediction from model 5 with a certainty of 0.9 (left) and the ground truth (right).

While these are some of the worst predictions (or lack of), they are quite rare and it is reassuring to see that when a galaxy is being predicted, the bounding box is almost perfectly in place with no visible differences between the prediction and ground-truth. Therefore, we can say that detection is being performed successfully. In order to make use of Ultralytics' fine-tuning capabilities, we make another model from scratch with the yolov9c architecture.

5.1.1.2 Ultralytics YOLOv9

First, the Ultralytics YOLOv9 model, yolov9c, was fine-tuned using pre-trained weights. The fine tuning process can be seen in *Figure 19*, where 300 different models have been trained for 30 epochs each.

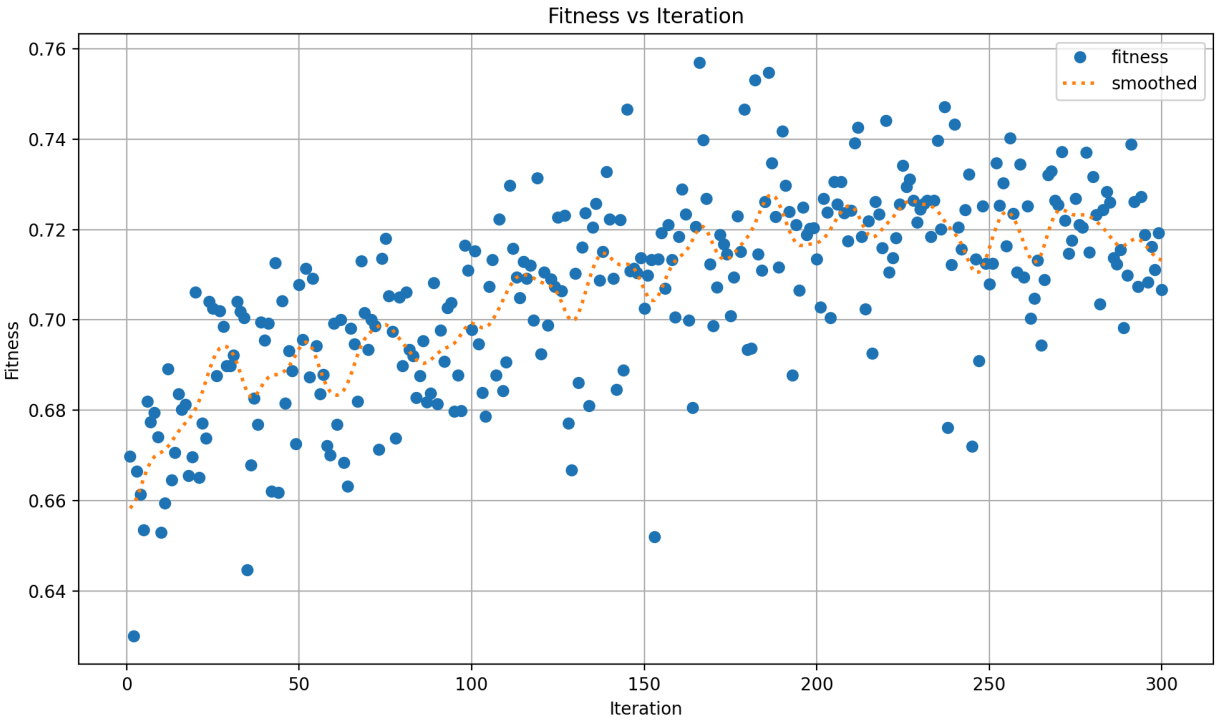


Figure 19 - The fine-tuning of a pre-trained Ultralytics yolov9c model. Fitness represents the combination of a variety of accuracy metrics.

While 30 epochs is quite small compared to the 600 epochs the previous models were training for, this fine tuning was designed as a breadth-first approach to finding the best hyperparameters. Looking at *Figure 19*, we can see that changing the hyperparameters makes a massive difference to the fitness of the model, with the results spiking up and down between each iteration. As the overall iteration number increases, we can see the smoothed fitness slowly trend upwards before plateauing, indicating that the learned hyperparameters are no longer improving overall.

Figure 20 shows how training many separate models can show trends in hyperparameter adjustments. While simply taking the hyperparameters of the highest accuracy case is a valid approach, examining the scatter plots helps us to understand the ideal value for each hyperparameter. That stated, as 30 epochs is a low number for training, the hyperparameters from the iteration with the highest fitness were used to train a model from scratch, the training results from which can be seen in *Figure 20*.

This shape of the mAP50 graph (*Figure 21*) for this model is very typical. The model learns key features very quickly, before plateauing around 0.8. This model did not seem to overfit, however, it did not improve after roughly epoch 200. This model then underwent further fine tuning in an attempt to improve its accuracy.

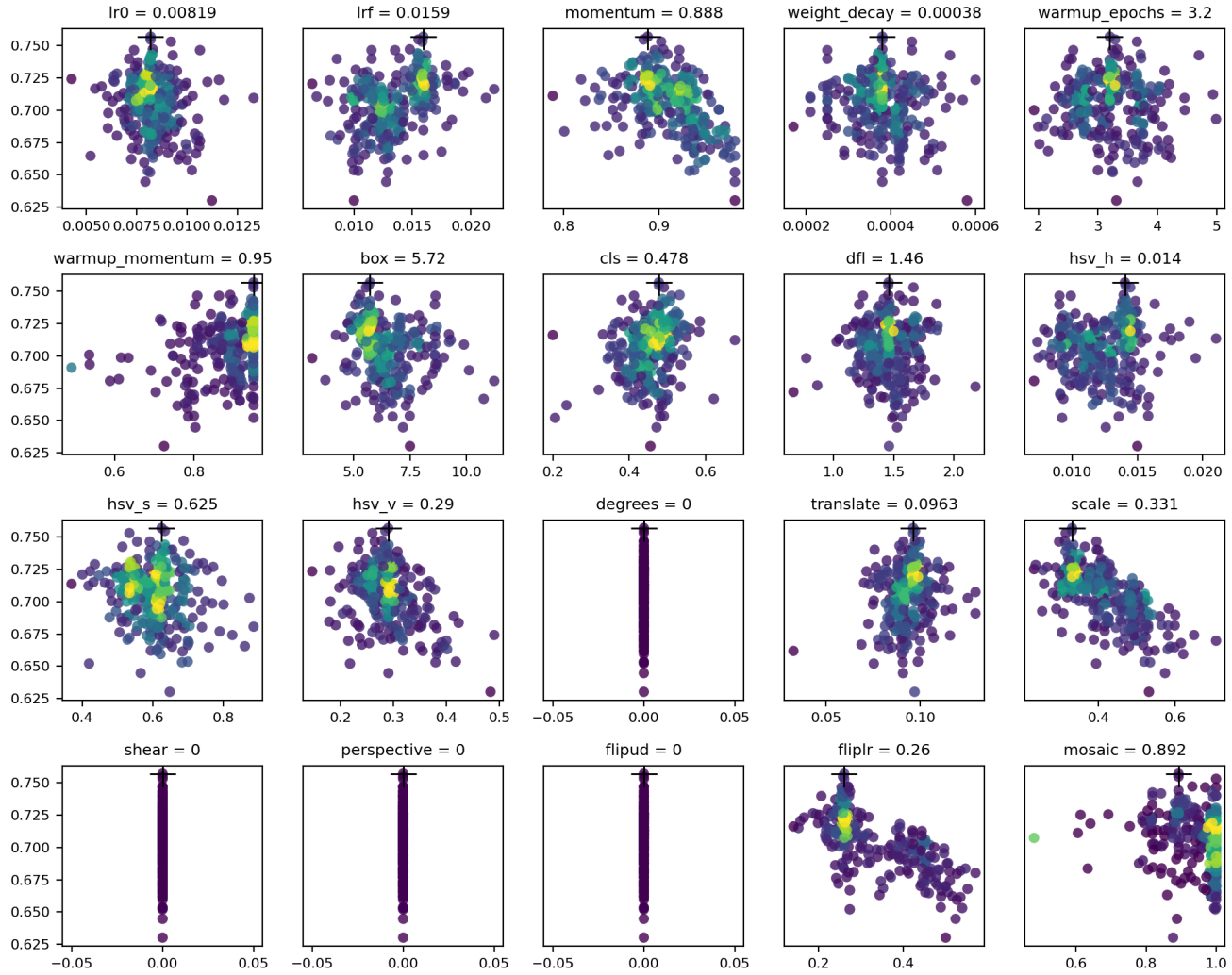


Figure 20 - Hyper-parameter scatter plot of the fine-tuning of the pretrained yolov9c model. Vertical axis represents fitness and the cross represents the best case. The data points move from purple to yellow as interactions increase.

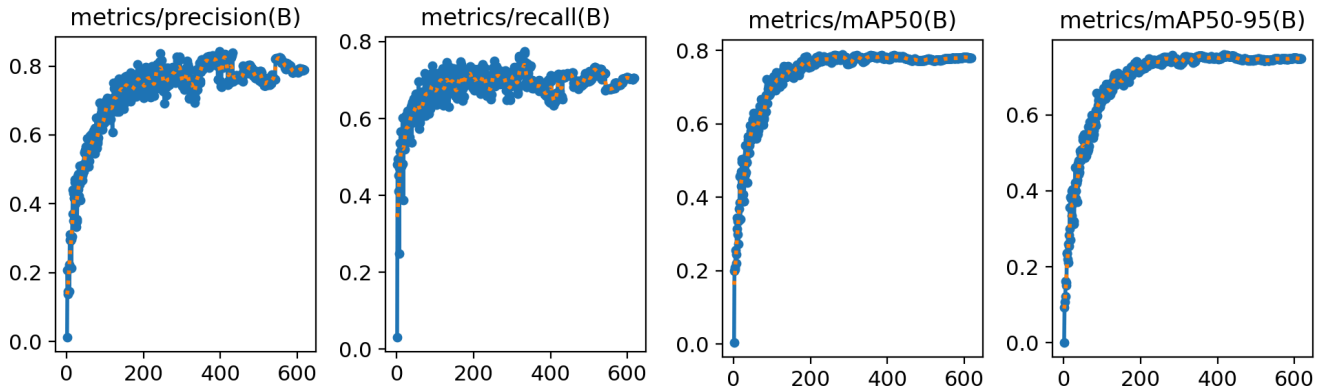


Figure 21 - The training metrics of a yolov9c model from scratch. The hyperparameters were taken from the fine-tuning of a pretrained model.

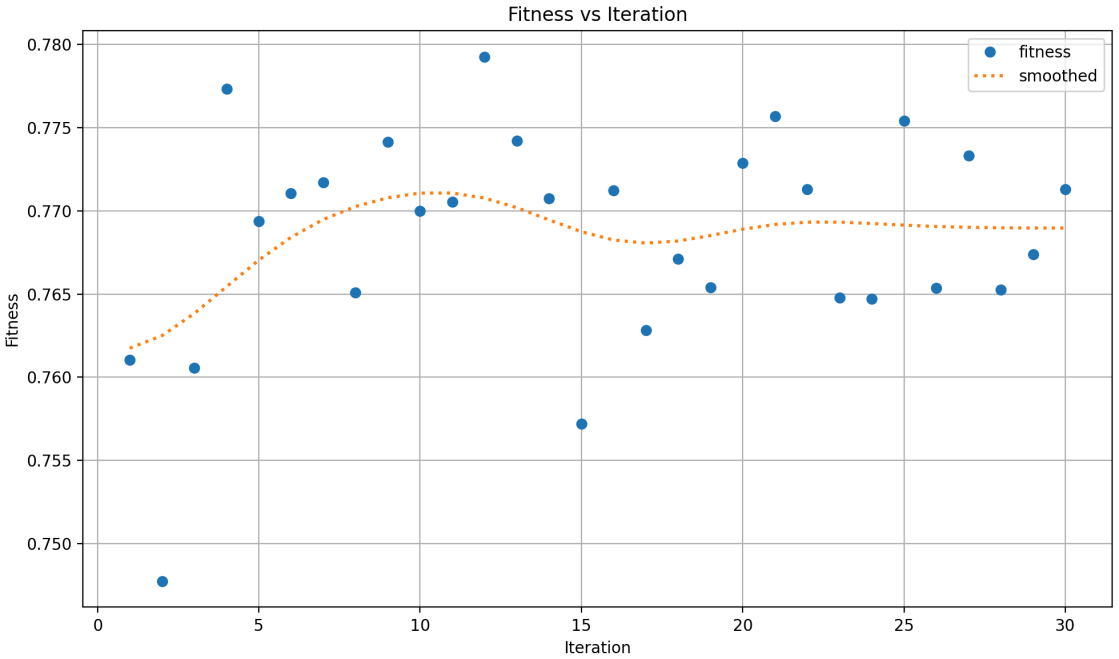


Figure 22 - First round of fine-tuning for the Ultralytics yolov9c model trained from scratch. Fitness represents the combination of a variety of accuracy metrics.

This model was fine tuned for 30 iterations of 200 epochs. Unfortunately, as can be seen in *Figure 22*, the overall fitness of the model did not increase significantly, as training the model further led to overfitting. Another model was fine tuned beyond this, however, it also led to overfitting. In the end, The highest test mAP50 obtained from this model was 0.798, slightly lower than the Wang et al. yolov9-c model.

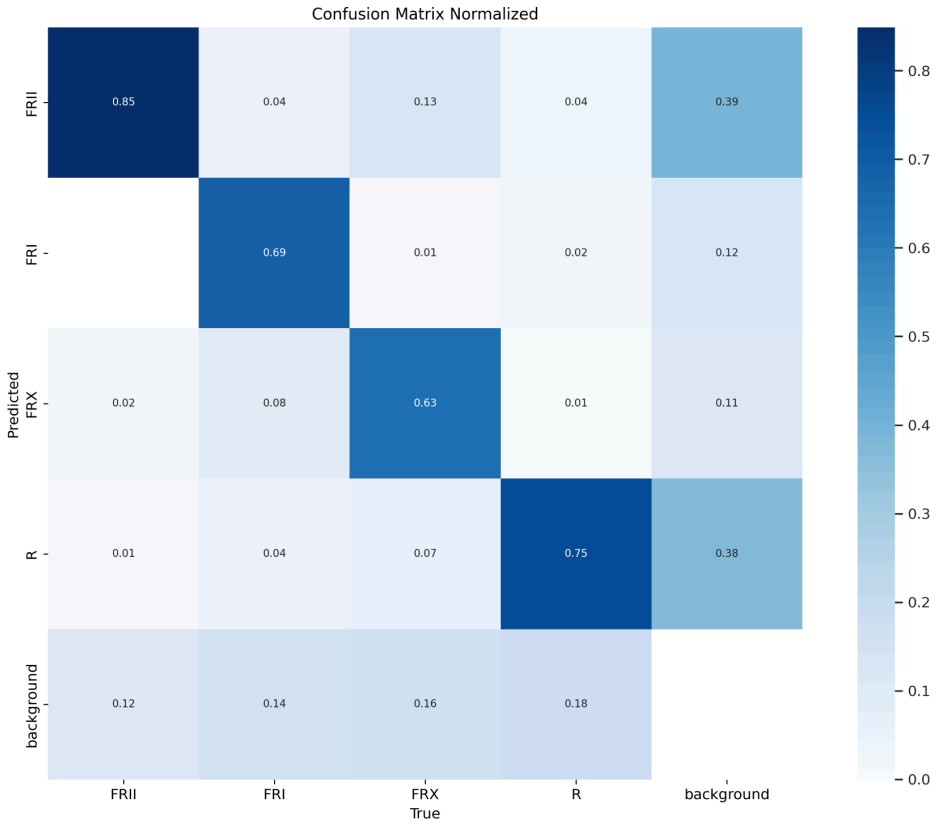


Figure 23 - The confusion matrix of the highest mAP50 Ultralytics yolov9c model - test set.

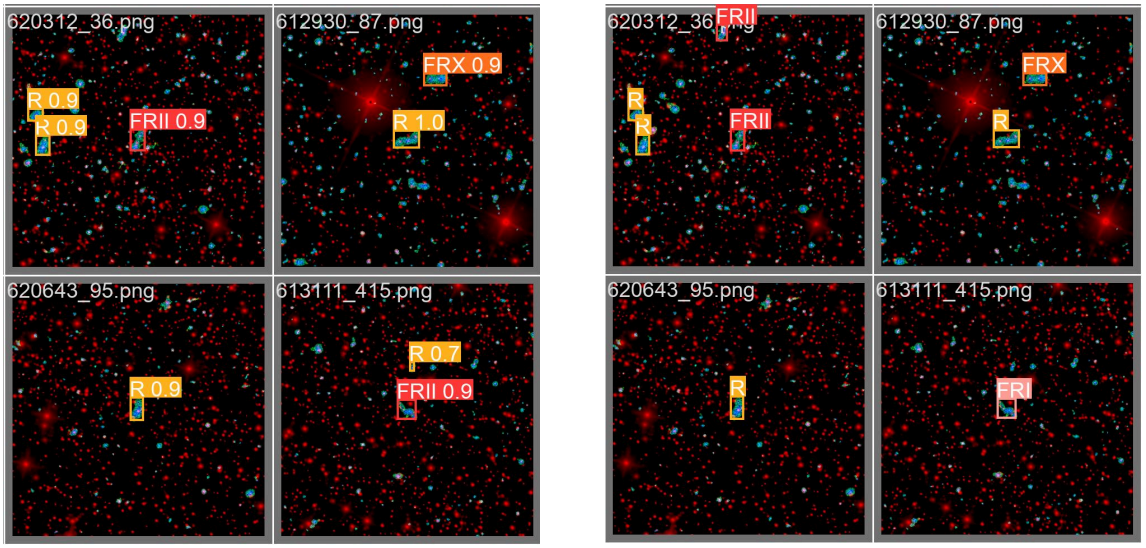


Figure 24 - Predictions from the highest mAP50 Ultralytics YOLOv9 model (left) and the ground truths (right).

Figure 23 visualises the confusion matrix of the highest accuracy case, and *Figure 24* shows some of the predictions obtained from the model. Generally speaking, we see the same trends as the Wang et al. yolov9-c model. Classes that are underrepresented in the dataset tend to be predicted less and accuracy is lowered by the model incorrectly classifying background noise as radio galaxies or, alternatively, missing radio galaxies entirely.

5.1.2 Segmentation Models

5.1.2.1 Panoptic Segmentation with YOLOv9

A model was trained from scratch using the Ultralytics yolov9c segmentation architecture. The training graphs of the model can be seen in *Figure 25*.

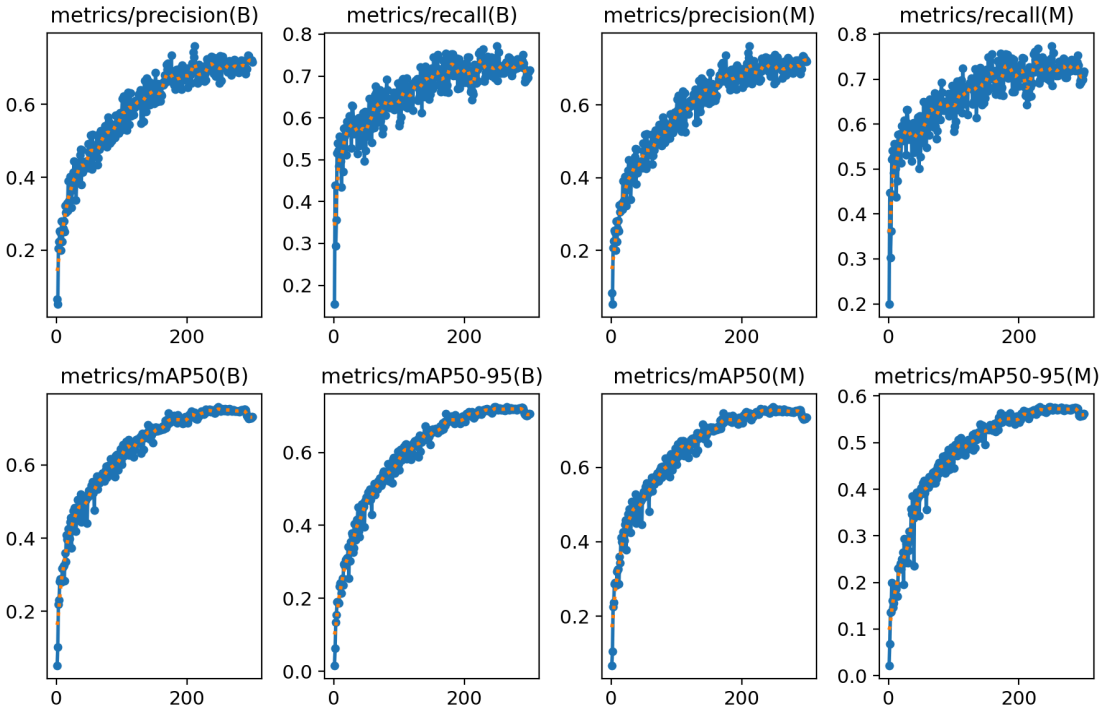


Figure 25 - The training graphs of the YOLOv9 Segmentation model. B stands for Bounding Box Metrics and M stands for Segmentation Mask Metrics.

This model was trained for 300 epochs. As the model approaches the end of the training, the metrics begin to plateau, however, more epochs are required to confirm this. This model was then fine-tuned to examine the effects of further training.

The model was fine tuned for 20 iterations of 300 epochs each. Striking a balance between the number of iterations required versus the number of epochs each iteration is trained for is a difficult process. As we can see, the smoothed fitness decreases as iterations increase, however, 20 iterations is quite low, so it is unclear whether the fine-tuning process would improve or not.

In *Figure 27*, we can see the training graph of the highest fitness model. We observe a massive drop in performance after the first iteration. This occurs when the weights try to update on a pretrained model all at once and the model undergoes a process called “catastrophic forgetting”[91] losing all retained information at once. While, it would be preferable to freeze then gradually unfreeze the layers during the fine-tuning process so that more information was retained, this proved very difficult to implement using the Ultralytics functions. Therefore, each iteration had to be trained for long enough so that it could relearn what was forgotten, thus, reducing the number of iterations possible.

The model was then fine-tuned a second time for 30 iterations at 200 epochs (*Figure 28*).

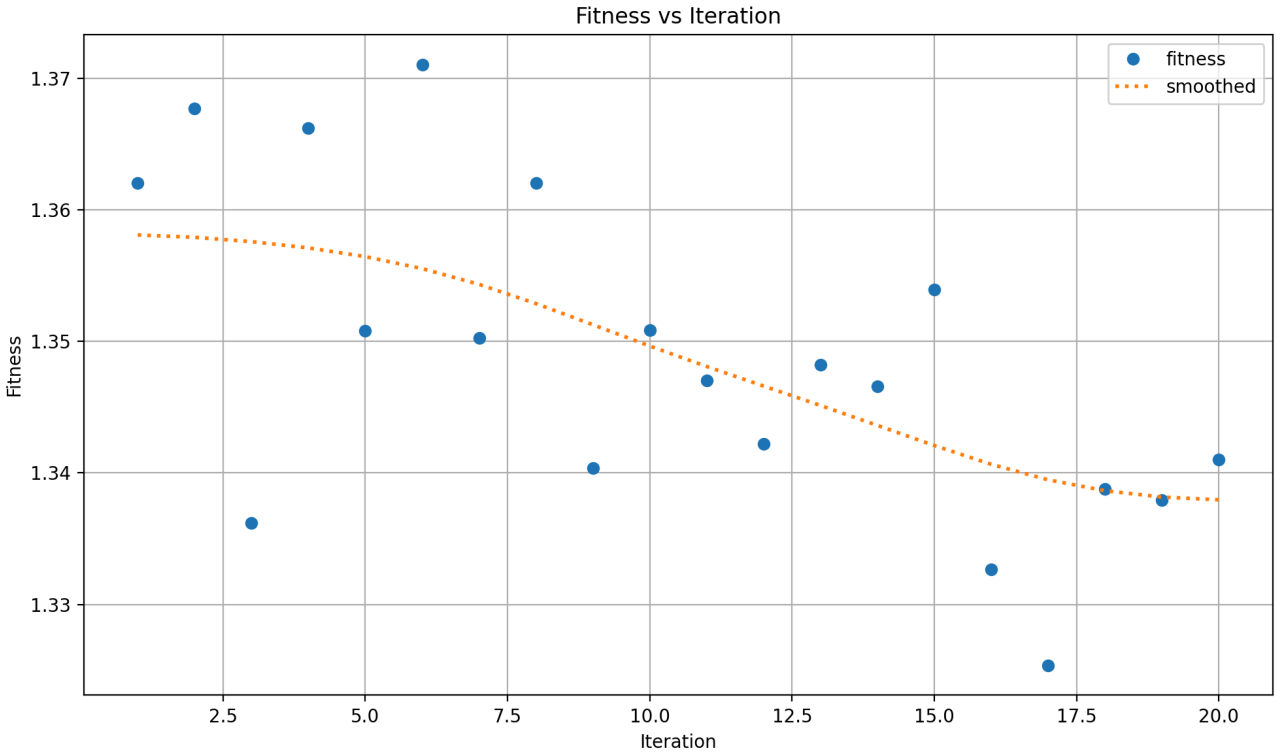


Figure 26 - First round of fine-tuning for the Ultralytics yolov9c segmentation model trained from scratch. Fitness represents the combination of a variety of accuracy metrics.

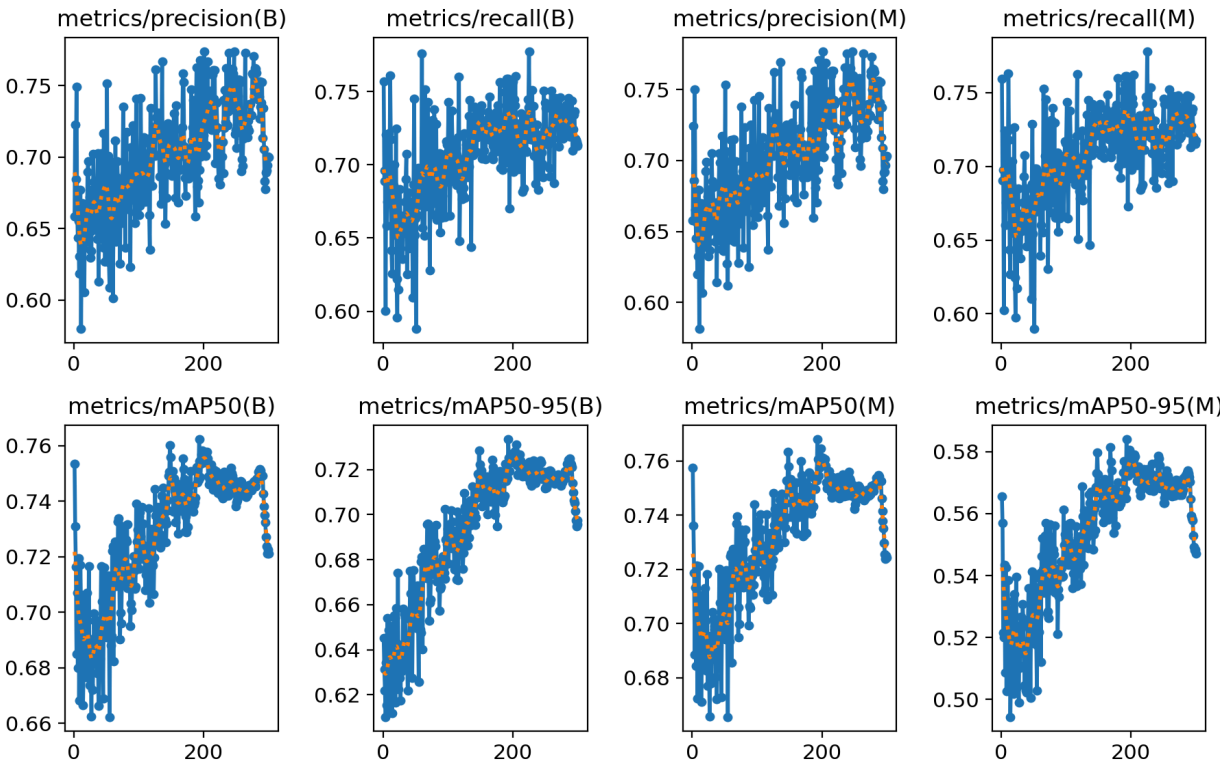


Figure 27 - The best training of the YOLOv9 Segmentation model during the first round of fine-tuning. B stands for Bounding Box Metrics and M stands for Segmentation Mask Metrics.

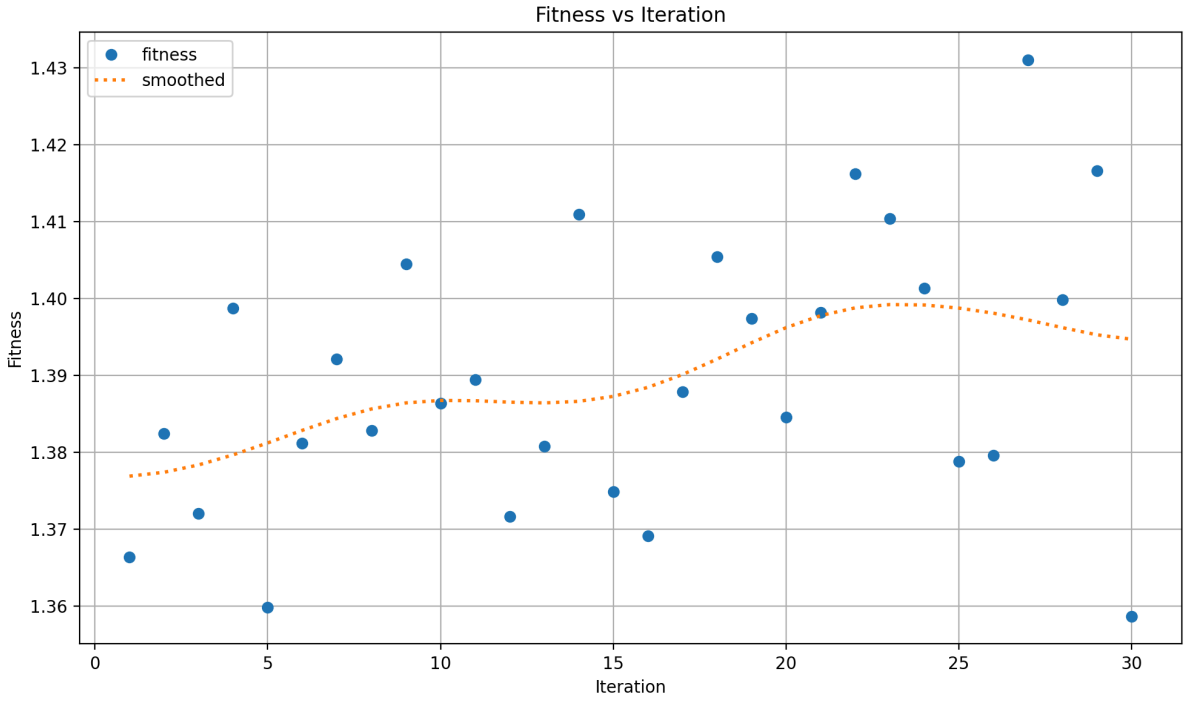


Figure 28 - Second round of fine-tuning for the Ultralytics yolov9c segmentation model trained from scratch. Fitness represents the combination of a variety of accuracy metrics.

While this round of fine-tuning had a slightly better result with the gradual increase of smoothed fitness, each iteration was still suffering from the catastrophic forgetting. The highest fitness model was taken and fine-tuned a third time for 20 iterations at 300 epochs.

During this fine-tuning (*Figure 29*), we see very little improvement, therefore further fine-tuning runs were considered not to be worthwhile.

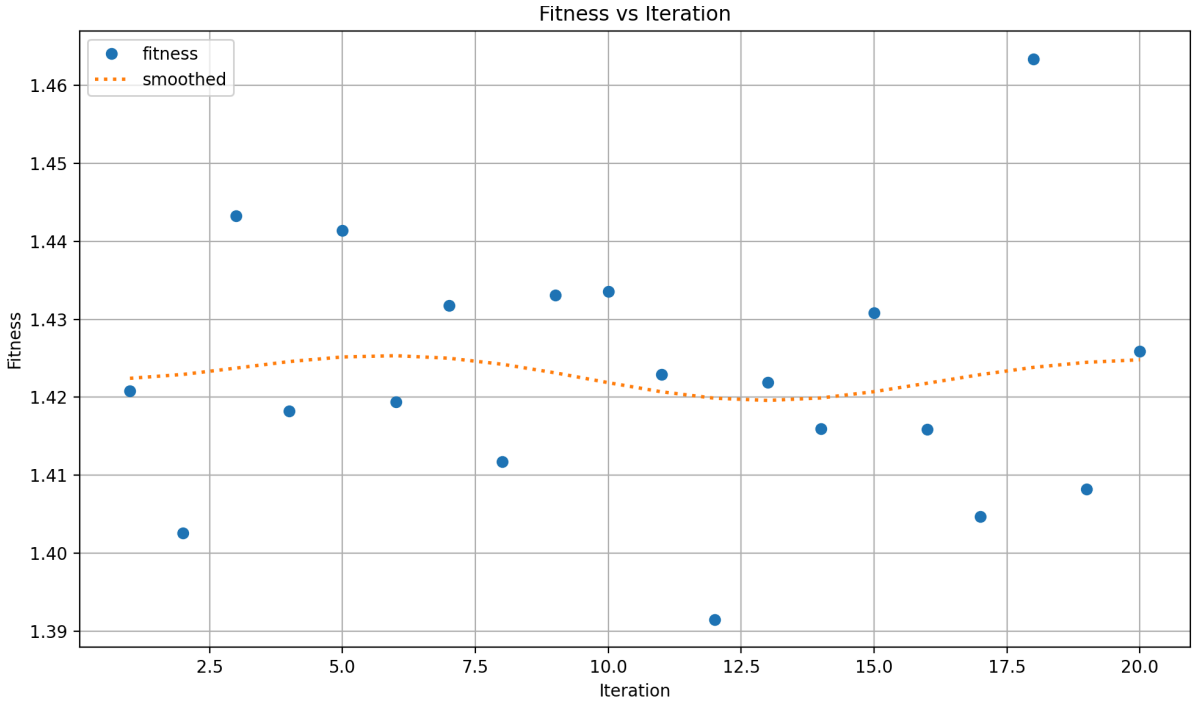


Figure 29 - Third round of fine-tuning for the Ultralytics yolov9c segmentation model trained from scratch. Fitness represents the combination of a variety of accuracy metrics.

The confusion matrix of the best accuracy model can be seen in *Figure 30*. Here, we see the same trends present in the YOLOv9 detection model. The underrepresented classes have a much lower accuracy than the more evenly represented classes, background galaxies are often being incorrectly labelled as radio galaxies, or galaxies are being missed altogether. In the case of the segmentation model, however, we can see that radio galaxies are being incorrectly classified as different radio galaxies at a slightly higher rate. The most logical reason for this is that the segmentation model is more computationally expensive than the detection model, therefore, it takes longer to train. This would result in the classifier portion of the segmentation model being worse than the classifier of the detection model.

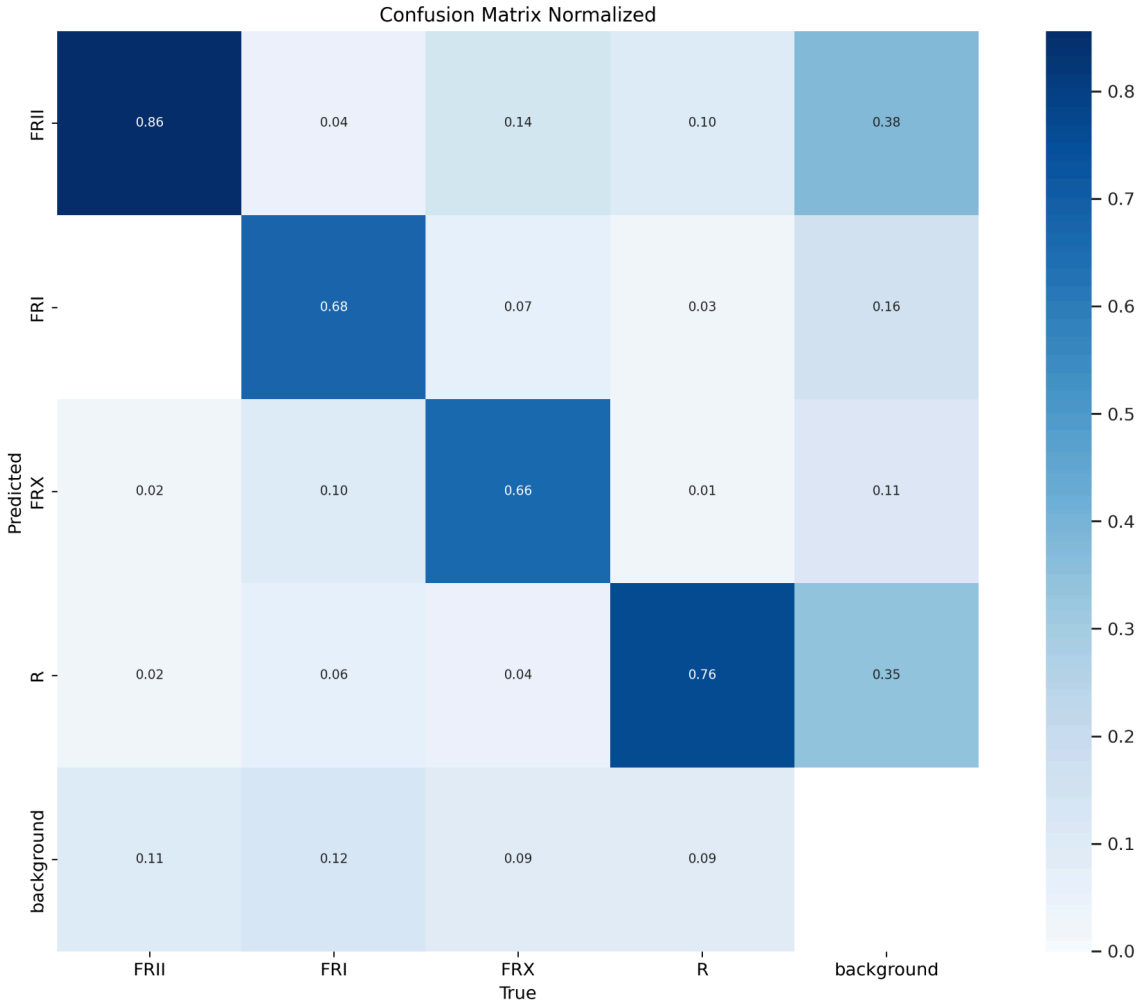


Figure 30: The confusion matrix of the highest fitness Ultralytics yolov9c segmentation model on the test set.

This theory is supported by the mAP50 of the best segmentation model being 0.7974, slightly lower than that of the detection model. While the comparison isn't exactly one-to-one (mAP50 relies on IOU which would be different between a segmentation mask and a bounding box), it is still a useful comparison. The mIOU of the final segmentation model was calculated to be 0.5083.

The output of the model and the ground truth are visualised in *Figure 31*. As we can see, the galaxies are detected and classified in a very similar fashion to the detection model, the only difference being an additional segmentation mask. To the naked eye, these segmentation masks are incredibly accurate. The size of the galaxies within the image mean that the slightest change to the positioning of the mask drastically changes the IOU, therefore, achieving a high IOU for this task is incredibly difficult.

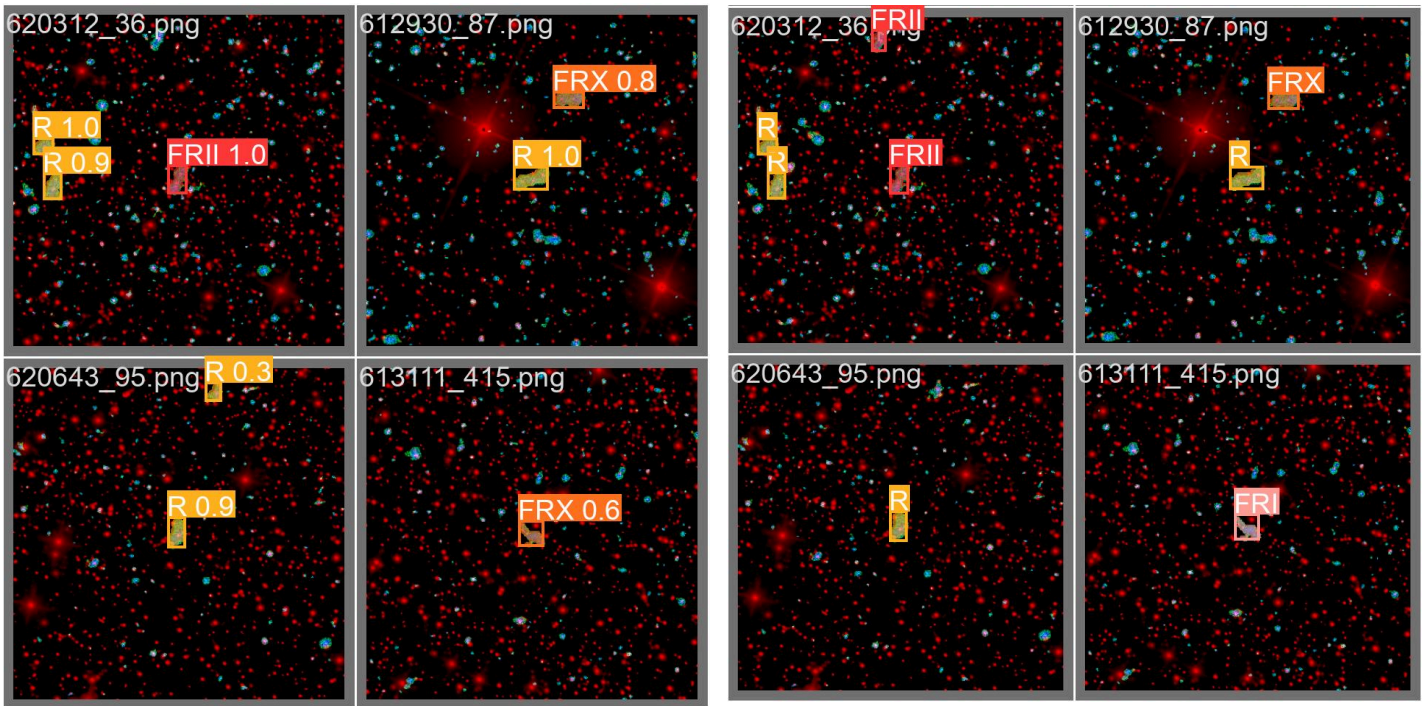


Figure 31: Predictions from the highest mAP50 Ultralytics YOLOv9 segmentation model (left) and the ground truths (right).

Furthermore, whenever a galaxy is segmented correctly, but incorrectly classified (such as the bottom right galaxy in *Figure 31*), the IOU is 0. Thus, we can see that the model's segmentation output is working correctly overall and that the main errors lie within the classification task.

5.1.2.2 Semantic Segmentation with U-Net

Table 6 shows the IoU of each category in our dataset as well as the per-pixel accuracy and F1 score. *Figure 32* shows the normalised per-pixel confusion matrix. The rows of this matrix contain the distribution of predictions for each true label and therefore sum to 1. *Figure 32* visualizes the IoU of several predictions over a center-cropped region to observe fine details. The true masks for these same images are shown in *Figure 32*.

Table 6: Per-category Metrics for Semantic Segmentation with U-Net.

	Background	**FRII**	**FRI**	**FRX**	**R**
Intersection over Union (IoU)	0.9983	0.6342	0.4609	0.4544	0.4293
F1 Score	0.9991	0.7761	0.6310	0.6249	0.6007
Accuracy	0.9992	0.7480	0.6672	0.6176	0.6067

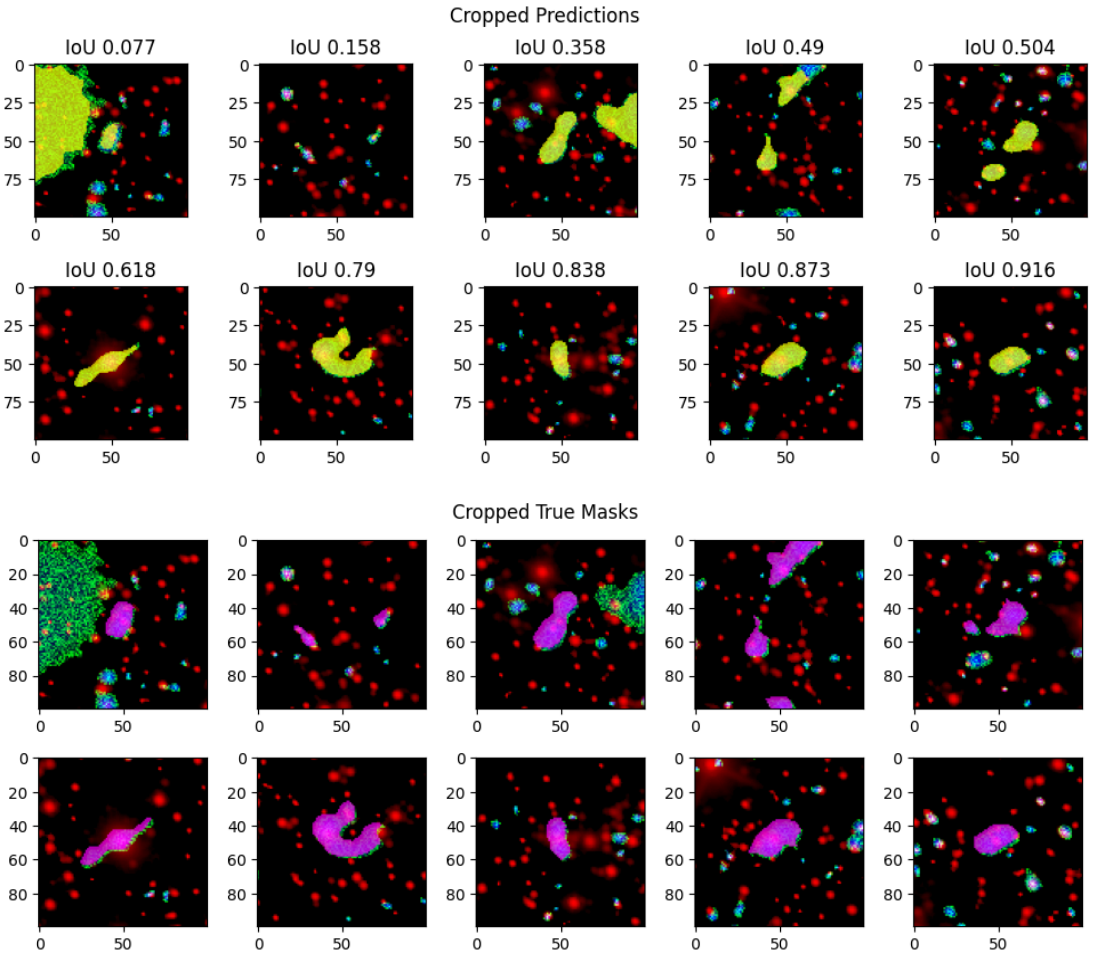


Figure 32 - Cropped Predicted Masks and Corresponding IoUs from U-Net.

Accuracy, IoU, and F1 are highest for the background which occupies the most space in an image. Radio-noise and galaxies are comparatively smaller and it is easy to identify the remaining large black regions as background. Predictably, metrics for FRII are highest amongst galaxy categories as it constitutes 48% of the dataset and the network benefits from additional examples. Metrics for R are consistently the lowest despite constituting a greater percentage of the dataset than FRI and FR-X. By definition an 'R' galaxy is unresolved and hence unclassifiable. This explains why the R category is more frequently confused with the background or 'Bg' than other categories, as seen by the confusion matrix.

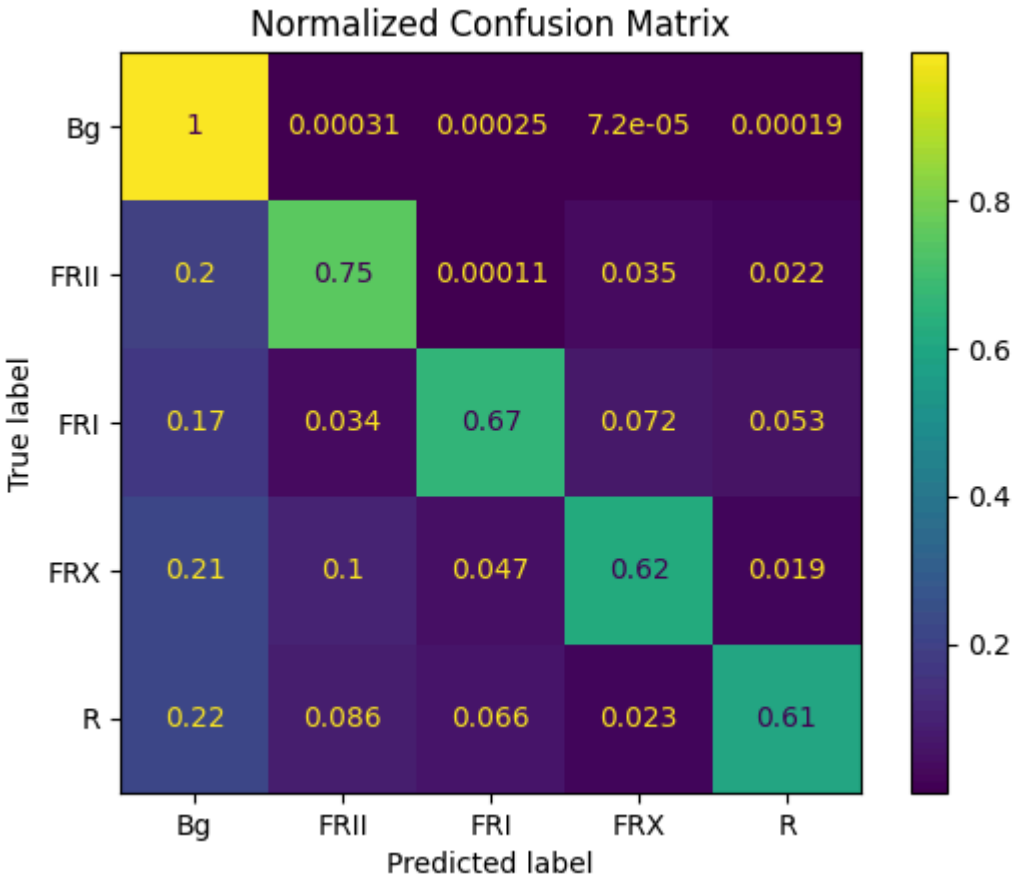


Figure 33 - Pixel Classification Confusion Matrix for U-Net.

The confusion matrix shows that the most frequent misclassification of a radio galaxy is as the background or 'Bg'. This is likely due to the small size of galaxies and the presence of visually similar radio-noise that is sometimes more prominent as seen in *Figure 32*. *Figure 32* also shows that the network frequently 'overactives' over such regions. Instances of 'underactivation' are fewer in comparison. A low IoU in most cases is caused by the misclassification of large background regions as galaxies. These findings suggest that the task of distinguishing galaxies from noise is more difficult than morphological classification.

The mean IoU (mIoU) across Table 4 is 0.595 inclusive of background and 0.495 exclusive of background. As galaxies are small objects and it is harder to delineate their boundaries, the latter is more informative of the performance of the network. The bottom-left image in *Figure 32* shows that an IoU of 0.618 corresponds to a reasonably faithful segmentation mask while lower values can suggest overactivation or underactivation. As such, the performance of the network is less than acceptable. *Figure 32* shows that the bottom-left

predicted mask has a higher overlap with the galaxy than the true mask. This may suggest that the labels are somewhat noisy. This can be integrated into deflate the IoU of small objects.

5.1.2.3 SAM

We report the same metrics as U-Net for semantic segmentation with SAM with and without the use of the LoRA method.

Table 7: Per-category Metrics for Semantic Segmentation with SAM.

		Background	FRII	FRI	FRX	R
Intersection over Union (IoU)	**with LoRA**	0.9975	0.5029	0.2430	0.2339	0.2783
	without LoRA	0.9983	0.4843	0.0439	0.0434	0.2035
F1 Score	**with LoRA**	0.9987	0.6692	0.3910	0.3792	0.4354
	without LoRA	0.9991	0.6525	0.0840	0.0833	0.3381
Accuracy	**with LoRA**	0.9982	0.7494	0.4099	0.4282	0.5048
	without LoRA	0.9994	0.7584	0.0523	0.0466	0.2858

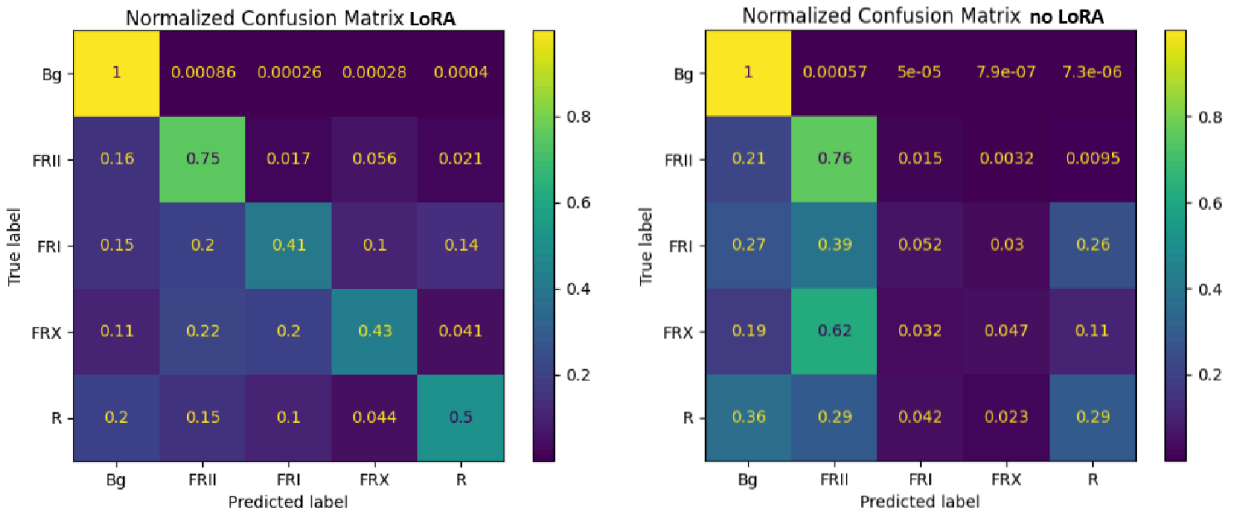


Figure 34: Pixel Classification Confusion Matrix for SAM (Left - with LoRA, Right - without LoRA).

Results obtained by training SAM without LoRA are consistently worse than those from training with LoRA. Table 7 shows that the IoU, F1, and accuracy are less than 0.1 for the minority classes FRI and FR-X without LoRA. Accuracy and F1 are classification metrics whereas IoU is a segmentation metric. As such, these results correspond to a network that is both unable to distinguish morphological categories and delineate the boundaries of galaxies. This is supported by the confusion matrix which shows only ~5% correct pixel

classifications for FRI and FR-X. These findings heavily suggest that the features extracted by pretraining on natural images are not useful for radio imagery even with the use of huge networks and large datasets.

Results obtained by training SAM with LoRA share many similarities with those of previous methods including the best predicted categories and overactivation on large regions of radio-noise. Unlike the U-Net, the worst performing categories are FR-X and FRI as seen in Table 5 and the confusion matrix in *Figure 34*. This suggests that the network struggles to learn from the fewer examples of minority classes despite the advantage of size over U-Net. The confusion matrix shows that the most frequent misclassification of FRI and FR-X is as FRII. This suggests that the network is biased towards the majority class and overpredicts FRII for difficult samples. FR-X is misclassified as FRII with almost the same frequency as FRI. This could be because the features extracted by SAM, even with the use of LoRA, are not rich enough to accurately distinguish between morphological categories.

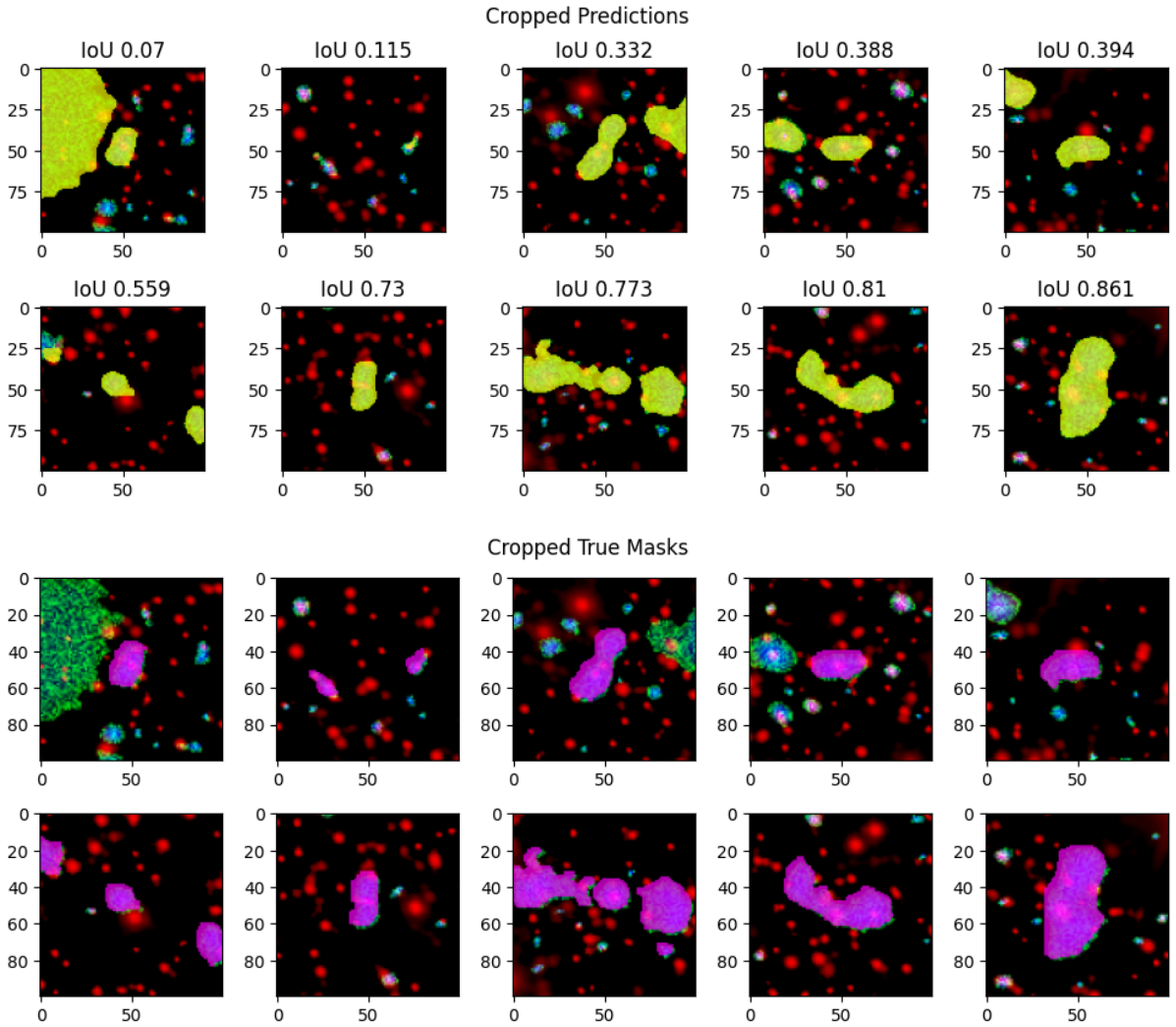


Figure 35 - Cropped Predicted Masks and Corresponding IoUs from SAM with LoRA.

The mean IoU (mIoU) with the use of LoRA is 0.451 inclusive of background and 0.315 exclusive of background. Visualisation of the results in *Figure 35* shows that an IoU less than ~0.6 corresponds to large regions of overactivation and underactivaton. As such, the best results by SAM are not adequate for the semantic segmentation of radio galaxies.

5.1.3 Self Supervised Pre-training

5.1.3.1 Effect of DARGN sampling proportion on DINO Loss

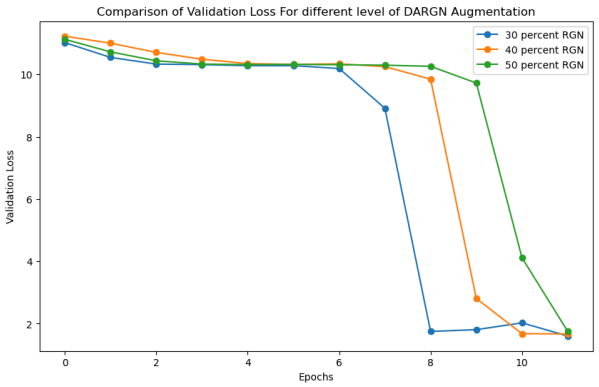


Figure 36 - DINO validation Loss with varying level of augmentation.

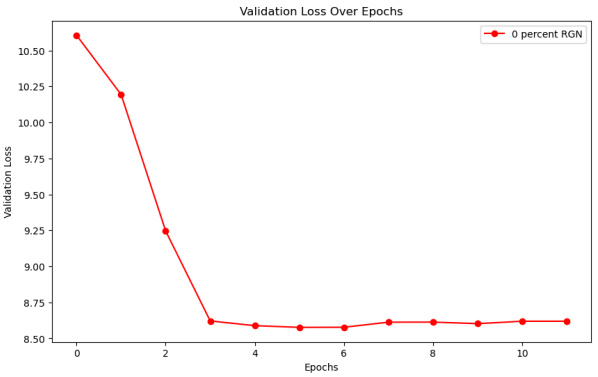


Figure 37 - DINO validation Loss using pure augmented Dataset.

The right figure above (*Figure 37*) depicts the influence that varying levels of DARGN augmentation have on the validation loss of the DINO model, illustrating that higher levels of augmentation correlate with distinct rates of learning. Specifically, the dataset with 70% augmentation (30% RGN) shows a pronounced drop in validation loss at epoch 7, demonstrating rapid model convergence. Conversely, the dataset augmented at 60% (40% RGN) achieves this steep decline by epoch 8, while the dataset with the lowest augmentation (50% RGN) only sees a significant reduction by epoch 10. These trends underscore DARGN's efficacy in enhancing the model's ability to quickly learn from complex and varied data.

This study highlights the DARGN strategy's capability in transforming single-channel radio images from CRUMB into two-channel, multi celestial outputs. Thereby enhancing the DINO model's ability to extract complex features pertinent to the real RadioGalaxyNET data. Additionally, the results suggest the potential of the DINO model for transfer learning and self-supervised fine-tuning, especially in small datasets like the one tested, which consists of approximately 1,800 unlabeled and 1,800 labeled instances. Notably, using the experiment designs outlined in the method section, DINO is able to achieve convergence on the augmented dataset, simulating 15,000 data points from just 1,800 instances. The model previously would require millions of images to train, this indicates a robust augmentation process and the DINO's adaptability to cross-disciplinary tasks.

After observing the above phenomena, an experimental setup using only augmented CRUMB datasets with validation on real RadioGalaxyNET data challenges initial assumptions about the necessity of incorporating real data into training batches when loading data into DINO. Despite expectations, the model showed an even faster convergence of validation loss without utilizing any real data, suggesting that DARGN is effective in enabling the model to learn RadioGalaxyNET characteristics from simulated data. While CRUMB dataset and RadioGalaxyNET are distinctively different datasets taken from different telescopes, the result still raises concerns about the model potentially memorising rather than learning to generalise features from the augmented data. Addressing these concerns would require an unprecedented effort to manually filter out all CRUMB celestial objects within RadioGalaxyNET, ensuring the model is tested against entirely novel data and validating its true learning capabilities.

5.1.3.2 Performance Comparison (Backbone Frozen)

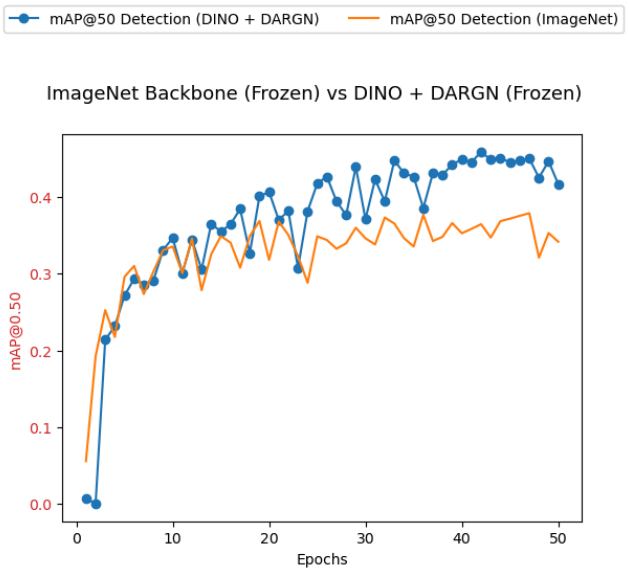


Figure 38 - Image Net Backbone vs DINO + DARGN Backbone (Backbone Frozen).

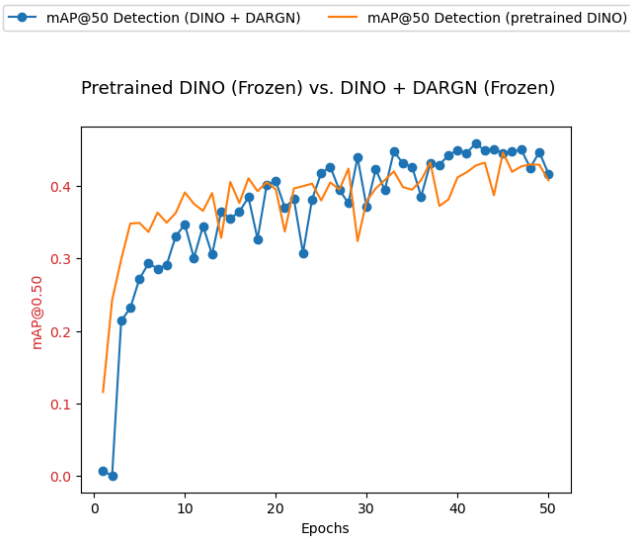


Figure 39 - Pre-trained DINO Backbone vs DINO + DARGN Backbone (Backbone Frozen).

Table 8: Best mAP50 Accuracy (Validation Set).

	DINO + DARGN	**DINO pretrained**	**Supervised Pretrained (ImageNet)**
mAP50 (Detection)	0.459	0.447	0.379

The table above compares the best mAP50 accuracy over 50 epochs for three ResNet50 models. The model using self-supervised fine-tuning and DARGN (blue) outperformed both the version pre-trained with supervised learning on ImageNet (left figure) and the one pre-trained using DINO on millions of general data points (right figure).

The "DINO + DARGN" approach demonstrates an approximate 8% improvement in peak accuracy over the ResNet50 trained via supervised learning on ImageNet. This trend shows consistently higher performance after 25 epochs.

The difference between "DINO + DARGN" and "DINO pretrained" is minimal, around 1%, which some may argue is the result of the stochastic nature of neural network optimisation, however this could also be the result of unsuitable hyperparameters. As described in the method section above, fixed hyperparameters benchmarked on a barebone model are used to ensure a reduced bias is inducted during testing. Therefore, a closer analysis of the accuracy graph over the training process is needed.

The accuracy graph indicates that the model utilising DINO weights pretrained on ImageNet data achieves higher accuracy more quickly prior to 20 epochs but then struggles with complex feature extraction after

achieving set accuracy. In contrast, "DINO + DARGN" maintains consistently higher performance after this point. This early advantage of pre-trained DINO is likely due to the extensive feature filters from training on a massive dataset, which enables the model to recognise simple patterns quickly. However, when faced with more challenging tasks, the lack of exposure to real radio galaxy data diminishes its effectiveness compared to "DINO + DARGN," which is fine-tuned on a diverse set of radio galaxy data.

To enhance our understanding of the feature extraction capabilities of our backbones, we employ Class Activation Maps (CAMs) [92] to elucidate the areas where convolutional filters are most active in extracting features. A heatmap is generated using CAMs, where regions of significant interest are marked in dark red, and areas of lesser interest are highlighted in blue. This visualisation aids in pinpointing the specific regions within the input that influence the model's decision-making process.

Note: While application of Class Activation Maps are explored in prior works, it is not deemed relevant for this particular project. Please refer to the "Class Activation Map and Prior Investigations" Section in "Appendix A1" for a more detailed explanation of Class Activation Map and previous findings that inspired the development of the current method.

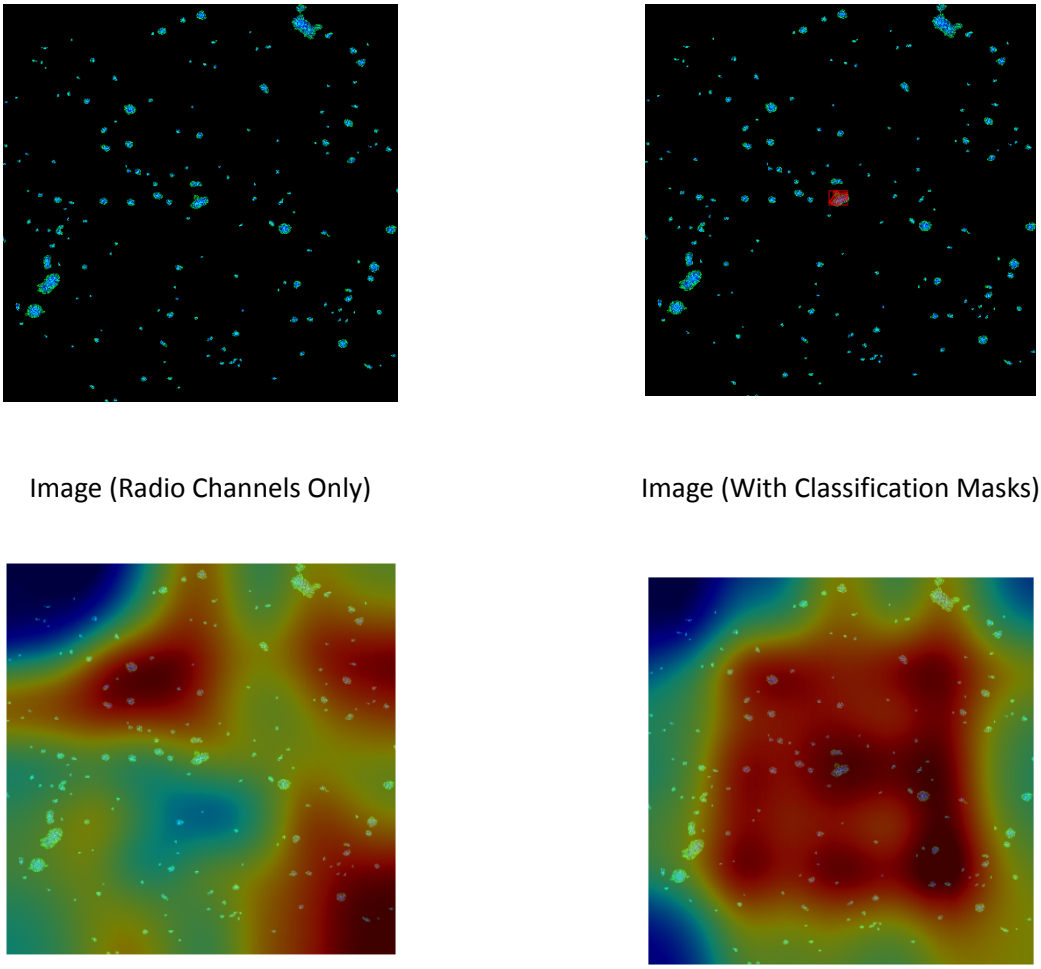


Figure 40 - Visualization of Testing Data and CAM for different Models.

An effective backbone should extract features primarily from regions where the detected object resides. In the sample image provided, the desired region is approximately at the centre, as indicated by the red mask in the top right image, which contains the target object. Although the backbone trained using "DINO + DRAGN" is not flawless—the most intensive region (dark red) includes some irrelevant instances—it is notable that the region of highest intensity encompasses the location of the desired object. Conversely, the backbone pre-trained on ImageNet via supervised methods fails to discern the desired radio signal from others, focusing on random regions that offer minimal contribution to the detection of desired Radio Galaxies.

5.1.3.3 Performance Comparison (Backbone Unfrozen)

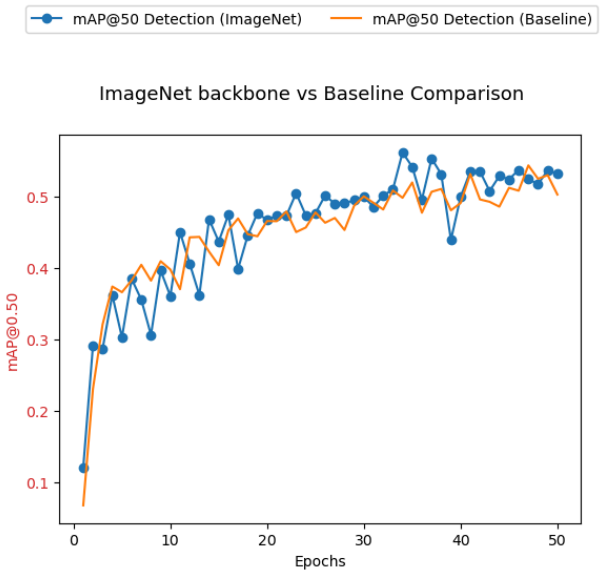


Figure 41 - Image Net Backbone vs Baseline Comparison.

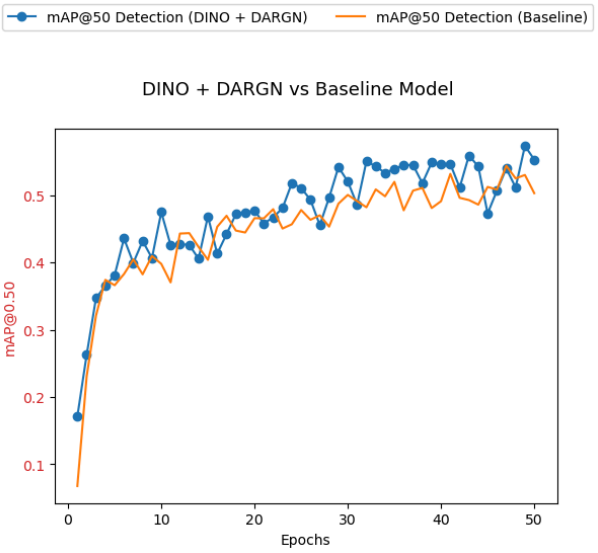


Figure 42 - DINO + DARGN vs Baseline Comparison.

Table 9 - Best mAP50 Accuracy (Validation Set).

	DINO + DARGN	**Supervised Pretrained (ImageNet)**	**Baseline (Trained from scratch)**
mAP50 (Detection) Validation	0.573	0.562	0.543

The analysis of the mAP50 accuracy for three models trained on a galaxy dataset with an unfrozen backbone reveals that the "DINO + DARGN" approach achieves the highest accuracy at 0.573. This outperforms the Supervised Pretrained (ImageNet) model and the Baseline (trained from scratch) model, which scored 0.562 and 0.543 respectively. This demonstrates the effectiveness of combining self-supervised learning with DARGN augmentation, highlighting its superior feature extraction and adaptation capabilities for specialised datasets compared to conventional supervised learning and training from scratch. The reasons contributing

to the minimal differences in performance between the models have been previously discussed in a similar context and will not be reiterated here.

Table 10 - Best mAP50 and mIoU Accuracy (Testing Set).

	DINO + DARGN	**Supervised Pretrained (ImageNet)**	**Baseline (Trained from scratch)**
mAP50 (Detection) Test set	0.628	0.5264	0.588
mIoU (Segmentation) Test set	0.684	0.6740	0.655

The DINO + DARGN model distinctly outperforms its counterparts in addressing the core research questions of detecting and segmenting radio galaxies, as demonstrated in comparative testing with three models. This model achieves the highest mAP50 score for detection at 0.628, surpassing the supervised pretrained (ImageNet) model's 0.5264 and the baseline 0.588. For segmentation, DINO + DARGN leads with an mIoU of 0.684, slightly ahead of the ImageNet model at 0.674 and significantly exceeding the baseline of 0.655. These results highlight the efficacy of DINO + DARGN self-supervised learning and advanced augmentation techniques, which make it particularly adept at the intricate tasks of classifying, detecting, and segmenting radio galaxies.

Variability between test and validation performance is attributed to differences in data distribution, model fitting, and dataset characteristics. Particularly with RadioGalaxyNet, the smaller dataset size might lead to performance anomalies where a model's superior results during certain validation epochs arise from random chance rather than a genuine understanding of the data. This was observed with the ImageNet model, which excelled in the validation set but faltered in the testing set. In contrast, DINO + DARGN consistently demonstrated superior performance across both testing and validation sets, underscoring the value of self-supervised pre-training in accurately identifying generic features of radio galaxies.

5.1.3.4. Further Discussions

- In the DINO paper[58], the authors have found that the teacher model consistently outperformed the student model, in our study we have also explored the difference in performance of two models and it can be found that two models exhibit similar performance. Figure representation of the results can be found in the “Appendix A2 - Additional Data” The Student and Teacher network does not make much of a difference.

- By applying basic training and fine-tuning techniques to the Baseline Faster-RCNN using two channels from RadioGalaxyNET, we achieved an mAP50 for detection of 0.588. This result is comparable to the best models from the RadioGalaxyNET authors[7] and significantly surpasses the mAP50 of 0.484 recorded by the "Gal-Faster RCNN," which underwent training for 20,000 epochs compared to only 50 in our project. These findings underscore the potential impact of

hyperparameter fine-tuning and suggest that the inferred channel might not enhance, and could sometimes detract from, the performance of radio galaxy detection.

- The primary objective of this project is to assess both the accuracy and training speed of the self-supervised, pre-trained "DINO + DARGN" backbone in comparison to other backbones and the baseline model. Despite the Faster-RCNN with the "DINO + DARGN" backbone showing potential for further improvement beyond 50 epochs, we decided to terminate the training at this point. This decision was informed by data from the 200-epoch training of a baseline Faster RCNN, detailed in "Appendix A2 - Additional Data," which indicates that the baseline model begins to overfit after approximately 50 epochs.

5.2. Findings

5.2.1. Detection Findings

Table 11: Summary of the best mAP50 from every model as well as the previous best model available in literature, Gal-DINO.

Models	mAP_{50}
YOLOv9 (Wang et al. Version)	0.817
YOLOv9 (Ultralytics Version)	0.798
Faster RCNN (DINO + DARGN)	0.628
Faster RCNN (Baseline)	0.588
Gal-DINO (DETR with Improved deNoising anchOr boxes)[93]	0.602

Here, we can see that Gal-DINO, the previous best detection model for radio galaxies, is significantly outperformed by both YOLOv9 models and our Faster RCNN (DINO + DARGN) model. Importantly, DINO (self distillation with no labels) used in this project is different to the DINO (DETR with Improved deNoising anchOr boxes) [93] used in the RadioGalaxyNET paper. Therefore, our radio galaxy detection models can be said to be field-leading.

When analysing the training Gal-DINO model, we can see that while Gupta et al. did alter the architecture of the traditional DINO model to include keypoint detection, they did not optimise the model through fine-tuning, meaning that our models were able to outperform it quite easily.

Our Faster-RCNN trained with DINO and DARGN was the next most accurate model, achieving an mAP50 of 0.628. Here we demonstrate how self-supervised fine-tuning and data augmentation can improve a baseline model. While this detection model is quite powerful, it is also quite old, with the newer YOLOv9 models outperforming it.

Both YOLOv9 models achieved a very similar mAP50, the Wang et al. version achieving 0.817 and the Ultralytics version achieving 0.798. While the Ultralytics version was capable of being fine-tuned more easily, with access to specific hyperparameter optimisation functions, it was outperformed by the Wang et al. model which had to be manually optimised. This indicates that the Wang et al. model is generally more suited to our task. There are slight differences between the two YOLOv9 architectures which could be the cause of the discrepancy - the most likely either being the DualDDetect head in the Wang et al. model being replaced by a much simpler Detect head in the Ultralytics model or the difference in train scripts between the models. Furthermore, it can be said that YOLOv9's specialisation in information retention makes it better suited to our task than traditional detection methods.

5.2.2. Segmentation Findings

Table 12: Summary of the best mIOU from every segmentation model as well as the type of segmentation being performed.

Models	**Type**	$mIoU$
YOLOv9	Panoptic	0.508
Faster-RCNN (DINO + DARGN) + segmentation head	Panoptic	0.684
Faster-RCNN (Baseline) + segmentation head	Panoptic	0.655
SAM with LoRA	Semantic	0.451
U-Net	Semantic	0.595

Comparing the mIoU of all segmentation networks shows that the Faster-RCNN excels above other networks with and without pretraining. Although this is surprising as the network specialises in detection, we note that the Faster-RCNN disentangles the task of identifying radio galaxies from the task of categorising them. Our results consistently show that machine learning methods struggle to distinguish small radio galaxies from the background and overactivate on large regions of radio-noise. The Faster-RCNN uses a specialized CNN to identify regions of interest before categorisation. This two-step method is useful for small objects in noisy images.

Although YOLOv9 is a similar two-step method, it is outperformed by a lightweight U-Net. This suggests that the success of machine learning methods for radio imagery is not bottlenecked by the number of parameters in modern networks. In fact, the largest network SAM performs the poorest. This observation is in concurrence with existing literature which frequently concerns the use of smaller architectures such as 18 and 50-layer ResNets for computer vision tasks in radio imagery. Our U-Net is also instantiated with ConvNeXt network blocks that mimic the skip connections of a ResNet. The Faster-RCNN uses a ResNet feature extractor. The skip connection is noted to increase the stability of training by protecting against vanishing and exploding gradients. Our results confirm its usefulness for radio imagery as suggested by the literature.

SAM, a large ViT-based method, is outperformed by convolutional networks. This finding is surprising given the success of ViTs across domains. We note that training SAM is prohibitively expensive so increasing our training duration given more resources could increase performance. A more compelling explanation concerns the fact that the attention mechanism of ViTs is designed to capture global long-range dependencies without the inductive bias of the importance of neighbouring image regions. This may be useful for natural imagery but it is counterintuitive when applied to the segmentation of small galaxies. By dividing an image into smaller patches, the ViT wastes computation and parameters on patches that likely miss the object of interest.

SAM performs poorly despite pretraining on millions of natural images. This suggests that such pretraining is not useful for radio imagery. Pretraining on a dataset of radio galaxies does improve the performance of Faster-RCNN. As such, the paradigm of pretraining large networks for improved downstream performance is applicable to radio imagery when the dataset used for pretraining shares some characteristics with the dataset used for fine tuning.

In summary, we find that two-step segmentation methods perform best for small galaxies in noisy images. Disentangling the task of morphological categorisation from the task of differentiating galaxies and noise is key to this success. Convolutional networks are well-suited to radio imagery as they use localised kernels that give greater importance to nearby image regions. Some useful architectures are Residual Networks (ResNets) and ConvNeXt, which use similar skip connections to integrate features at different scales and improve the stability of training. Machine learning methods for radio imagery are not restricted to use large networks for good results. Pretraining on radio imagery datasets instead of natural imagery datasets can improve the performance of these networks.

All code for the generated models and pre-trained weights can be found on GitHub as per our aims.

5.3. Limitations and Future Work

5.3.1 Key Point Detection and Custom Architecture

A key limitation of this project, compared to preceding work, is the lack of key point detection, which has been highlighted as crucial in the work by the D61 team from CSIRO[7]. This capability is particularly sought after by astronomers applying these models to their projects. To address this, a significant area for future research would be the refinement and design of custom architectures capable of integrating key point detection. Building on the foundational work of RadioGalaxyNET, such modifications may also enhance the detection accuracy of our models. Given that the YoloV9 model currently shows promising results after extensive testing and fine-tuning, further improvements are anticipated through continued enhancements to the YOLO architecture.

5.3.2 Better YOLO

One significant limitation of our current project arises from constraints related to the timeframe and the scale of the dataset available. The restricted duration allotted for the project has limited our ability to conduct extensive experimentation and optimisation. Additionally, the limited dataset has not only hindered the comprehensive training of the YOLO object detection and segmentation model but also exacerbated issues of class imbalance. This imbalance skews the model's learning, leading to a preference for specific classes, thereby undermining the model's ability to learn diverse features effectively. In the future we hope to:

- **Extended Training:** Increase the number of training epochs and instances if time permits, to deepen the model's learning and enhance generalization.
- **Layer Freezing:** Implement a layer freezing protocol to prevent catastrophic forgetting, thus preserving the integrity of previously learned features while updating others[94].
- **Hyperparameter Analysis:** Conduct detailed investigations into the trends of hyperparameters to understand interdependencies and optimise performance.
- **Optimiser Change:** Explore the potential benefits of changing the optimisation algorithm to improve learning dynamics.
- **Weighted Data:** Introduce weighted training approaches to address class imbalance, ensuring fair representation and accuracy across all classes.

5.3.3 SAM Not Segmenting Everything

Despite the Segmenting Anything Model (SAM) being a foundational model for general segmentation, it exhibits significant shortcomings when applied to specialised fields, as demonstrated by recent analyses of SAM on medical data[95]. SAM’s performance heavily depends on the input prompts and shows poor results in salient object detection tasks. Additionally, SAM is complex to train, complicating the process of scaling up training batches, even with multiple GPUs. To address these challenges, we propose two methods:

- **Multistage Segmentation:** Leveraging bounding box prompts from another object detection model, such as YOLO, SAM could be fine-tuned to perform semantic segmentation using these inputs.

- **Architecture Modification:** Following the approach of Fast Segment Anything Model (FastSAM)[96], which uses a YOLO backbone for the image encode

5.3.4 Better self-supervised method

The DARGN model is currently used to facilitate fine-tuning on the DINO model with a ResNet50 backbone, demonstrating superior performance in feature extraction from CRUMB data with RadioGalaxyNET datasets. However, concerns remain regarding potential data contamination, as the full tagging of instances in both datasets is unverified. This raises doubts about whether our model might inadvertently be trained with some pre-knowledge. Previous tests in the result section have shown that DINO trained on generic images competes closely with DINO trained in conjunction with DARGN. Future investigations might focus on:

- **Cross-validation Methods:** Ensuring no pre-existing data from the CRUMB dataset is reused in the RadioGalaxyNET dataset could enhance model integrity.
- **Fine-tuning Pretrained Models:** Examining the impacts of fine-tuning DINO with pretrained weights using the DARGN could provide insights into achieving more reliable and robust performance.

6. Conclusion

This study evaluates state-of-the-art methods for the detection and segmentation of radio galaxies in the recent RadioGalaxyNET dataset. Some challenges we address include the scarcity of labelled data and the stark contrast of radio imagery to the conventional datasets used with these methods. Our proposed methods outperform existing solutions for detection. We are among the first to explore supervised segmentation and achieve qualitatively and quantitatively significant results. We develop our own pipeline for the self-supervised pretraining of a foundation model for radio-imagery for quantitatively improved performance. Our findings include key insights into the choice of architecture and methods suitable for radio-imagery that confirm and extend the current literature.

Our detection models include YOLOv9 (mAP 0.817) and Faster-RCNN (mAp 0.628) which outperform the existing solution Gal-DINO (mAP 0.602) by significant margins. Our segmentation models include YOLOv9 (mIoU 0.508), Faster-RCNN (mIoU 0.684), U-Net (mIoU 0.595), and the foundation model SAM (mIoU 0.451). Our best mIoU of 0.684 corresponds to excellent overlap with true masks as seen in visualizations. Some crucial insights we obtained include the superiority of traditional Convolutional Neural Networks (CNNs) over the powerful and emerging Vision Transformers (ViTs) for radio imagery. These CNNs encode the importance of neighboring image regions as a useful inductive bias whereas ViTs divide images into patches that frequently miss small galaxies. We also find that differentiating galaxies from noise or background is a harder task than morphological categorization which is more frequently explored in literature. We show that disentangling these two tasks leads to improved performance.

Furthermore, the research explored self-supervised learning to alleviate the burden of scarce labelled data, particularly using the DINO pre-training technique to enhance deep learning models. Through comprehensive exploration of DINO, ultimately a self-supervised ResNet-50 backbone is developed with the help of DARGN augmentation. The backbone is tested using a Faster R-CNN to demonstrate an improved performance of 5-10% over backbones trained through supervised approaches on ImageNet data, supporting the viability of self-supervised learning in this context. This work will also open the door for future research in different architectures trainable under similar preprocessing of the task of radio astronomy.

This study has not only established new benchmarks in the detection and segmentation of radio galaxies but also introduced novel approaches for data augmentation and model training. Future work should continue to extend the use of self-supervised learning models and assess their adaptability to various astronomical datasets, potentially expanding the scope of machine learning applications in radio astronomy. However, further investigation into the development of a fully self-supervised learning method and the combination of multiple models is still needed.

7. Reflection on Project Management

7.1. Project Scope

The major objective of this project is to explore the role Machine Learning can play in the exploration of galaxies using datasets generated from radio telescopes. The exploration process will mostly be performed using publicly available datasets such as RadioGalaxayNET, Mirabest (CRUMB).

Quality and Integrity: The project is scoped to be a research project. The team is expected to conduct research in a manner that is up to the standards and integrity of creating publishable work.

Model and Training: The models developed for this project will involve YOLOv9 (You Only Look Once), a real-time object detection model, SAM (Segment Anything Model) an image segmentation model and finally DINO integrated with FasterRCNN. The models are expected to be trained under supervision or weakly self-supervision.

Cross-disciplinary Research: The project is scoped to be a cross-disciplinary research project involving the field of cosmology, astrophysics, and machine learning.

Table 13: Scope Table.

In Scope	Out of Scope
1. The study produced by the team must present results and evidence of original research. Experiments, modelling, and other analyses are conducted to a high standard. 2. The segmentation model is expected to be trained using the segmentation masks from the RadioGalaxyNET dataset. 3. The detection model is expected to be trained using the bounding boxes from the RadioGalaxyNET dataset. 4. Different model training environments include: MASSIVE[30], Monash AI lab environment and Google Colabs. 5. Self-supervised pre-training or other methods should also be explored to improve the model accuracy. 6. Pytorch will be the main machine learning framework used for the model. 7. Reproduction of state of the art models created previously are also to be expected.	1. The project should focus on researching the Computer Vision and machine learning aspects of vision models. Astrophysics side implications, while important, will not be the focus of this project. 2. Labelled data will be used in the model training; therefore, the research will not focus on developing a fully unsupervised model. 3. Model architectures such as Tensor flow and Keras will not be used for this specific project. 4. The size of the model developed should be trainable under 24 GB of RAM. 5. The project will mostly be focusing on the classification, detection and segmentation of data from radio telescopes. Other forms of data: e.g. optical telescope data, VOC12 data may be leveraged (or partially used) during the pre-training of the machine learning model, but will not be the main focus of the project.

8. The model developed will mainly be tasked with the classification, segmentation and detection of Radio Galaxies into four types: FR-I, FR-II, FR-X, and R.	

7.2. Project Plan & Timeline

Please refer to the “Appendix E” for the more detailed timeline for the management of the project.

7.2.1 Task Completion Status

Table 14: Simplified Short Term Plans.

Task	Revised Due Date	Original Timeline	Status	Comments
Finish implementation of public models without method changes	4th May 2024	27th March 2024	Completed	Finished
Finish SAM, FasterRCNN and DINO model implementations with project specific changes	10th May 2024	26th April 2024	Completed	Finished
Integrate FasterRCNN with DINO	10th May 2024	4th May 2024	Completed	Finished, however further modification to DINO is made as more knowledge is gathered.
Poster and video soft deadlines	15th May 2024	10th May 2024	Completed	Finished, these are the initial drafts of the deliverables.
Final Report soft deadline	22nd May 2024	17th May 2024	Completed	Finished, this is the draft of the final report deliverable.
Poster and video deliverables	17th May 2024	17th May 2024	Completed	Finished

Final Report deliverable	24th May 2024	24th May 2024	Completed	Finished

Table 15: Plan alteration.

Task ID	Original Task Description	Modified Task Description	Reason for the change
19	Draft up a report explaining what has been done and what has been found during holiday	Come together and discuss models of interest, in particular recent papers that have been published in the field. Decide whether these are relevant to our work.	The team did not get to do as much work as originally planned over the break between semesters, due to overlapping schedules. Therefore, the workload was reduced and the task was made less formal.

7.3. Reflection on Project

The overall output of our project is not only tangible but also significant. We were able to produce meaningful results that furthered the field of astronomical image classification and segmentation. As a team, we successfully split the work in a way that meant workloads were allocated fairly and this allowed everyone to contribute to the project output.

We maintained strong communication through consistent Messenger updates and meetings that occurred one to two times a week. One thing that worked particularly well was hosting exclusive meetings between team members that were working on the same model, this way those who were not involved could focus on their outputs.

It should be noted that each team member was able to individually produce or significantly contribute to an object detection model. Initially, we planned to produce one effective model; however, given the way we divided the work, we were able to go far beyond this and produce three meaningful outputs. With Zach focusing on Yolov9, Suleman on the Segment Anything Model (SAM), and Yide and Katherine on the integration of DINO and FasterRCNN.

A useful tool we implemented was an activity tracker, almost like a Kanban Board (see below), which kept track of current individual tasks and their status, for example: in progress or next up.

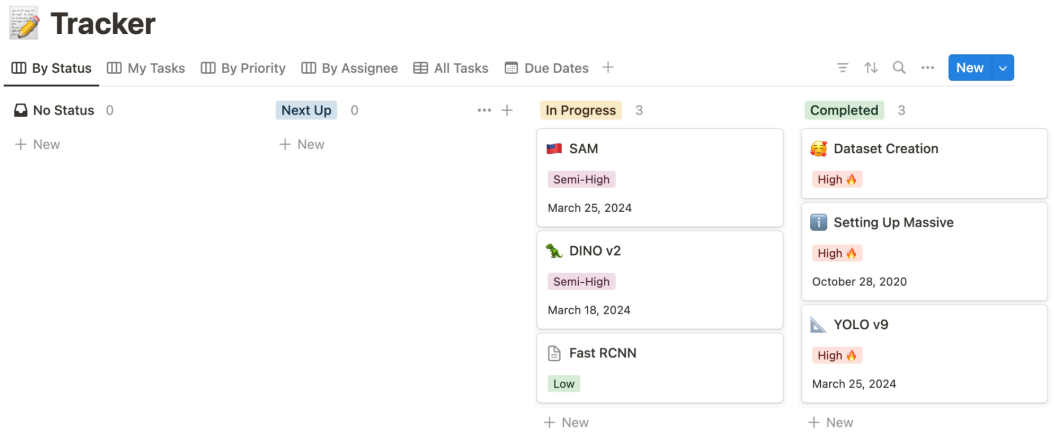


Figure 43 - Sample Photo of the Notion Tracker Made for this project.

One issue that did arise was the exit of one of our team members during the second phase of the project. Unfortunately, with someone leaving the project, this meant the workload had to be shifted amongst the remaining team. Although initially challenging, we did manage to effectively do this. Mainly through consistent meetings and a timeline that was created so that each member could note down their workloads in other areas of their life. Using this timeline, we could then fairly distribute any outstanding tasks that were left by the departing member.

Another setback that we faced in the first semester was the inability to access the CSIRO's Data61 dataset. Initially we had planned to use these images to train our models, but due to the delayed publication of the paper, this was not something we could achieve. To overcome this, we initially investigated the models using alternative datasets, a Summary of our findings can be found in Appendix A1 - Prior works. Although these were not the ideal images we wanted to use, they allowed us to gain some basic understanding of classification of radio galaxies.

In the future, it would be beneficial to continue the exploration over periods of break. Given the large break between the semesters, we initially decided to continue work to a lesser degree, such that we could come back at the beginning of the year and share our findings. Due to overlapping schedules and various other unpredictable factors, this was not maintained or achieved. If there was a clearer direction with which areas we wanted to research over the break, this could have set us up for further success in the semester, and potentially allowed for greater exploration into more novel models that we lacked time to explore. This was also a catalyst in the need to move soft deadlines back, since the beginning of the second semester required more research than initially planned.

8. References

[1] M. H. Jones, R. J. Lambourne, and D. J. Adams, Eds., "3.3.3 Radio Galaxies," in *An introduction to galaxies and cosmology*, Milton Keynes : Cambridge, UK ; New York: Open University ; Cambridge University Press, 2004, pp. 142–144.

[2] D. Lynden-Bell, "Galactic Nuclei as Collapsed Old Quasars," *Nature*, vol. 223, no. 5207, pp. 690–694, Aug. 1969, doi: 10.1038/223690a0.

[3] Hsiang-Yi Karen Yang, *Two types of FRI & FRII Galaxies*. [Online]. Available: http://www.phys.nthu.edu.tw/~hyang/BlackHole/LessonPDF/L11-Jets.pdf

[4] Oei, Martijn, *False-colour image showing a 2048 arcsecond by 2048 arcsecond solid angle centred around Alcyoneus' host galaxy, J081421.68+522410.0, with LOFAR radio data at 144 MHz (orange) and WISE infrared data at 3.4 micron (blue) overlaid. To draw attention to the radio emission, the infrared emission has been blurred.* [Online]. Available: https://en.wikipedia.org/wiki/Alcyoneus_(galaxy)#/media/File:GRGAlcyoneusIRBlueRadioOrange.png

[5] R. Pool, "Drowning in Data," *International Society for Optics and Photonics (SPIE)*, May 01, 2020. [Online]. Available: https://spie.org/news/photonics-focus/mayjun-2020/square-kilometer-array-big-data

[6] Dragonfly Media, *Antennas of CSIRO's ASKAP telescope at the Murchison Radio-astronomy Observatory in Western Australia.* 2012. [Digital Photograph]. Available: http://www.scienceimage.csiro.au/image/2161

[7] N. Gupta, Z. Hayder, R. P. Norris, M. Huynh, and L. Petersson, "RadioGalaxyNET: Dataset and Novel Computer Vision Algorithms for the Detection of Extended Radio Galaxies and Infrared Hosts." arXiv, Nov. 30, 2023. doi: 10.48550/arXiv.2312.00306.

[8] Ultralytics, "Ultralytics YOLOv8 Tasks." Accessed: May 24, 2024. [Online]. Available: https://docs.ultralytics.com/tasks

[9] B. L. Fanaroff and J. M. Riley, "The Morphology of Extragalactic Radio Sources of High and Low Luminosity," *Mon. Not. R. Astron. Soc.*, vol. 167, no. 1, pp. 31P-36P, Apr. 1974, doi: 10.1093/mnras/167.1.31P.

[10] M. J. Hardcastle and J. H. Croston, "Radio Galaxies and Feedback from AGN Jets," *New Astron. Rev.*, vol. 88, p. 101539, Jun. 2020, doi: 10.1016/j.newar.2020.101539.

[11] B. Mingo *et al.*, "Revisiting the Fanaroff–Riley Dichotomy and Radio-galaxy Morphology with the LOFAR Two-Metre Sky Survey (LoTSS)," *Mon. Not. R. Astron. Soc.*, vol. 488, no. 2, pp. 2701–2721, Sep. 2019, doi: 10.1093/mnras/stz1901.

[12] A. D. Kapińska *et al.*, "Radio Galaxy Zoo: A Search for Hybrid Morphology Radio Galaxies," *Astron. J.*, vol. 154, no. 6, p. 253, Nov. 2017, doi: 10.3847/1538-3881/aa90b7.

[13] "Linear/Fully-Connected Layers User's Guide," NVIDIA Docs. Accessed: May 24, 2024. [Online]. Available: https://docs.nvidia.com/deeplearning/performance/dl-performance-fully-connected/index.html

[14] M. Sahay, "Neural Networks and the Universal Approximation Theorem," Medium. Accessed: May 24, 2024. [Online]. Available: https://towardsdatascience.com/neural-networks-and-the-universal-approximation-theorem-8a389a33d30a

[15] A. Jain, "All about convolutions, kernels, features in CNN," Medium. Accessed: May 24, 2024. [Online]. Available: https://medium.com/@abhishekjainindore24/all-about-convolutions-kernels-features-in-cnn-c656616390a1

[16] R. M. Battleday, J. C. Peterson, and T. L. Griffiths, "From convolutional neural networks to models of higher-level cognition (and back again)," *Ann. N. Y. Acad. Sci.*, vol. 1505, no. 1, pp. 55–78, Dec. 2021, doi: 10.1111/nyas.14593.

[17] G. Boesch, "Vision Transformers (ViT) in Image Recognition - 2024 Guide," viso.ai. Accessed: May 24, 2024. [Online]. Available: https://viso.ai/deep-learning/vision-transformer-vit/

[18] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, “ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias.” arXiv, Dec. 23, 2021. Accessed: May 24, 2024. [Online]. Available: http://arxiv.org/abs/2106.03348
[19] M. Awais *et al.*, “Foundational Models Defining a New Era in Vision: A Survey and Outlook.” arXiv, Jul. 25, 2023. doi: 10.48550/arXiv.2307.13721.
[20] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” arXiv, Jan. 06, 2016. doi: 10.48550/arXiv.1506.01497.
[21] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, Art. no. 4, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection.” arXiv, May 09, 2016. doi: 10.48550/arXiv.1506.02640.
[23] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information.” arXiv, Feb. 28, 2024. Accessed: May 24, 2024. [Online]. Available: http://arxiv.org/abs/2402.13616
[24] Ultralytics, “YOLOv9.” Accessed: May 24, 2024. [Online]. Available: https://docs.ultralytics.com/models/yolov9
[25] “YOLOv9: Advancing the YOLO Legacy.” Accessed: May 24, 2024. [Online]. Available: https://learnopencv.com/yolov9-advancing-the-yolo-legacy/
[26] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, “Designing Network Design Strategies Through Gradient Path Analysis.” arXiv, Nov. 09, 2022. doi: 10.48550/arXiv.2211.04800.
[27] X. Du *et al.*, “SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 11589–11598. doi: 10.1109/CVPR42600.2020.01161.
[28] G. Ghiasi *et al.*, “Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 2917–2927. doi: 10.1109/CVPR46437.2021.00294.
[29] Y. Li, Y. Chen, N. Wang, and Z. Zhang, “Scale-Aware Trident Networks for Object Detection.” arXiv, Aug. 19, 2019. doi: 10.48550/arXiv.1901.01892.
[30] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, “Panoptic Segmentation,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 9396–9405. doi: 10.1109/CVPR.2019.00963.
[31] “Introduction to Semantic Segmentation.” Accessed: May 24, 2024. [Online]. Available: https://encord.com/blog/guide-to-semantic-segmentation/
[32] S.-H. Tsang, “Review: FCN — Fully Convolutional Network (Semantic Segmentation),” Medium. Accessed: May 24, 2024. [Online]. Available: https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1
[33] P. Wang *et al.*, “Understanding Convolution for Semantic Segmentation,” Mar. 2018, pp. 1451–1460. doi: 10.1109/WACV.2018.00163.
[34] S. Hesaraki, “Receptive Field,” Medium. Accessed: May 24, 2024. [Online]. Available: https://medium.com/@saba99/receptive-field-1726fe6ea94f
[35] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” arXiv, May 18, 2015. doi: 10.48550/arXiv.1505.04597.
[36] A. Lou, S. Guan, and M. Loew, “DC-UNet: Rethinking the U-Net Architecture with Dual Channel Efficient CNN for Medical Images Segmentation.” arXiv, May 30, 2020. doi: 10.48550/arXiv.2006.00414.
[37] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation.” arXiv, Jul. 18, 2018. doi: 10.48550/arXiv.1807.10165.
[38] Z. Gu *et al.*, “CE-Net: Context Encoder Network for 2D Medical Image Segmentation,” *IEEE Trans. Med. Imaging*, vol. 38, no. 10, pp. 2281–2292, Oct. 2019, doi: 10.1109/TMI.2019.2903562.
[39] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 10, 2015. doi: 10.48550/arXiv.1512.03385.
[40] A. Kirillov *et al.*, “Segment Anything.” arXiv, Apr. 05, 2023. doi: 10.48550/arXiv.2304.02643.

[41] E. J. Hu *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models.” arXiv, Oct. 16, 2021. doi: 10.48550/arXiv.2106.09685.
[42] C. Zhang *et al.*, “A Comprehensive Survey on Segment Anything Model for Vision and Beyond.” arXiv, May 19, 2023. doi: 10.48550/arXiv.2305.08196.
[43] J. Brownlee, “A Gentle Introduction to Cross-Entropy for Machine Learning,” MachineLearningMastery.com. Accessed: May 24, 2024. [Online]. Available: https://machinelearningmastery.com/cross-entropy-for-machine-learning/
[44] hengtao tantai, “Use weighted loss function to solve imbalanced data classification problems,” Medium. Accessed: May 24, 2024. [Online]. Available: https://medium.com/@zergtant/use-weighted-loss-function-to-solve-imbalanced-data-classification-problems-749237f38b75
[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection.” arXiv, Feb. 07, 2018. doi: 10.48550/arXiv.1708.02002.
[46] V. Rajput, “Robustness of different loss functions and their impact on network’s learning capability”.
[47] “FCN Explained | Papers With Code.” Accessed: May 24, 2024. [Online]. Available: https://paperswithcode.com/method/fcn
[48] B. Sahu, “The Evolution of Deeplab for Semantic Segmentation,” Medium. Accessed: May 24, 2024. [Online]. Available: https://towardsdatascience.com/the-evolution-of-deeplab-for-semantic-segmentation-95082b025571
[49] “What is RNN? - Recurrent Neural Networks Explained - AWS,” Amazon Web Services, Inc. Accessed: May 24, 2024. [Online]. Available: https://aws.amazon.com/what-is/recurrent-neural-network/
[50] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation.” arXiv, Mar. 08, 2015. doi: 10.48550/arXiv.1411.4038.
[51] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs.” arXiv, May 11, 2017. Accessed: May 24, 2024. [Online]. Available: http://arxiv.org/abs/1606.00915
[52] A. Salvador *et al.*, “Recurrent Neural Networks for Semantic Instance Segmentation.” arXiv, Apr. 12, 2019. doi: 10.48550/arXiv.1712.00617.
[53] M. Y. Yang, B. Rosenhahn, and V. Murino, Eds., “Copyright,” in *Multimodal Scene Understanding*, Academic Press, 2019, p. iv. doi: 10.1016/B978-0-12-817358-9.00003-2.
[54] A. Newell and J. Deng, “How Useful Is Self-Supervised Pretraining for Visual Tasks?,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 7343–7352. doi: 10.1109/CVPR42600.2020.00737.
[55] T. Brown *et al.*, “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 1877–1901. Accessed: May 23, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
[56] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, Nov. 2020, pp. 1597–1607. Accessed: May 24, 2024. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html
[57] J.-B. Grill *et al.*, “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 21271–21284. Accessed: May 24, 2024. [Online]. Available: https://papers.nips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html
[58] M. Caron *et al.*, “Emerging Properties in Self-Supervised Vision Transformers,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9630–9640. doi: 10.1109/ICCV48922.2021.00951.
[59] W.-C. Chen, C.-C. Chang, C.-Y. Lu, and C.-R. Lee, “Knowledge Distillation with Feature Maps for Image Classification.” arXiv, Dec. 03, 2018. doi: 10.48550/arXiv.1812.00660.
[60] T. Truong, S. Mohammadi, and M. Lenga, “How Transferable are Self-supervised Features in Medical Image Classification Tasks?,” in *Proceedings of Machine Learning for Health*, PMLR, Nov. 2021, pp. 54–74. Accessed: May 24, 2024. [Online]. Available:

https://proceedings.mlr.press/v158/truong21a.html
[61] Y. Guo *et al.*, "A Broader Study of Cross-Domain Few-Shot Learning," in *Computer Vision – ECCV 2020*, vol. 12372, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., in Lecture Notes in Computer Science, vol. 12372. , Cham: Springer International Publishing, 2020, pp. 124–141. doi: 10.1007/978-3-030-58583-9_8.
[62] J. Cai and S. Shen, *Cross-Domain Few-Shot Learning with Meta Fine-Tuning*. 2020.
[63] Y. Yu, S. Zuo, H. Jiang, W. Ren, T. Zhao, and C. Zhang, "Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 1063–1077. doi: 10.18653/v1/2021.naacl-main.84.
[64] I. V. Slijepcevic *et al.*, "Radio galaxy zoo: towards building the first multipurpose foundation model for radio astronomy with self-supervised learning," *RAS Tech. Instrum.*, vol. 3, no. 1, pp. 19–32, Jan. 2024, doi: 10.1093/rasti/rzad055.
[65] I. Slijepcevic, A. Scaife, M. Walmsley, and M. Bowles, *Learning useful representations for radio astronomy "in the wild" with contrastive learning*. 2022. doi: 10.48550/arXiv.2207.08666.
[66] K. Mohale and M. Lochner, "Enabling unsupervised discovery in astronomical images through self-supervised representations," *Mon. Not. R. Astron. Soc.*, vol. 530, no. 1, pp. 1274–1295, May 2024, doi: 10.1093/mnras/stae926.
[67] N. Gupta *et al.*, "Deep Learning for Morphological Identification of Extended Radio Galaxies using Weak Labels," 2023, doi: 10.48550/ARXIV.2308.05166.
[68] H. Tang, A. M. M. Scaife, and J. P. Leahy, "Transfer learning for radio galaxy classification," 2019, doi: 10.48550/ARXIV.1903.11921.
[69] J. Y.-Y. Lin, S.-M. Liao, H.-J. Huang, W.-T. Kuo, and O. H.-M. Ou, "Galaxy Morphological Classification with Efficient Vision Transformer," 2021, doi: 10.48550/ARXIV.2110.01024.
[70] "PyTorch," PyTorch. Accessed: May 24, 2024. [Online]. Available: https://pytorch.org/
[71] "MASSIVE." Accessed: May 24, 2024. [Online]. Available: https://www.massive.org.au/
[72] "fastai/fastai." fast.ai, May 24, 2024. Accessed: May 24, 2024. [Online]. Available: https://github.com/fastai/fastai
[73] "Self Supervised Learning with Fastai | self_supervised." Accessed: May 24, 2024. [Online]. Available: https://keremturgutlu.github.io/self_supervised/
[74] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 9912–9924. Accessed: May 24, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/70feb62b69f16e0238f741fab228fec2-Abstract.html
[75] "torchrun (Elastic Launch) — PyTorch 2.3 documentation." Accessed: May 24, 2024. [Online]. Available: https://pytorch.org/docs/stable/elastic/run.html
[76] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
[77] F. A. M. Porter, "MiraBest Batched Dataset", doi: 10.5281/zenodo.4288837.
[78] F. A. M. Porter, "CRUMB: the Collected Radiogalaxies Using MiraBest dataset." Accessed: Oct. 15, 2023. [Online]. Available: https://zenodo.org/records/7746094
[79] C. Wu *et al.*, "Radio Galaxy Zoo: CLARAN – a deep learning classifier for radio morphologies," *Mon. Not. R. Astron. Soc.*, vol. 482, no. 1, pp. 1211–1230, Jan. 2019, doi: 10.1093/mnras/sty2646.
[80] K.-Y. Wong, "WongKinYiu/yolov9." May 24, 2024. Accessed: May 24, 2024. [Online]. Available: https://github.com/WongKinYiu/yolov9
[81] "Figure 2. The architecture of Unet.," ResearchGate. Accessed: May 24, 2024. [Online]. Available: https://www.researchgate.net/figure/The-architecture-of-Unet_fig2_334287825
[82] K. Zhang and D. Liu, "Customized Segment Anything Model for Medical Image Segmentation." arXiv, Oct. 17, 2023. Accessed: May 24, 2024. [Online]. Available: http://arxiv.org/abs/2304.13785
[83] x-engineer.org, "Bilinear interpolation – x-engineer.org." Accessed: May 24, 2024. [Online]. Available: https://x-engineer.org/bilinear-interpolation/

[84] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
[85] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” presented at the International Conference on Learning Representations, Oct. 2020. Accessed: May 24, 2024. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy
[86] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.
[87] J. Maurício, I. Domingues, and J. Bernardino, “Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review,” *Appl. Sci.*, vol. 13, no. 9, Art. no. 9, Jan. 2023, doi: 10.3390/app13095521.
[88] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random Erasing Data Augmentation.” arXiv, Nov. 16, 2017. Accessed: May 24, 2024. [Online]. Available: http://arxiv.org/abs/1708.04896
[89] X. Hao *et al.*, “MixGen: A New Multi-Modal Data Augmentation.” arXiv, Jan. 09, 2023. Accessed: May 24, 2024. [Online]. Available: http://arxiv.org/abs/2206.08358
[90] R. Girshick, “Fast R-CNN,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.
[91] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proc. Natl. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, doi: 10.1073/pnas.1611835114.
[92] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 2921–2929. doi: 10.1109/CVPR.2016.319.
[93] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, “DN-DETR: Accelerate DETR Training by Introducing Query DeNoising”.
[94] V. V. Ramasesh, E. Dyer, and M. Raghu, “Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics.” arXiv, Jul. 14, 2020. Accessed: May 24, 2024. [Online]. Available: http://arxiv.org/abs/2007.07400
[95] Y. Huang *et al.*, “Segment Anything Model for Medical Images?,” *Med. Image Anal.*, vol. 92, p. 103061, Feb. 2024, doi: 10.1016/j.media.2023.103061.
[96] X. Zhao *et al.*, “Fast Segment Anything.” arXiv, Jun. 21, 2023. doi: 10.48550/arXiv.2306.12156.
[97] M. University and W. University, “MASSIVE.” Accessed: Oct. 15, 2023. [Online]. Available: https://www.massive.org.au/
[98] H. Tang, “FR-DEEP.” Jun. 13, 2023. Accessed: Oct. 21, 2023. [Online]. Available: https://github.com/HongmingTang060313/FR-DEEP
[99] D. CSIRO, “Artificial Intelligence for Science report.” Accessed: Oct. 15, 2023. [Online]. Available: https://www.csiro.au/en/research/technology-space/ai/artificial-intelligence-for-science-report
[100] N. United, “THE 17 GOALS | Sustainable Development.” Accessed: Oct. 15, 2023. [Online]. Available: https://sdgs.un.org/goals
[101] “Open Code,” PLOS. Accessed: Oct. 19, 2023. [Online]. Available: https://plos.org/open-science/open-code/
[102] “Open science | UNESCO.” Accessed: Oct. 19, 2023. [Online]. Available: https://www.unesco.org/en/open-science
[103] “Open science | UNESCO.” Accessed: Oct. 19, 2023. [Online]. Available: https://www.unesco.org/en/open-science
[104] “Open Access – Monash University Publishing.” Accessed: Oct. 20, 2023. [Online]. Available: https://publishing.monash.edu/books/open-access/
[105] J. Jumper *et al.*, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, vol. 596, no. 7873, Art. no. 7873, Aug. 2021, doi: 10.1038/s41586-021-03819-2.

9. Appendices

Appendix A: Additional Information

Appendix A1: Previous Work

Alternative Datasets

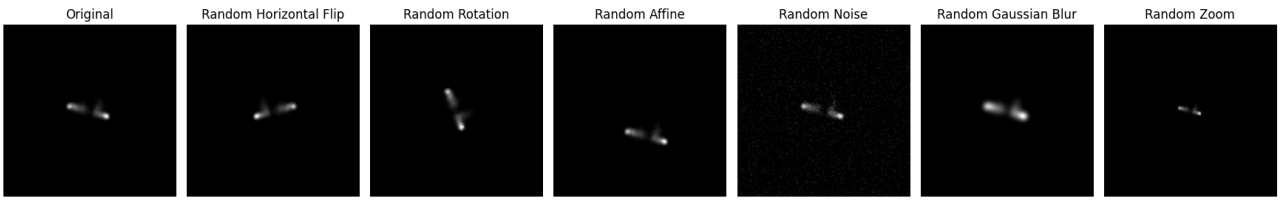
While the intention of this project was to develop a neural network with the ASKAP data provided by CSIRO's Data61, access to the RadioGalaxyNET dataset was restricted to the team for the first half of the project while the data was under review for publishing. Thus, it became necessary to use alternate, publicly available datasets which could be used as a testing bed for our designs.

While several datasets were explored, the Collected Radio galaxies Using MiraBest (CRUMB) dataset [29] was ultimately chosen for testing, as it was the largest and most accessible radio galaxy dataset available. CRUMB combines the MiraBest dataset with supplementary datasets, FR-DEEP [30] and AT17, to form one larger dataset, however, this dataset is significantly smaller than the RadioGalaxyNET dataset. These images are single channel (grayscale), as opposed to the CSIRO's three channel data (RGB), and much lower in quality. Additionally, these images were categorised into three classes (FR-I, FR-II, and FR-X) as opposed to the four provided in the RadioGalaxyNET set (FR-I, FR-II, FR-X, and R).

Furthermore, unlike the more detailed RadioGalaxyNET images, the CRUMB images only have a single galaxy per image, located in the centre. This meant that creating a detection or segmentation model using these images would not be worthwhile, as a detection model would simply generate a bounding box in the centre each time, and a segmentation model would only learn to select any white pixels i.e. the results of such neural networks would be trivial. Additionally, the CRUMB dataset did not contain bounding boxes nor segmentation masks, thus a conventional detection or segmentation model could not be trained on this dataset. Therefore, it was decided that these images would be used to create a simple classifier. While a classifier wasn't the main goal of the project, it was still a very important stepping stone to a full segmentation model.

The CRUMB images were transformed in accordance with the recommended procedure [29]. Images were processed to have pixel intensities within a standardized range, guided by a specific mean and standard deviation. Further transformations included a center crop of size 150x150 and a random rotation. An issue with this dataset was a low number of FR-X images. This imbalance in the dataset led to the developed classifiers to rarely classifying images as FR-X, being correct even more rarely. One potential solution to this involved artificially increasing the proportion of FR-X images in the training set via a series of augmentation techniques. Each FR-X image within the training set was duplicated three times and each copy augmented through this procedure, thus bolstering the pool of FR-X images four-fold.

Data augmentation techniques were also explored using the CRUMB dataset. The images were subjected to randomized rotations within a defined range, typically between -45° to 45°, aiming to simulate various potential observational orientations. In addition to this, the images were rescaled within a specific range, typically between 0.9x to 1.1x. This rescaling introduced notable variability in the observed size and structural features of the radio galaxies. Finally, to further enhance the diversity and robustness of the dataset, the images were mirrored horizontally. This horizontal flipping ensured that the model would become adept at recognizing radio galaxy structures from different orientations.



Appendix A Figure 1 - Visualization of the different transformations applied to the augmented dataset.

While these augmentations were beneficial, some techniques were deliberately avoided to maintain the data's integrity and relevance. Random cropping was eschewed as it risked removing essential features and valuable information from the images. Similarly, excessive color augmentations were not considered due to the unique nature of radio emissions, which would render such changes irrelevant. Finally, noise injection was avoided since radio telescopic data inherently contains noise, and artificially introducing more might have further obscured the clarity and distinction of morphological features.

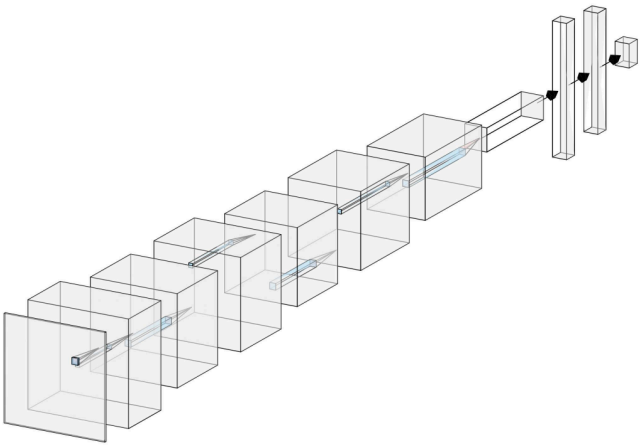
There were three main concepts for potential classifiers, a pretrained Vision Transformer (ViT), a pretrained Convolutional Neural Network (ResNet), and a custom Convolutional Neural Network (CNN) designed and trained from scratch.

Custom CNN

The original purpose of the Custom CNN was to serve as a proof of concept and a testing ground for different dataset augmentations, however, as time went on - the model continued to perform incredibly well. The initial architecture for the CNN was based partially upon the recommended structure given in the MiraBest test document [31].

This Custom CNN went through many iterations, the final architecture shown in FIgure 17.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 150, 150]             160
              ReLU-2         [-1, 16, 150, 150]               0
         MaxPool2d-3           [-1, 16, 75, 75]               0
            Conv2d-4           [-1, 32, 75, 75]           4,640
              ReLU-5           [-1, 32, 75, 75]               0
         MaxPool2d-6           [-1, 32, 37, 37]               0
            Conv2d-7           [-1, 64, 37, 37]          18,496
              ReLU-8           [-1, 64, 37, 37]               0
         MaxPool2d-9           [-1, 64, 18, 18]               0
           Conv2d-10          [-1, 128, 18, 18]          73,856
             ReLU-11          [-1, 128, 18, 18]               0
           Conv2d-12          [-1, 128, 18, 18]         147,584
             ReLU-13          [-1, 128, 18, 18]               0
AdaptiveAvgPool2d-14            [-1, 128, 3, 3]               0
          Sigmoid-15            [-1, 128, 3, 3]               0
           Linear-16                  [-1, 512]         590,336
           Linear-17                  [-1, 256]         131,328
           Linear-18                    [-1, 3]             771
================================================================
Total params: 967,171
Trainable params: 967,171
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.09
Forward/backward pass size (MB): 12.04
Params size (MB): 3.69
Estimated Total Size (MB): 15.82
----------------------------------------------------------------
```



Appendix A Figure 2 - Custom CNN Architecture Version 2.0: Five Convolutional Layers, Three Linear Layers.

This architecture has five convolutional layers and three fully connected linear layers. The activation layers are all ReLU, except for the final sigmoid function. This final model was fine tuned and in order to account

for the imbalanced dataset, the weights of the optimiser were also adjusted. These weights are in the form [FR-I, FR-II, FR-X], with a higher weight corresponding to a greater significance.

Appendix A Table 1: Iterative hyperparameter tuning process of Custom CNN Version 2.0

Recorded Iteration Number	**Hyperparameters**	**Artificially Enhanced FR-X**	**Accuracy**	**FR-X Predictions (correctly predicted /predicted)**
11	Learning Rate = 1e-4 LR Decay Step = 400 LR Decay Gamma = 0.5 Weight Decay = 5e-5 Epochs = 1000 Early Stopping = 75 Weights = [1, 1, 1.6]	No	87.6%	3/5
13	Learning Rate = 1e-4 LR Decay Step = 250 LR Decay Gamma = 0.5 Weight Decay = 5e-5 Epochs = 1000 Early Stopping = 125 Weights = [1, 0.68, 1.55]	No	88.6%	0/0
15	Learning Rate = 3e-4 LR Decay Step = 400 LR Decay Gamma = 0.5 Weight Decay = 5e-5 Epochs = 1000 Early Stopping = 125	Yes	86.1%	7/16

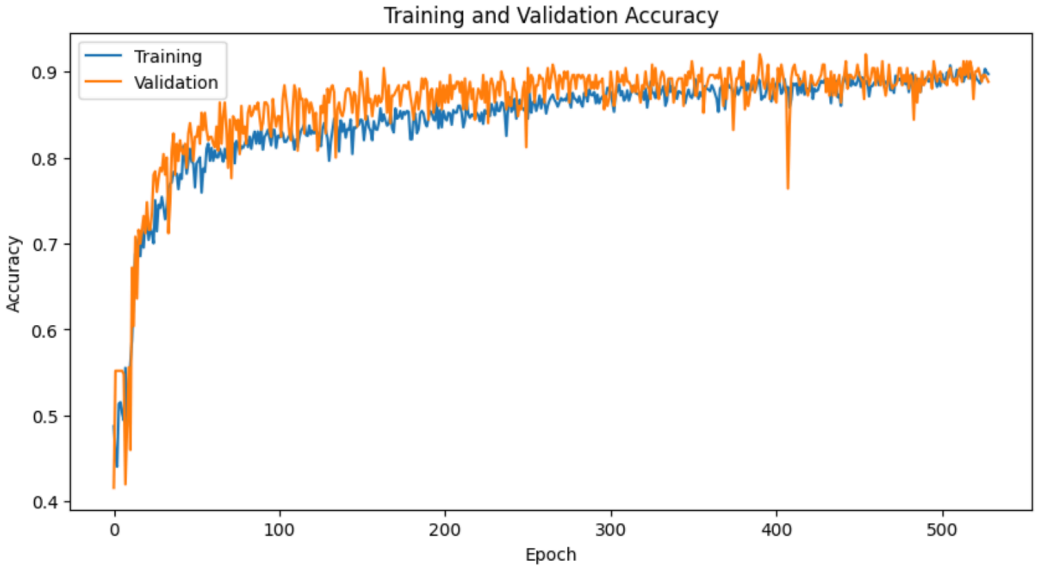
	Weights = [1, 1, 1.25]			

The most important models and their hyperparameters are recorded in the table above. The prediction of FR-X galaxies proved to be quite difficult, thus the performance of each model on that class was of particular significance.

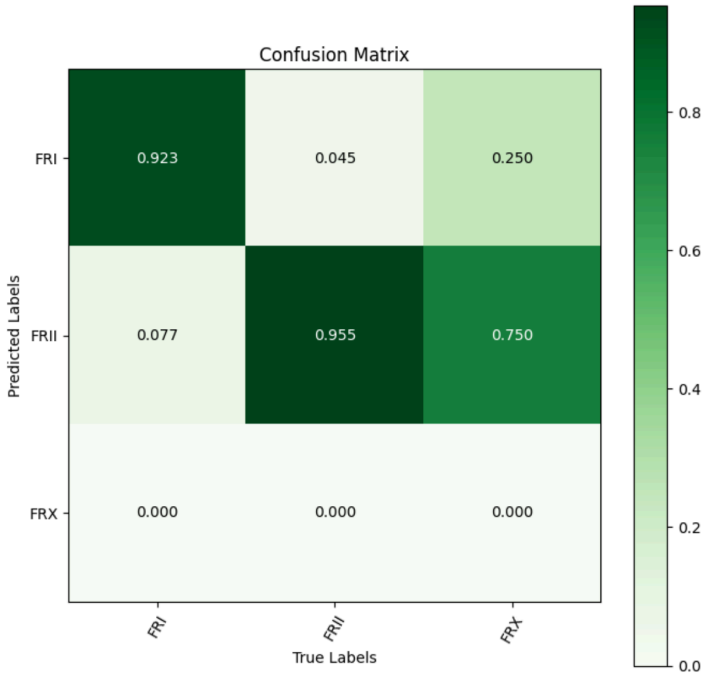
Model #11 had the highest ratio of correctly predicted FR-X galaxies to total times FR-X was predicted.

Model #13 had the highest total accuracy of 88.6%. The training and validation curves, as well as the confusion matrix can be seen for model #13.

Appendix A Figure 3 – Training and Validation Loss for Model #13: Highest Overall Accuracy Case.



Appendix A Figure 4– Training and Validation Accuracy for Model #13: Highest Overall Accuracy Case.

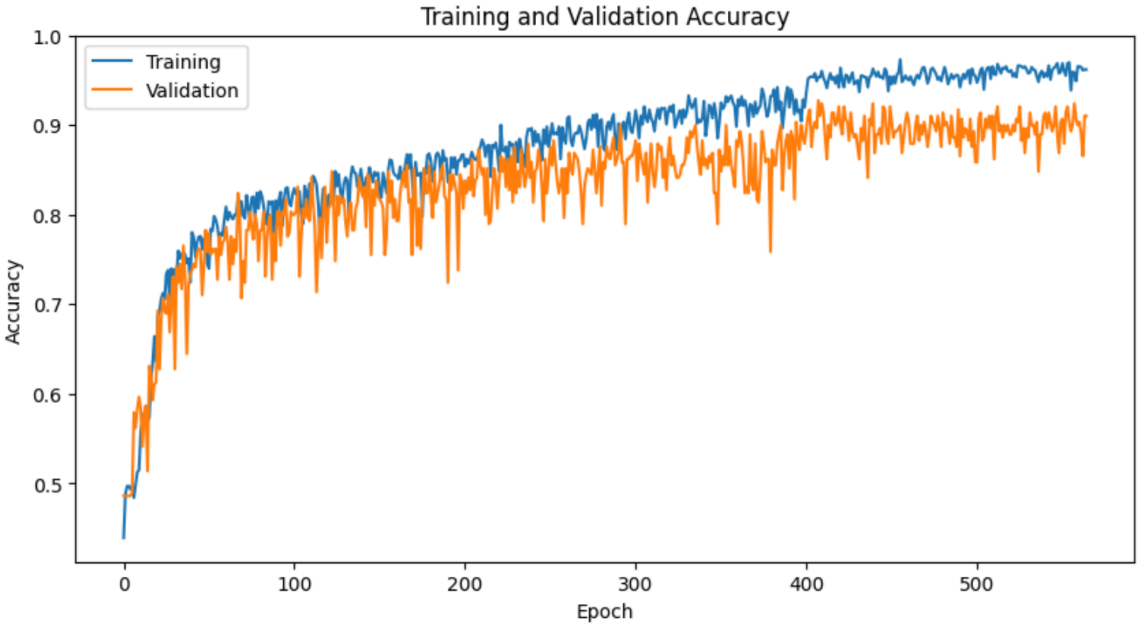


Appendix A Figure 5 – Confusion Matrix for Model #13: Highest Overall Accuracy Case.

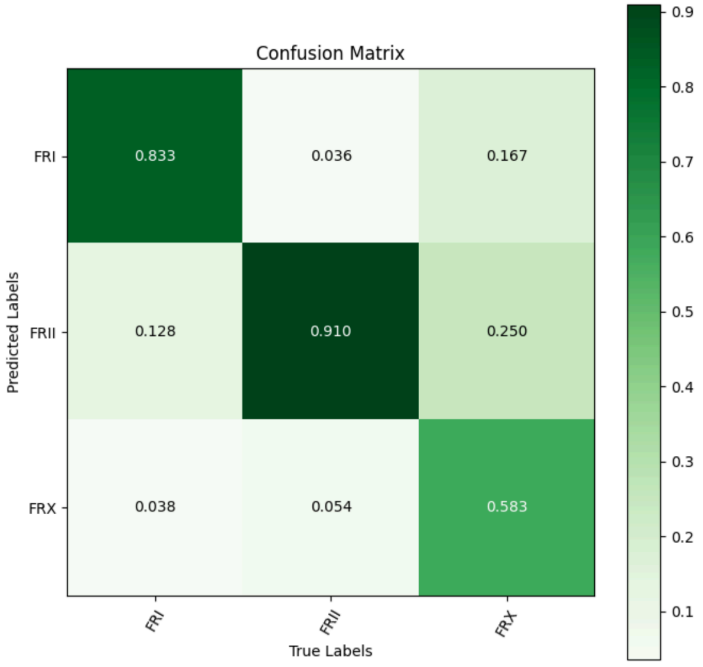
While the accuracies for FR-I and FR-II images are exceptional (92.3% and 95.5%, respectively), the prediction accuracy for the FR-X galaxies is 0%.

To account for this lack of FR-X predictions, model #15 was trained with the augmented FR-X data which essentially multiplied the instances of FR-X images by four. The training graphs and confusion matrix for model #15 can be seen below.

Appendix A Figure 6 – Training and Validation Loss for Model #15: Highest FR-X Accuracy Case.



Appendix A Figure 7 – Training and Validation Accuracy for Model #15: Highest FR-X Accuracy Case.



Appendix A Figure 8 – Confusion Matrix for Model #15: Highest FR-X Accuracy Case.

Here, it can be seen that using augmented data had the desired effect. It increased the FR-X accuracy from 0% to 58.3%, a massive improvement, however, this had two side effects. Firstly, it decreased the accuracy of the other two classes to 83.3% and 91.0% for FR-I and FR-II, respectively. This is not unexpected, as the previous model was essentially acting as a two class classifier. The second effect is that the model's FR-X predictions are not as accurate as they would first seem. While it is true that the model is predicting the FR-X class correctly more often, it can be seen that out of the 16 times the model predicted an image to be an FR-X, it was only truly correct 7 times, which is less than 50% accuracy.

Ultimately, while an improvement was made with the FR-X predictions, the CRUMB dataset was simply too shallow. Further duplications of the training data could have been made, however, it likely would not have made a difference to the model as there was not enough variety in the training images. If further improvements to the classifier were to be made, an entirely new dataset would need to be used.

While this Custom CNN served as a proof of concept that classification of radio galaxies could be performed, classification was not the final goal of the project. Unfortunately, this model lacked the complexity needed to be used as a detection or segmentation model. Therefore, further work on this model was not required.

Fine-Tuned ViT

Another classifier model explored was a fine-tuned Vision Transformer (ViT). The ViT architecture has emerged in recent times as an alternative to traditional CNNs. ViTs harness the power and flexibility of transformers to process images. Specifically, the model variant under consideration is the ViT-B/16, a pre-trained model that divides input images into fixed-size, non-overlapping patches of 16x16 pixels. Each patch is linearly embedded, forming a sequence of tokens that are processed by the transformer's architecture. These tokens pass through an assortment of self-attention layers and feed-forward networks, culminating with a pass through the classifier head of the transformer, resulting in the assignment of a predicted label to the image.

With transformers demonstrating promise in both the computer vision and natural language processing domains, they have opened up exploration beyond traditional CNNs. One significant advantage is the reduced computational resources required. For these reasons, the ViT model will be explored.

This model was trained on ImageNet-21k (14 million images, 21,843 classes) at resolution 224x224, and fine-tuned on ImageNet 2012 (1 million images, 1,000 classes) at resolution 224x224. With its foundational knowledge derived from an extensive dataset, the model possesses a vast bank of visual features. When this pre-trained architecture is adapted to the catalog of radio galaxy datasets, it requires reduced training durations, converges quicker, and possesses the capability to effectively extract and determine features in generic images as well as specific to radio galaxy morphologies.

Using a pre-trained model as a starting point addresses some potential training challenges. One primary concern when dealing with intricate architectures like the ViT, especially when data is limited, is overfitting. By building upon a model already equipped with generalized features, the risk of over-relying on the specificities of the limited dataset is reduced, promoting better generalization.

```
==========================================================================================================================
Layer (type (var_name))                                 Input Shape          Output Shape         Param #
==========================================================================================================================
VisionTransformer (VisionTransformer)                   [32, 3, 224, 224]    [32, 2]              768
├─Conv2d (conv_proj)                                    [32, 3, 224, 224]    [32, 768, 14, 14]    (590,592)
├─Encoder (encoder)                                     [32, 197, 768]       [32, 197, 768]       151,296
│    └─Dropout (dropout)                                [32, 197, 768]       [32, 197, 768]       --
│    └─Sequential (layers)                              [32, 197, 768]       [32, 197, 768]       --
│    │    └─EncoderBlock (encoder_layer_0)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_1)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_2)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_3)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_4)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_5)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_6)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_7)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_8)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_9)              [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_10)             [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    │    └─EncoderBlock (encoder_layer_11)             [32, 197, 768]       [32, 197, 768]       (7,087,872)
│    └─LayerNorm (ln)                                   [32, 197, 768]       [32, 197, 768]       (1,536)
├─ClassifierHead (heads)                                [32, 768]            [32, 2]              --
│    └─BatchNorm1d (bn)                                 [32, 768]            [32, 768]            1,536
│    └─Dropout (dropout)                                [32, 768]            [32, 768]            --
│    └─Linear (linear)                                  [32, 768]            [32, 2]              1,538
==========================================================================================================================
Total params: 85,801,730
Trainable params: 3,074
Non-trainable params: 85,798,656
Total mult-adds (G): 5.52
==========================================================================================================================
Input size (MB): 19.27
Forward/backward pass size (MB): 3330.93
Params size (MB): 229.21
Estimated Total Size (MB): 3579.41
==========================================================================================================================
```

Appendix A Figure 9 – Architecture details of VisionTransformer model.

Num	Layer type	Output Shape	Parameters
0	Input	(32, 3, 224, 224)	0
1	Conv2d	(32, 768, 14, 14)	590,592
2	Dropout	(32, 197, 768)	0
3-14	EncoderBlock (x12)	(32, 197, 768)	85,054,464
15	LayerNorm	(32, 197, 768)	1,536
16	BatchNorm1d	(32, 768)	1,536
17	Dropout	(32, 768)	0
18	Linear	(32, 3)	2,307

Appendix A Figure 10 - Summary Architecture details of VisionTransformer model

Appendix Table 3: Outlining the details of the encoder block in the ViT

Layer Name	Sub-layer	Layer Type	Configuration
2*ln_1	-	LayerNorm	(768,), eps=1e-06
2*self_attention	out_proj	NonDynamicallyQuantizableLinear	in=768, out=768
dropout	-	Dropout	p=0.0
2*ln_2	-	LayerNorm	(768,), eps=1e-06
5*mlp	0	Linear	in=768, out=3072
	1	GELU	approximate='none'
	2	Dropout	p=0.0
	3	Linear	in=3072, out=768
	4	Dropout	p=0.0

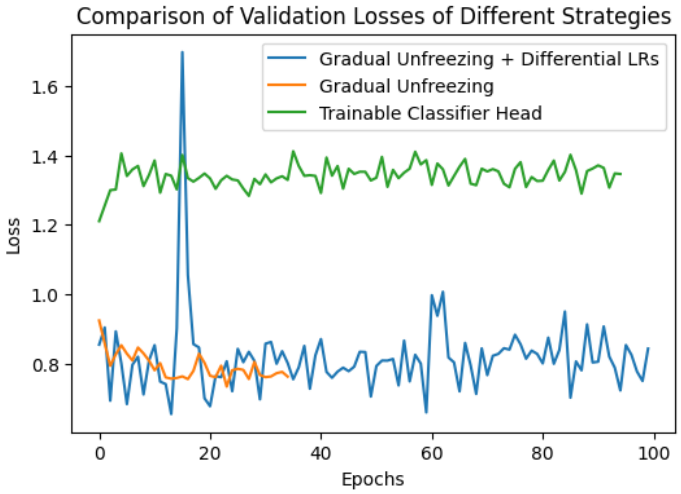
The model was first trained to distinguish between the two well-defined classes, FR-I and FR-II, via binary classification. Once the model could reliably do this, it was trained to distinguish between three classes: FR-I, FR-II, and FR-X. The model was trained in this sequential way to make sure that it could learn the clear-cut distinctions before moving on to the more subtle features of the FR-X class.

An integral component of the fine-tuning process was the concept of gradual unfreezing, implemented to counteract the phenomenon known as catastrophic forgetting [32]. The challenge of catastrophic forgetting surfaces when a model, in its pursuit to grasp new patterns, unintentionally relinquishes its prior learnings. Introducing gradual unfreezing served as a remedy, enabling the model to initially adjust its top-tier representations to the task. Complementing the gradual unfreezing was the implementation of differential learning rates. Here, each of the 12 encoder layers were adjusted using a distinct learning rate, corresponding to its depth and importance. Conversely, layers nearer to the model's output, which were more attuned to task-specific nuances, experienced more robust updates. This methodology inherently fostered a harmonious equilibrium, facilitating the pre-trained model's smooth adaptation to the distinctive characteristics of the radio galaxy dataset.

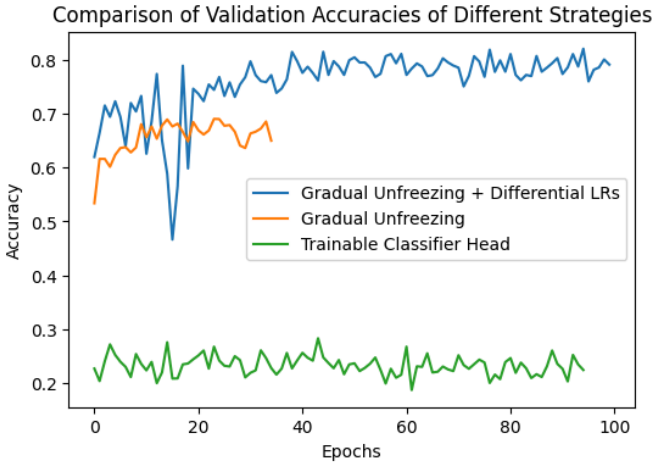
```
grouped_parameters = [
     {'params': layer_groups[0][0].parameters(), 'lr': 1e-6},
     {'params': layer_groups[1][0].parameters(), 'lr': 1e-5},
     {'params': layer_groups[2][0].parameters(), 'lr': 1e-4},
     {'params': layer_groups[3][0].parameters(), 'lr': 1e-4},
     {'params': layer_groups[4][0].parameters(), 'lr': 1e-4},
     {'params': layer_groups[5][0].parameters(), 'lr': 1e-4},
     {'params': layer_groups[6][0].parameters(), 'lr': 1e-3},
     {'params': layer_groups[7][0].parameters(), 'lr': 1e-3},
]
```

Appendix A Figure 11 - Differential Learning Rates for the layers appearing at different depths of the model.

These strategic interventions—gradual unfreezing and differential learning rates—yielded noticeable improvements in the model's performance, ensuring stability in learning and enhanced feature extraction specific to radio galaxy morphologies.



Appendix A Figure 12 - Validation losses for each of the three scenarios (Early Stopping was used): 1) Fine-tuning the classifier and freezing all other layers, 2) Using Gradual Unfreezing, 3) Employing Differential Learning Rates in addition to Gradual Unfreezing.



Appendix A Figure 13 - Validation accuracies for each of the three scenarios (Early Stopping was used): 1) Fine-tuning the classifier and freezing all other layers, 2) Using Gradual Unfreezing, 3) Employing Differential Learning Rates in addition to Gradual Unfreezing.

In the end, this classifier achieved results slightly below that of the custom CNN, with a validation accuracy of just over 80%. This model suffers from the same issue as the custom CNN, in that the quality and number of the images within the CRUMB dataset severely restrict the model's ability to learn complex patterns. Therefore, this model could not be improved until the ASKAP data became available.

While ViT segmentation models do exist, they tend to struggle with the detection of small objects, a trend that can be observed when comparing this model's accuracy to the Custom CNN model. The ViT model itself splits the image into 16 patches and with images in the datasets being gray-scale and having large areas of empty pixels, some of these patches would contain minimal to no galaxy data. These unnecessary patches mean the model will be spending large amounts of time on redundant computation, which will increase model computational cost and latency. Therefore, it was ultimately decided not to pursue the use of ViT models further.

Class Activation Map and Prior Investigations

1. What is CAM?

A Class Activation Map (CAM) is a visualization technique in deep learning that highlights regions in an image crucial for a convolutional neural network's (CNN) class prediction. By overlaying the CAM on the original image, one can see the areas where the network focuses, enhancing model interpretability. CAMs are generated by weighting the output feature maps of a CNN's last convolutional layer with the class-specific weights from the fully connected layer, aiding in object localization without extra annotations. The core function responsible for the generation of the class activation map is shown here:

$$M_C(x,y) = \sum_k w^c{}_k f_k(x,y)$$

$w^c{}_k$: Represents the weight of the kth feature map for class c, obtained from the fully connected layers of the Convolution Neural Network.

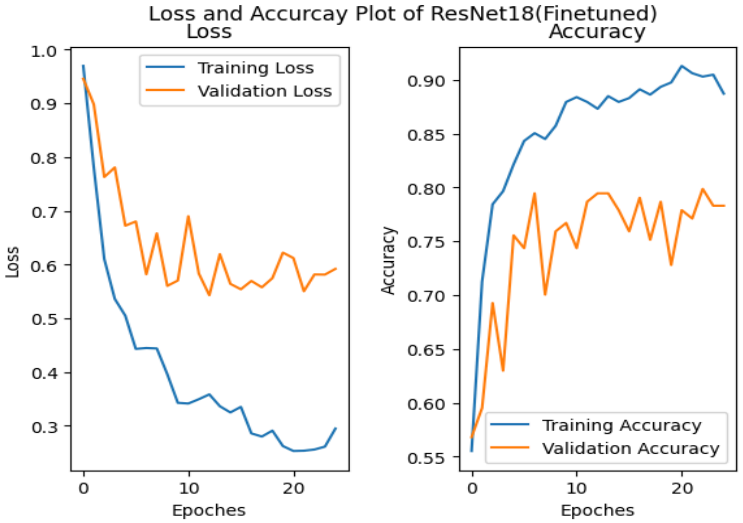
f_k: Denotes the k-th feature map from the last convolution layer of the network.

2. CAM for self-supervised learning

CAMs can be used to train a supervised segmentation model. These CAMs visualise the most salient parts of images useful for distinguishing radio galaxies in images with black backgrounds. CAMs can be used in conjunction with a variety of other methods. For example, saliency maps have been used to highlight the most salient or distinguishable regions of an image and can provide accurate object boundaries that are not present in CAMs. Explicit Pseudo-pixel Supervision (EPS) refines a classifier by predicting saliency maps from its CAMs. This technique is known to produce high quality masks with precise boundaries that can distinguish co-occurring pixels without further refinement. Other ways include ReCAM, where it trains a secondary classifier using Softmax Cross-Entropy (SEC) that is plug-and-play for any CAM variant. These methods can be adopted to begin to generate masks for the model.

To explore the extraction of Class Activation Maps (CAM) from image classification models, a ResNet18 model pre-trained on the ImageNet dataset is used as the backbone of the CAM extraction. The purpose of a CAM is to discover which parts of the image are driving the decision making of the model, also known as a saliency map. These saliency maps can be turned into the pseudo-masks used for training segmentation models, especially for self-supervised machine-learning tasks.

The model is trained on the CRUMB dataset using ADAM optimizer with a learning rate of 0.0001 for 25 epochs. Ultimately, a ResNet18 model with a 87% for the training accuracy and 77% validation accuracy is created for the extraction of CAM.

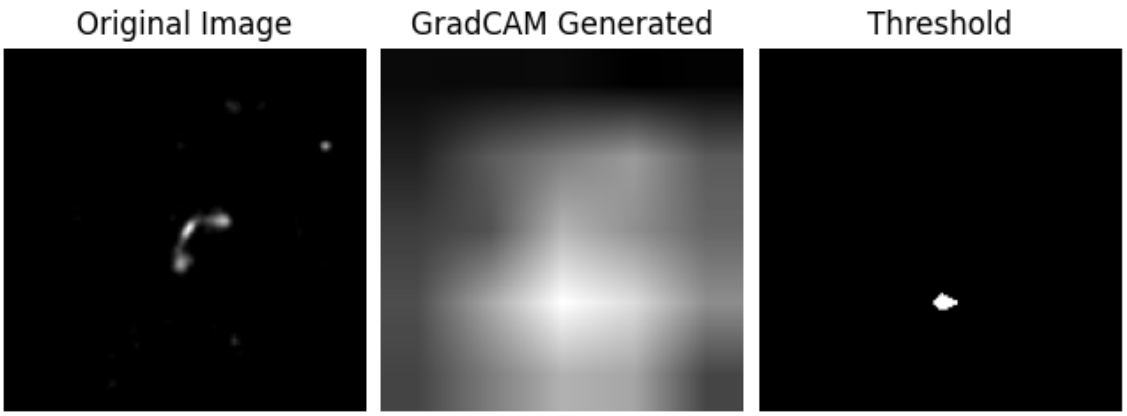


Appendix A Figure 14 – Loss and Accuracy of ResNet18 Model.

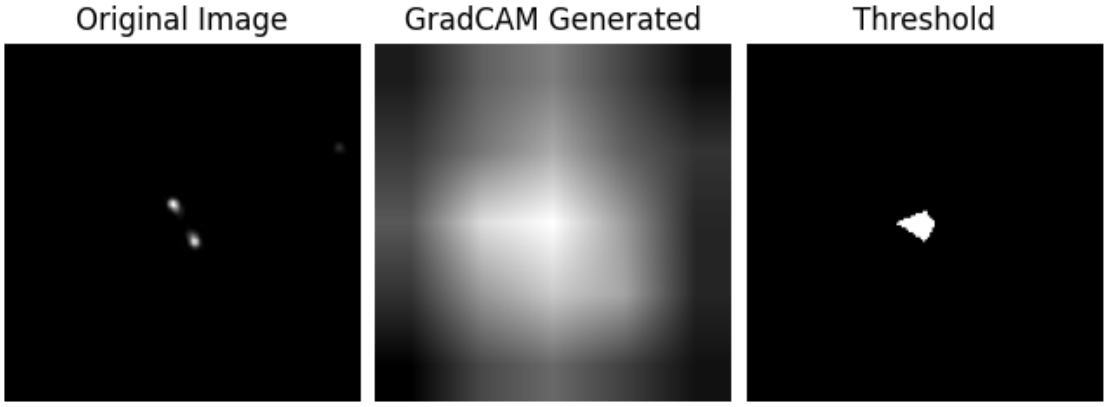
It was possible to convert the extracted Grad-CAM into pseudo-masks through thresholding:

$$Seg[i,j] = 1, \forall CAM[i,j] > I_{th}, 0 < I_{th} < 1$$

Two examples extractions are shown below:



Appendix A Figure 15 – FR-I raw data and CAM.



Appendix A Figure 16 – FR-II raw data and CAM.

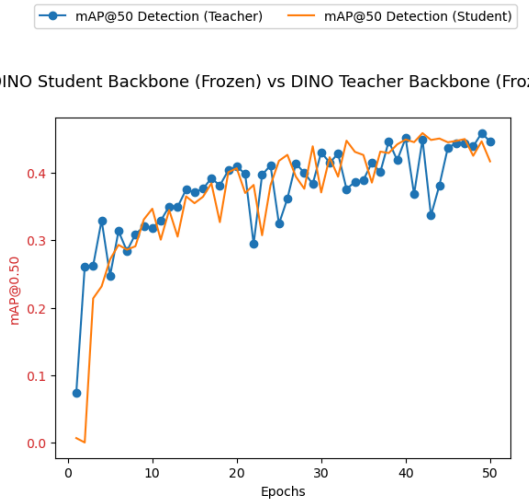
As observed from the example extractions, the pseudo masks generated can be used to identify the general locations of the FR-I and FR-II galaxies, but were unable to be used to extract any distinguishing features that outlines the object (radio galaxies) from the original image. The pseudo-masks generated through thresholding are therefore not very useful for training a segmentation model. Once again, this issue stems from the quality and number of images in the CRUMB dataset.

Appendix A2: Additional Data

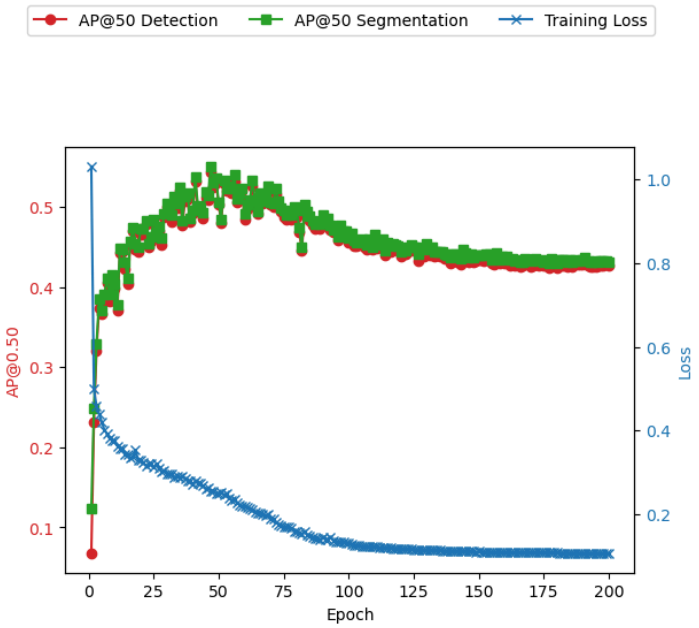
Appendix A Table 4: Training details of the yolov9-c detection model.

Iteration	*Hyperparameters*	*mAP50*
1	patience: 200, batch_size: 8, optimiser: SGD, lr0: 0.01, lrf: 0.01, momentum: 0.937, weight_decay: 0.0005, warmup_epochs: 3.0, warmup_momentum: 0.8, warmup_bias_lr: 0.1, box: 7.5, cls: 0.5, cls_pw: 1.0, obj: 0.7, obj_pw: 1.0, dfl: 1.5, iou_t: 0.2, anchor_t: 5.0, fl_gamma: 0.0, hsv_h: 0.015, hsv_s: 0.7, hsv_v: 0.4, degrees: 0.0, translate: 0.1, scale: 0.9, shear: 0.0, perspective: 0.0, flipud: 0.0, fliplr: 0.5, mosaic: 1.0, mixup: 0.15, copy_paste: 0.3	0.783
2	patience: 200, batch_size: 4, optimiser: SGD, lr0: 0.01, lrf: 0.01, momentum: 0.937, weight_decay: 0.0005, warmup_epochs: 3.0, warmup_momentum: 0.8, warmup_bias_lr: 0.1, box: 7.5, cls: 0.5, cls_pw: 1.0, obj: 0.7, obj_pw: 1.0, dfl: 1.5, iou_t: 0.2, anchor_t: 5.0, fl_gamma: 0.0, hsv_h: 0.015, hsv_s: 0.7, hsv_v: 0.4, degrees: 0.0, translate: 0.1, scale: 0.9, shear: 0.0, perspective: 0.0, flipud: 0.0, fliplr: 0.5, mosaic: 1.0, mixup: 0.15, copy_paste: 0.3	0.783
3	patience: 200, batch_size: 8, optimiser: ADAM, lr0: 0.001, lrf: 0.01, momentum: 0.937, weight_decay: 0.0005, warmup_epochs: 3.0, warmup_momentum: 0.8, warmup_bias_lr: 0.1, box: 7.5, cls: 0.5, cls_pw: 1.0, obj: 0.7, obj_pw: 1.0, dfl: 1.5, iou_t: 0.2, anchor_t: 5.0, fl_gamma: 0.0, hsv_h: 0.015, hsv_s: 0.7, hsv_v: 0.4, degrees: 0.0, translate: 0.1, scale: 0.9, shear: 0.0, perspective: 0.0, flipud: 0.0, fliplr: 0.5, mosaic: 1.0, mixup: 0.15, copy_paste: 0.3	0.806
4	patience: 200, batch_size: 8, optimiser: SGD, lr0: 0.01, lrf: 0.01, momentum: 0.937, weight_decay: 0.0005, warmup_epochs: 3.0, warmup_momentum: 0.8, warmup_bias_lr: 0.1, box: 7.5, cls: 0.5, cls_pw: 1.0, obj: 0.7, obj_pw: 1.0,	0.733

	dfl: 1.5, iou_t: 0.2, anchor_t: 5.0, fl_gamma: 0.0, hsv_h: 0.015, hsv_s: 0.7, hsv_v: 0.4, degrees: 90, translate: 0.1, scale: 0.9, shear: 0.0, perspective: 0.0, flipud: 0.5, fliplr: 0.5, mosaic: 1.0, mixup: 0.15, copy_paste: 0.3	
5	patience: 200, batch_size: 8, optimiser: SGD, lr0: 0.00816, lrf: 0.01042, momentum: 0.94361, weight_decay: 0.00041, warmup_epochs: 3.21425, warmup_momentum: 0.78359, warmup_bias_lr: 0.1, box: 6.6458, cls: 0.4862, cls_pw: 1.0, obj: 0.7, obj_pw: 1.0, dfl: 1.58827, iou_t: 0.2, anchor_t: 5.0, fl_gamma: 0.0, hsv_h: 0.01368, hsv_s: 0.68502, hsv_v: 0.37691, degrees: 0, translate: 0.09005, scale: 0.52978, shear: 0.0, perspective: 0.0, flipud: 0, fliplr: 0.53474, mosaic: 1.0, mixup: 0, copy_paste: 0	0.817
6	patience: 200, batch_size: 8, optimiser: SGD, lr0: 0.00756, lrf: 0.01015, momentum: 0.93531, weight_decay: 0.00033, warmup_epochs: 2.74096, warmup_momentum: 0.93923, warmup_bias_lr: 0.1, box: 7.896, cls: 0.44753, cls_pw: 1.0, obj: 0.7, obj_pw: 1.0, dfl: 1.29197, iou_t: 0.2, anchor_t: 5.0, fl_gamma: 0.0, hsv_h: 0.00933, hsv_s: 0.54556, hsv_v: 0.2921, degrees: 0, translate: 0.09114, scale: 0.41609, shear: 0.0, perspective: 0.0, flipud: 0, fliplr: 0.25366, mosaic: 0.97214, mixup: 0, copy_paste: 0	0.794



Appendix A Figure 17 – DINO Student Backbone vs. DINO Teacher Backbone.



Appendix A Figure 18 – Barebone Faster RCNN 200 Epochs.

Appendix B: Project Risk Assessment

9 NUMBER	RISK DESCRIPTION	TREND	CURRENT	RESIDUAL
51605	FYP Risk Assessment Plan, Project: AI Driven Exploration of Galaxies		Medium	Medium

RISK TYPE
1. Activity or Task Based Risk Assessment

DOCUMENTS REFERENCED
Monash Ergonomics: https://www.monash.edu/ohs/info-docs/ergonomics Visual Fatigue: https://www.comcare.gov.au/about/forms-pubs/docs/pubs/safety/hsr-helper-sore-eyes.pdf Work Stress: https://www.worksafe.vic.gov.au/work-related-stress https://www.monash.edu/staff-health-wellbeing/mind/mental-health/managing-stress

RISK OWNER	RISK IDENTIFIED ON	LAST REVIEWED ON	NEXT SCHEDULED REVIEW
YIDE TAO	10/08/2023	25/08/2023	25/08/2026

RISK FACTOR(S)	EXISTING CONTROL(S)	PROPOSED CONTROL(S)	OWNER	DUE DATE
Fire and Electric shock as result of lose wire. The project will involve using high power GPUs both inside and outside of Monash University. Appropriate protocols should be established to protect the participants of the project as well as those working in the same environment from being	**Control:** - Monash OHS will be responsible for checking loose wire and functioning power sockets periodically to ensure the electrical appliances on campus are functioning as normal.	**Control:** Students should actively report loose or worn wire to OHS through SARAH and put visible indications of the potential hazards in the work environment to prevent further escalation of hazard, students should be aware of all equipment in the workspace even if the stationed equipment is not part of	YIDE TAO	17/08/2023
potentially harmed by these electrical appliances e.g. being electrocuted by loose wire, loose wire starting a fire	**Control Effectiveness:** **Control:** - Students are told to check for the state of electrical appliances before usage, e.g. check for damaged sockets, loose wire etc. **Control Effectiveness:** **Control:** Fire extinguishers are provided around the University. **Control Effectiveness:**	the student's project (e.g. someone else's equipment). **Control:** Students should read the documentations of the hardware they have access to before conducting any experimentation using the high power GPUs. **Control:** Lab induction should clearly indicate the location of fire extinguishers to students. Students should also be inducted on how to appropriately use the fire extinguishing equipment.	YIDE TAO YIDE TAO	17/08/2023 17/08/2023
Ergonomic issue due to sitting in a fixed position for long period of time. The project involves sitting in front of the desktop for a long period of time monitoring and testing machine learning models developed. This action can impose short-term and long-term health implications on the participants as outlined in https://www.monash.edu/ohs/info-docs/ergonomics.	**Control:** Administratively the students are asked to learn about how to relax and exercise themselves after long period of time, however this is not compulsory. **Control Effectiveness:**	**Control:** Implement adjustable chairs and tables which can are more human-centric. These equipment should be more suitable for the general human dynamics, reduce user fatigue and can support a range of user height and weight., **Control:** Students in the team can supervise each other to ensure they are moving around and exercising after a period of time.	YIDE TAO YIDE TAO	22/05/2024 30/08/2023
Visual fatigue and eye strain due to staring at screen for long period of time. The project involves writing extensive code, which requires participants to stare at the screen for a long period of time. This may result in sore eyes and poor vision as outlined in:	**Control:** Monash recommend students look away from screen after a period of time to rest their eyes. Australian Government's department of Comcare recommend managers to impose 20-20-20 rule on his/her employees, they also recommend			

https://www.comcare.gov.au/about/forms-pubs/docs/pubs/safety/hsr-helper-sore-eyes.pdf.	implementing appropriate lighting in the work environment. **Control Effectiveness:** **Control:** It is recommended that the participants is to use glasses suitable to their sight and carry eye-drops with them for the work **Control Effectiveness:**			
Students are often required to relocate while carrying multiple items such as laptops, headphone and begs, which can often be quite heavy and distracting. During their relocation, students is susceptible to the hazard of being tripped by wires or step on moving parts.	**Control:** Try to practice engineering with professionalism, try to ensure a tidy and safe work environment that does not have parts or equipment being scattered across the floor surface. **Control Effectiveness:** **Control:** Participants of the project can reduce the risk by moving one objects at a time and try not to look at the monitor screen while relocating. **Control Effectiveness:**			
Students may experience stressful events such as files being lost, result fail to be reproducible and computer hardware failing. These type of disasters can lead to huge amount of the project participants' time being wasted, which can impose	**Control:** Establish a GitHub or other version control methodologies to prevent large amount of work being lost due to human error. **Control Effectiveness:**	**Control:** Students should also update their computer often and ensure their hardware are well-maintained (PC is functioning) before conducting the project.	YIDE TAO	18/08/2023

psychological and mental stress of the participants.	**Control:** Establish a cloud backup e.g. Google Cloud, One drive of the work done. As indicated by the FYP induction lectures, students should fully utilise the resources provided by Monash to their fullest advantage and actively backup their work to cloud storage to ensure retainment of information even with hardware failure. **Control Effectiveness:**			
The project is expected to involve large amount of computations which can result in heat from multiple electrical components (CPU, GPU) being generated. Students should follow appropriate protocols to ensure the hardware are cooled and maintained to the standard outlined in the product information. The students should also ensure their BIOS software are up to date.	**Control:** Most of the pre-built computational systems will have onboard fans, heat sinks or even water coolers to ensure the system is well-cooled and will not pose as a heat hazard. **Control Effectiveness:**	**Control:** Participants of the project should follow the system updates and regularly check the BIOS and hardware drivers are up to date. Doing so will ensure that software malfunctioning will not cause hardware failure.	YIDE TAO	18/08/2023
Research professionalism and treating each members of the team with respect is one of the most important bottom-line of the collaboration. Since, the team involves members from a diverse demography, therefore respecting each member's value and culture as well as communicating with maturity	**Control:** Monash requires all undergraduate students to complete a module on diversity and respect before enrolling in the University. **Control Effectiveness:**	**Control:** Each members of the team should focus mainly on the task at hand and keep potentially sensitive and triggering discussions at minimum. Each members should also actively present their distaste to the group when the topics of discussion has become uncomfortable for them.	YIDE TAO	19/08/2023

and respect should be emphasised greatly.				
This project is expected to have high workload, it may even be hard to manage when there are other events the students have to tend to (such as Student Teams, work and family). Without proper time management plans and fair distribution of work loads, this may cause excessive stress and fatigue on certain individual.	**Control:** By setting out an appropriate team management plan and project proposal, the team will ensure the workload of the project has been faidy distributed. Therefore reducing the potential stress imposed on individuals in the team. **Control Effectiveness:** **Control:** By participating in activities such as meditation and attending stress management seminars, the students are expected to reduce the harmful effects of stress. **Control Effectiveness:** **Control:** When the project has becoming hugely stressful due to some unexpected events, the students are allowed to apply for extensions on their FYP deadlines, which can help students manage stress, unfortunately Monash is unlikely to grant this extension simply due to poor time management. **Control Effectiveness:**	**Control:** Weekly meetings should be hold by the team to discuss the potential problems members of the team may face. Doing so will allow for collaboration and combined efforts in resolving specific problems, which can boost team morale and let members of the team feel supported.	YIDE TAO	19/08/2023
The work requires using both Monash AI Lab equipment and data sourced externally from D61. Handling data	**Control:** FYP students will only have access to the data after the paper	**Control:** Frequent meeting with the cooperating organisation and having the	YIDE TAO	17/08/2023

and equipment with professionality and care will ensure minimum damage is done to all participating parties.	from the cooperating organisation has been published, meaning even if the participants is to accidently leak the data the damage would be minimal. **Control Effectiveness:**	cooperating organisation lead students through an induction process should be relatively effective.
While Monash has mostly adapted to the post-COVID teaching and learning environment, it still needs to be noted that Victorian Government still recommend self-isolation for those who are tested positive for COVID. Students should still be aware of the virus as well as the potential damage virus can cause to surrounding people (particularly to those more vulnerable).	**Control:** It is still recommended that students are to wear a mask around elderlies and vulnerable. It is also recommended that students maintain a good hygiene e.g. wash hands often, block sneezes with tissue. **Control Effectiveness:** **Control:** When attracted COVID, it is recommended that students is to uptake immediate self-isolation and monitor their conditions closely. If they feel they may be endangered by COVID, please use the checklist (https://www.coronavirus.vic.gov.au/checklist-cases) provided by Victorian government to access the seriousness of your situation and seek medical help when necessary. **Control Effectiveness:** **Control:** Take the necessary vaccination as recommended for the student's age group and health conditions.	

Appendix C: Team Contract and Meeting Minutes

Team Contract

Team Contract – FYP – Galaxy Classification Task.

Team Name: Galaxy Classification FYP

Team Member Names:

Muhammad Suleman	Zach Drinkall
Katherine Hawkins	
Yide Tao	

1. Document Purpose

The purpose of this team contract is to outline the standard operating practices and team norms of the above-named team and individually listed members for the duration of the team's lifespan. The guidelines outlined in this document are agreed to by all team members as indicated by their signature at the end of the contract. Any amendments to the contract must be discussed and agreed to by all signing members. Failure to abide by the outlined standard operating practices of this contract could harm the team's overall functioning and result in penalizing action as detailed in the contract.

2. Rules and Regulations

The team agrees to the following guidelines regarding general procedures, practices, and behaviors that are deemed acceptable.

A. Expectations

i. Project Expectations

- It is essential to establish that: the project should not be conducted in any way that can impose harm or inflict damage upon others. All project members should follow the OHS Risk Management Plan outlined in the project proposal. Members should also be responsible for the surrounding people by actively reporting any potential hazards that may be a source of harm.

- It is also important to indicate that: the project requires formal research to be conducted throughout the project duration, therefore it is essential that all works produced in this project should be reproducible and of original research. Any usage of works and ideas of external origin should be cited and credited appropriately.

ii. Member Expectations

- Each member of the team must follow the academic integrity , respect, cultural and diversity requirements set out by Monash University.

- Each member of the team is expected to actively communicate with each other and with supervisors (Professor Mehrtash Harandi).

- All members of the team are expected to participate in the weekly meetings and complete the allocated tasks. If the agreed upon meeting or work cannot be achieved, the member must notify other members as soon as possible.

- It is expected that the workload of the project should be shared evenly among the members of the team. While it is expected that some members of the team will complete more academic work than others, it is still also expected that members of the team completing less academic work will put in appropriate level of efforts in assisting with the administrative aspect of the research.

- If any members of the team are offended or displeased by another member's behavior or action. It is essential that the victim of the offence is to point out the other member's mistake with constructed criticism. All discussion of potential improvement should be conducted in a polite and progressive manner.

- All discussion of sensitive personal matters or controversial topics should be avoided in formal and non-formal group discussions. All members should conduct research and cooperate with professionalism and respect.

- Members of the team should demonstrate an appropriate level of care and responsibility to other members of the team. Common questions such as "Are you okay?", "Are you feeling alright?", "How can we help?" should be asked when members of the team are observed to be in an abnormal psychological and physical condition.

iii. Role Expectations

- Team Leader: responsible for organizing meetings, managing teams, and communicating with supervisors on specific tasks.

- Research Lead: responsible for managing the academic aspect of the whole report.

- Software Lead: responsible for managing the software and version control aspect of the whole report.

- Scientific Consultant: responsible for managing and informing the astrophysics aspects of the report.
- Scribe / Writing Lead: responsible for recording the team minutes.

B. Communication

i. Communication Medium

- Slack, Messenger, and Email

ii. Communication Timelines

- All messages are expected to be checked and replied to within 24 hours. All members of the team are expected to check slack and email at least 2 times per day.
- Slack Channel should be used actively in times between 8:00am – 11:00pm.

iii. Communication Code of Conduct

- As mentioned above, all communication should be conducted in a professional manner. Communication with supervisors, PHD students and D61 members should be conducted maturely with a high level of respect.
- Please use "Thank you" whenever help is provided.

C. Team Meetings

i. Scheduling

- One formal full team meeting conducted at 5:00pm Monday.
- One non-formal stand-up meeting 10:00am Friday.
- Other meetings between subgroups may be scheduled when needed.

ii. Involvement

- Before the full team meeting, prepare a detailed document stating what should be discussed in the meeting, this will ensure the meeting is conducted in a constructive manner ad no time is wasted.
- Control the time for the stand-up meeting to be below 30 minutes.

iii. Attendance & Notice

- Disclose absence at least 3 hours prior to the meeting.
- Not all members should be reminded when attending fixed-time meetings. If you have arranged for a non-fixed time meeting, the organizer should remind the attendees about the meeting 15 minutes prior to the meeting.

D. Team Conflict & Decision Making

i. Conflict Code of Conduct

- **Avoidance:** All members of the team should constrain the discussion to be of purely academic and project-based matters, all discussions and criticisms should be all about research conducted rather than personal attacks.

ii. Initial Conflict & Conflict Escalation

- **Maintenance:** Working as a team for a long period of time will unavoidably encounter some form of conflicts. All members of the team should actively take part in the maintenance of long-term corporation. Taking a step when you feel the conflict is escalating is a very mature way of de-escalating the situation.

- **Recognition:** All members of the team should recognize criticism made by other members of the team towards one's work are not personal. Members of the team should also recognize when other members of the team are trying to de-escalate the situation, therefore apologize when needed and take a step back as well.

- **Aftermath:** If escalation of the situation has inevitably happened. Members of the team should actively work together to "mend the fences". Maintaining healthy work relationships is essential for the successful outcome of this research.

iii. Decision-Making

- **Democracy:** Major decisions should be discussed as a team before being made, A single person should not make major decisions without informing other team members.

- **Expertise:** Often in science, popular opinion does not make you correct. We must recognize facts rather than the majority option. It is therefore expected and allowed: if members of the team are to attempt something that they believe in but is not agreed upon by other members of the team, however that member(s) must take the responsibility of their attempt and understand the consequences of failure. They are also expected to put extra work to validate their ideas.

E. Stress Management

i. Monitoring & Assistance

- Discuss upcoming assignments and work when allocating tasks for everyone.

- Check if the workload is acceptable to the members of the team before allocation.

- Actively terminate other members' workload, when it is exceeding their capabilities and causing heavy stress on that particular member.

ii. Resources

- Mindfulness: https://www.monash.edu/consciousness-contemplative-studies/mindfulness-at-monash

- Consultation: https://www.monash.edu/students/support/health/counselling

- Rest: Rest is always the best friend against stress.

F. Contract Code of Conduct

i. Contract Breaches

- Breach of contract should be noted depending on the severity of the situation. If all members of the team deem the breach of contract as minor, then the member will be given three formal warnings before escalation to the Academic Supervisor.
- If the breach of a specific section of the contract is very serious, then without any warning the member of the team will be reported to the Academic Supervisor.

ii. Penalties

- Three formal warnings would mean that the member of the team's action should be noted in the mark allocation section of the final report. Thereby their marks will be deducted as a result.
- Reporting to the supervisor is more serious, which may lead to academic penalties and the member's removal from the research project.

3. Declaration

By signing below, team members acknowledge and agree to be bound by the guidelines outlined above.

Team Member Signature	Date
 Team Member Signature	1/09/2023 Date
 Team Member Signature	30/08/2023 Date
 Team Member Signature	30/08/2023 Date
 Team Member Signature	29/08/2023 Date
 Team Member Signature	30/08/2023 Date

Meeting Minutes

Only the last six week's Meeting Minutes are summarised and presented, if more information is needed, please send an email to ytao0016@student.monash.edu

Date of Meeting	Members Participated	Topic Discussed
18/09/2023 (Week 9)	Muhammad Suleman, Omar Khan, Katherine Hawkins, Zach Drinkall, Yide Tao	- Discussed SSH using visual studio and the different ResNet models (trained on different dataset). - Plan out the training and creation of the machine learning model. - Create the 500 by 500 images used for most of the prebuilt architectures. - Explore the N-dataset and consider sections to include in the report.
25/09/2023 (Mid Semester Break)	Muhammad Suleman, Katherine Hawkins, Zach Drinkall, Yide Tao, Omar Khan	- Explored ResNet and DenseNet architecture and plotted out the accuracy of both models. - Experienced huge problems with the four classes of classification, reduced the model classification to three classes. - Explored basic CNN models and found that CNN models generally are less prone to overfitting therefore more suitable as a starting point for the classification tasks.
2/10/2023 (Week 10)	Muhammad Suleman, Katherine Hawkins, Zach Drinkall, Yide Tao	- Omar is sick, leadership transferred to Zach. - Explored and optimized the classification model with the help of supervisors and explored the benefits of a new dataset CRUMB. - While exploring the classification and optimizing the hyper parameters, exploration of the python Grad-CAM package began.
9/10/2023 (Week 11)	Muhammad Suleman, Katherine Hawkins, Zach Drinkall, Yide Tao	- Optimized the dataset and explored the best transformation for the classification model. - Attempt to solve the problem of an unbalanced dataset.

		- Removed dropouts from simple CNN models. - Finalised a basic prototype for the Grad-CAM extractor.
15/10/2023 (Week 12)	Muhammad Suleman, Omar Khan, Katherine Hawkins, Zach Drinkall, Yide Tao	- Omar built a ViT model that outperformed all existing CNN models, indicating the significance of transformer based vision architectures. - Finalising the report. - Applied for the two day extension

Appendix D: Generative AI Statement

Generative AI use in FYP B (ENG4702)

The responses to this form will need to be copied and put into an appendix in your Final Report.

Email *

khaw0004@student.monash.edu

Name *

Katherine Hawkins

Campus

- (●) Clayton
- ( ) Malaysia

Host Department

- ( ) Chemical and Biological Engineering
- ( ) Civil Engineering
- (●) Electrical and Computer Systems Engineeing
- ( ) Materials Science Engineering
- ( ) Mechanical and Aerospace Engineering
- ( ) Software Engineering

Supervisor

Dr Mehrtash Harandi

This project has been conducted using AI tools *

- ( ) In this assessment, there will be no use of generative artificial intelligence (AI). All content in relation to the assessment task has been produced by the authors.
- ( ) In this assessment, the following generative AI will be used for the purposes nominated in part 2. (Please note: any use of generative AI must be appropriately acknowledged - see Learn HQ)
- ( ) In this assessment, AI writing assistants (e.g., Grammarly, Writesonic, Quillbot, Microsoft Editor) will be the only form of Generative AI used.
- (•) This project involves the development or authoring of Unique Generative AI, Unique operation of commercially available Generative AI OR Unique non-generative AI (Machine Learning, Artificial Neural Network, Logistic Regression, etc.)

Developing AI

In question 1, you answered your project involves the development or authoring of AI. Please choose the specific way you did this.

- ( ) Unique Generative AI
- ( ) Unique operation of commercially available Generative AI
- (•) Unique non-generative AI (Machine Learning, Artificial Neural Network, Logistic Regression, etc.)
- ( ) Other: .....

How has the technology be used?

How did you use this technology?

*

- [x] As a fundamental aspect of the study (i.e., this is a study centred on generative AI, human interaction, associated ethics, etc.)
- [ ] Audio Transcription
- [x] Coding/Scripting
- [ ] For the operation of robotics
- [x] Generation of novel content - Datasets
- [x] Generation of novel content - Graphics/Images
- [ ] Generation of novel content - Video
- [ ] Generation of novel content - Writing
- [ ] Idea generation
- [x] Initial research
- [ ] Machine Language Translation
- [ ] Mathematics
- [ ] Paraphrasing
- [ ] Proofreading
- [ ] Text Analytics
- [ ] Text Summarisation
- [ ] Thematic analysis

- [ ] Thematic analysis
- [x] Visualisation (of data)
- [ ] Writing assistance
- [ ] Other: .....

How was the Generative AI response validated?

.....

Permissions

The use of Generative AI has been discussed with and approved by my academic supervisor. *

- (•) Yes
- ( ) No

End

Thank you for completing this form - your responses will be emailed to you for your Progress Report

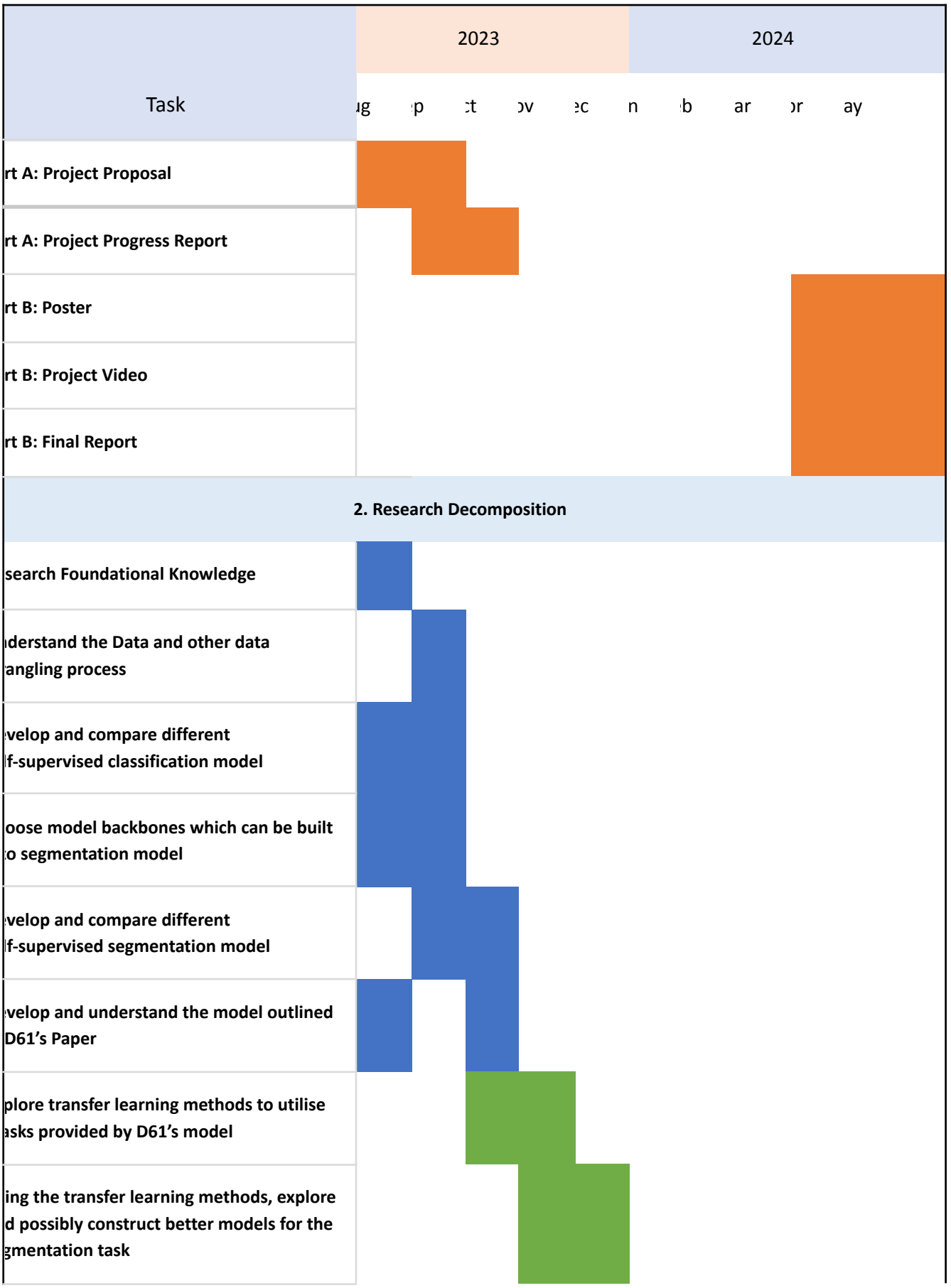
Appendix E: Full Time Line

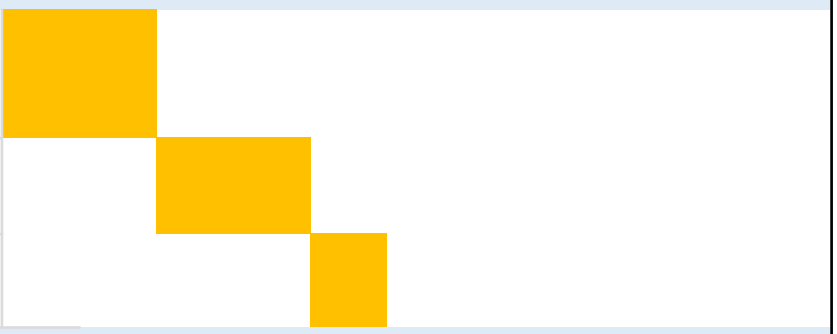
Stage	Activity Name	Criteria	Due Date	Gantt-chart colour
1. Assessment Milestones				
1	Part A: Project Proposal	Feedback and marks (30%) **Require Stage: 6, 7**	1st of September 2023	
2	Part A: Project Progress Report	Feedback and marks (70%) **Require Stage: 6, 7, 8, 9, 10, 11**	20th of October 2023	
3	Part B: Poster	Feedback and marks (10%) **Required Stage, 6-22**	17th of May 2024	
4	Part B: Project Video	Feedback and marks (10%) **Required Stage, 6-22**	17th of May 2024	
5	Part B: Final Report	Feedback and marks (80%) **Required Stage, 6-22**	24th of May 2024	
2. Research Decomposition				
6	Research Foundational Knowledge	Understand the basics of transformers, attention mechanism, self-supervised learning, transfer learning, machine vision evaluation metrics and other machine learning related knowledge.	13th of August 2023	
7	Understand the Data and other data wrangling processes	Understanding the Noise level, channel definition (RGB or other means) and data representation of MiraBest, Galaxy Zoo and possibly D61 data. **Required Stages: 6**	10th of September 2023	
8	Develop and compare different self-supervised classification model	Each member of the team should develop one of the models they are interested in and present it in a team discussion.	17th of September 2023	

		Required Stages: 6, 7		
9	Choose model backbones which can be built into segmentation model	N/A	23rd of September 2023	
10	Develop and compare different self-supervised segmentation model	The quality of the segmentation model created compared to the model outlined in Zeeshan's paper. **Required Stages: 8, 9**	1st of October 2023	
11	Develop and understand the model outlined in D61's Paper	**Required Stages: 9, 10**	8th of October 2023	
12	Explore transfer learning methods to utilise masks provided by D61's model	Discuss and record the attempts made. **Required Stages:10, 11**	25th of November 2023	
13	Using the transfer learning methods, explore and possibly construct better models for the segmentation task	Using the IOU criteria as the basis of comparison. **Required Stages:7, 10, 11, 12**	9th of December 2023	
3. Project Management				
15	Develop a foundational understanding of the expectations of the project.	Understand and develop the scopes, available resources, and potential paths for approaching this project.	25th of September 2023	
16	SWOTVAC and Exam Weeks	N/A	17th of November 2023	
17	Christmas Holiday	N/A	25th of December 2023	
4. Continued Research and Model Development				
18	Further Explore the potential architecture of different models and begin	**Required Stages: 14**	31st of January 2024	

	exploring the attention map of the models			
19	Come together and discuss models of interest, in particular recent papers that have been published in the field. Decide whether these are relevant to our work.	**Required Stages: 18**	28th of February 2024	
20	Finalise the model and compare data gathered. Begin creating plots and comparison that needs to be included on the final report	**Required Stages: 18**	31st of April 2024	
21	Finalise the theoretical backbone and conclusions need to be made in the report using the data gathered	**Required Stages: 6-21**	10th of May 2024	
22	Finalise the full detail of the paper	**Required Stages: 6-21**	22nd of May 2024	

Gantt Chart


2023
2024
Task
rt A: Project Proposal
rt A: Project Progress Report
rt B: Poster
rt B: Project Video
rt B: Final Report
2. Research Decomposition
search Foundational Knowledge
derstand the Data and other data angling process
velop and compare different f-supervised classification model
oose model backbones which can be built o segmentation model
velop and compare different f-supervised segmentation model
velop and understand the model outlined D61's Paper
plore transfer learning methods to utilise sks provided by D61's model
ing the transfer learning methods, explore d possibly construct better models for the gmentation task

3. Project Management	
velop a foundational understanding of the pectations of the project.	
/OTVAC and Exam Weeks	
ristmas Holiday	
4. Long Future	
rther Explore the potential architecture of ferent models and begin exploring the tention map of the models	
Come together and discuss models of interest, in particular recent papers that have been published in the field. Decide whether these are relevant to our work.	
alise the model and compare data thered. Begin creating plots and mparison that needs to be included on the al report	
alise the theoretical backbone and nclusions need to be made in the report ing the data gathered	
alise the full detail of the paper	

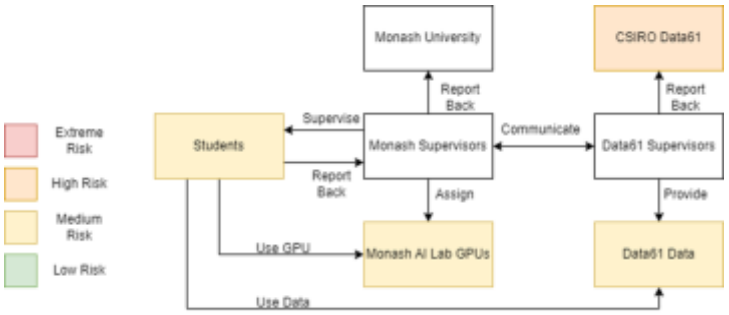
Appendix F: Risk Management Plan

Monash University mandates all FYP students to complete a Risk Management Plan for their project. The project "Unlocking Celestial Enigmas: AI Driven Exploration of the Universe" will mainly involve low risk computer-based work, but still carries some non-OHS and OHS related risks. Establishing a comprehensive risk management is vital for the project as the project will involve:

1. Working a long period of time with high powered computer units.
2. Collaborating with external parties particularly D61 from CSIRO.
3. Being a team-based project, involves collaborating with different members of the team for a long time.

With a mature Risk Management Plan, it will act as a guide for students when faced with unexpected events, therefore helping students stay calm and make correct decisions.

Hazard Overview



Appendix F Figure 1 - Risk Distribution Architecture

Using block diagrams to conduct a brief analysis, it has been noted that the major source of hazards is expected to be generated by:

1. Students Block: students are the major participant and therefore risk owner of the project. They are responsible for overseeing the entire project process and are also expected to be the major source of non-OHS and OHS related risks.
2. Monash AI Lab GPUs: these GPUs are the assets provided by Monash, being both high power and expensive, mismanagement of these GPUs is expected to cause problems.
3. Data61 Data: These data are close-source (as of 21st of August 2023) data provided by respected research institutes, mismanagement of these data can potentially lead to reputational and economical losses for all participating parties especially on CSIRO.
4. CSIRO Data61: While the participants of the project are promised with the data from the radio telescope. Due to administrative and security concerns the CSIRO may not approve such data.

Table 11: Risk Identification Table

Occupational Health and Safety	Project	Liability	Cultural, political social	Environmental
• Workplace injury (e.g., electrical shock, tripping and heat from GPUs, refer to appendix for close analysis). • Sight related health issues due staring at screen for a long time. • Ergonomic issues. • COVID • Psychological issue due to stress.	• Completion time exceeds the deadline. • Hardware provided is not powerful enough to train the proposed model. • Approval issues (Data is not approved by CSIRO).	• Breach of research protocol, students did not follow the academic integrity guide set out under Monash Academic Integrity. • Breach of confidentiality, data provided got leaked.	• Members in the group got offended by other members intentionally or unintentionally	• High energy usage by GPUs when training the model may be causing emission concerns.

OHS Project Risk

As requested in this task, the OHS risk management has been attached to the appendix.

Non-OHS Project Risk

Table X: Non-OHS Risk Assessment

Project Risk	Risk	Likelihood	Consequence	Risk level	Mitigation	Residual Risk
Delayed delivery of Assessment Milestone	Due to some unforeseeable events, components of the Assessment Milestone have failed to be delivered.	Rare	Disastrous	L	Whenever a potential event that may result in the delayed completion of the project has occurred, report the even to the supervisor and the unit coordinator immediately. To prevent such catastrophes from occurring, members of the team should communicate regularly and openly disclose problems they are having so that the team can resolve them together ASAP.	While the likelihood of this risk occurring is relatively low. If this risk is bound to happen and there is really nothing the team can do to prevent the risk from further escalating. The best way to minimise loss is to maintain an open line of communication with the unit supervisor to explain the predicament of the team and hope for some special considerations to be granted.

Delayed delivery of Project decomposition	Due to some unforeseeable events, some/all members of the team did not complete the project to the standard that is laid out in the project proposal	Unlikely	Serious	M	Ensure regular meetings and updates on the progress of each member's completion of the task. Make sure to allocate the tasks suitable to the abilities of the team member (it is not ideal to ask a mechanical engineer to draw semiconductor circuits).	If this event is to happen, make sure to catch up on the late completion. It is also recommended to send another team member to help with the completion of the uncompleted task.
Breach of confidentiality	Due to some mishaps, mistakes or in the worst-case intentional sabotage, the data that is provided by D61 is leaked before becoming open-source.	Unlikely	Catastrophic	S	Work with extreme caution when using the data provided. Establish a protocol for using and monitoring who is using the data and who has access to the data. Work as a pair when using the data so that one member of the pair can monitor the other's action.	Report back to the supervisors and all parties involved if the data is suspected of being leaked or the work environment is under-attack by external sources.
Data Approval issue	Due to some considerations, CSIRO ultimately did not approve the usage of data by Monash students during the FYP period.	Possible	Disastrous	S	No mitigation can be done to prevent this risk.	Actively explore other data sources available for usage that fits the scope of the project. By communicating regularly with D61 and hope to persuade CSIRO into unrestricting the data through work.
Violating Academic integrity	Due to some mistakes, members of the team have violated academic integrity rules in conducting research or publishing data.	Possible	Serious	M	The group should review the report and the data multiple times before submitting the work to the supervisor for further reviews. Discussion on academic integrity should be conducted and reviewed by the team. If violation of academic integrity is found by members of the team (whether if the violation is accidental and intentional), it should be	While the likelihood of intentional academic integrity violation is relatively low for this project due to the comparatively low stake. It is still possible for students to conduct research in an unprofessional way due to lack of experience.

					reported to the supervisor and Monash University without hesitation.	
Departure of a team member	Due to unforeseeable circumstances, a team member has to leave the project before its conclusion.	Unlikely	Serious	M	The group should meet regularly to ensure members can express concerns and come forward if they are having difficulty with workload. From here the team can work towards dividing more further so the individual can get back on track.	If a member does have to leave the project, remaining members should meet to divide their workload evenly. The work of the departing member should be analysed by those in the team and then reassigned based on ability.

Appendix G: Sustainability Plan

Admittedly, being a computer based project, the project is not expected to have many strong real world sustainability or ethical implications. However, being part of Data61's avocation in “AI for science”[33], the project is expected to help explore a new frontier of AI usage, which contains many long-term social-environmental implications and uncertainties. It is important for the researchers of this project to proceed with caution. Therefore, establishing a base-line for the ethical, sustainable usage of AI in alignment with the 17 Sustainable Development Goals proposed by the United Nations[34] is considered good practice.

The research focuses on exploring machine learning tools for assisting fundamental scientific research. The project emphasizes the importance of modern AI should not be monopolised but be used to encourage international partnerships and collaborations. Hence it is believed the project’s core value mostly aligns with Goal 17 of the SDG, which is “partnerships for the goals”, promoting unity in addressing challenging issues through AI exploration.

Partnerships for the goals

“The Earth is the cradle of humanity, but mankind cannot stay in the cradle forever”

Konstantin Tsiolkovsky

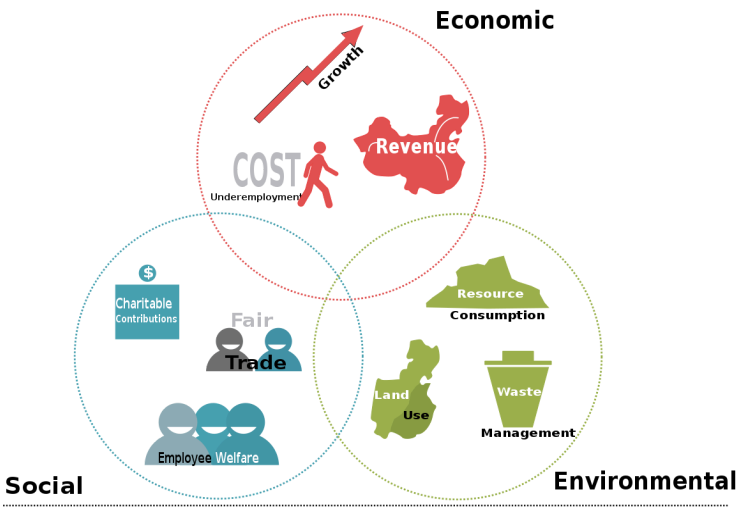
Targets and indicators

After accessing the Targets and indicators outlined in the “partnership and goals”, the three targets above are shown to be most aligned with this particular project, these targets will be discussed and referenced to in the sections below.

- **Target 17.17,** “Encourage and promote effective public, public-private and civil society partnerships, building on the experience and resourcing strategies of partnerships”.
- **Target 17.8,** “Fully operationalize the technology bank and science, technology and innovation capacity-building mechanism for least developed countries by 2017 and enhance the use of enabling technology, in particular information and communications technology”.
- **Target 17.5,** “Adopt and implement investment promotion regimes for least developed countries”.

Project Plan and Progress

Project Plan - Triple Bottom Line



Appendix G Figure 1 - Triple Bottom Lines

Table 13: Triple Bottom Lines Identification

Environmental Collaborations	Social Collaborations	Economical Collaborations
- Creating a more efficient and lightweight classification model will reduce the computing and manual resources needed for doing arbitrary tasks such as data labelling in scientific fields. - The Project aims to promote the environmentally friendly usage of computing resources, by advocating and enforcing sustainable practices such as turning off the computing resources when they are not being used and booking for the session usage beforehand.	- The plan of this project is to develop a project with open-source code accessible by the general public following the guidelines set out by the Open Code [36] and Open Science [37] initiatives. This will allow astronomy and Machine Learning enthusiasts from across the world to have access to the code (**Target 17.8**). - The plan of this project is to create a machine-learning model suitable for processing data generated using different telescopes, hence allowing the model to be used and improved upon by researchers from around the world (**Target 17.8,**	- In the medium to long term, machine-learning based data automation will allow for the processing of even larger quantities of data. Hence promote the construction of larger and more advanced telescopes from across the world, most of which will be in underdeveloped regions. This will promote regional awareness, raise the scientific foundation and generate positive economic externalities for the corresponding region (**Target 17.5)**. - Development in the machine-learning process for streamlining scientific procedures will reduce the

	Target 17.17).	burden of every-day researchers from across the world. Hence increasing the innovation efficiency, correspondingly boost economic growth and generate positive externalities.

Project Progress - Eight Pillars of Open Science

Table 14: Pillars of Open Science

Pillars of Open Science [38]	Relation to Project Progress
Public participation in research	- The program includes participants from diverse backgrounds and demographics. Including researchers from CSIRO and Monash University, as well as students of various genders, ethnicities and nationalities. - The project is developed using the open-source data such as MiraBest [31] and CRUMB , making the project easily accessible and reproducible for global public (enthusiasts, public institutions, private companies etc.) from across the world. This action aligns with **Target 17.17** of Goal 17 of SDG**,** further modification, collaboration across regions and improvements to the project are also welcomed, as long as it follows the guidelines set out by Monash Open Access [39].
Research Integrity	- As stated in Section 7 of the report, maintaining academic integrity and research integrity are essential bottom lines for this research project. Also stated in the Team Contract from Appendix B, research Integrity should be maintained for all collaborator parties to ensure long-term partnership, and minimise waste of resources and damages to the institute reputation due to false or careless publications. - Being an Open Source project, the documentation generated for the project is also expected to be maintained for consistent usage, hence attributing to the creation of a “technology bank” that is accessible to anyone in the world, hence aligning with **Target 17.8** from Goal 17 of SDG.
Next-generation metrics	- On top of researching and improving the accuracy of the classification model, this project also aims to explore the creation of lighter CV models that will reduce the energy and time consumption for computing the

	segmentation masks and labels of images. This will reduce the hardware requirement for uptaking “AI for Science” studies and reduce energy consumption during these studies.
Recognition and rewards	- Not Applicable to this project
The future of scholarly publishing	- Not Applicable to this project
FAIR data management	- The data generated from this project (Seeds, Pseudo-Segmentation Masks) will be accessible for public usage through the GitHub page linked to the project report after the completion of this project. The data generated can be studied and used for the: 1. Pretraining existing galaxy morphological segmentation models. 2. Exploring the model attention distribution for non-RGB images.
Education and skills	- The model developed is expected to be a lightweight model with a minimum hardware requirement, hence the model is capable of being used as educational material for countries of different levels of development . It is believed by the students and the supervisors that the application of AI and Machine Learning should not be limited to powerful hardware and purely commercial usages, people from less developed nations should also be able to participate in the research and be benefited from studies in this field. - The project also contains collaboration parties from both Monash University and CSIRO. The project aims to explore a new model for future collaborations, where research agencies and universities from across the world can come together to solve real-world research problems. Hence boosting research awareness and allowing undergraduate students to experience professional standards for research first-hand.

Project Implications

Table 15: Stakeholder Analysis

Stakeholders	Implications
Internal Stakeholders	
Monash University	The project's progress and performance are directly linked to Monash University. The project receives the software, hardware, educational and research support from the university. The project aims to follow Monash's 2030 "Net Zero" initiative through sustainable allocation and usage of computing resources. Unfortunately, the project will not be able to directly cut emissions, therefore will still have a net positive emission output.
CSIRO D61	The project receives technical and potential data support from the CSIRO D61 team (unconfirmed as of this moment). D61 is interested in this collaboration and believes AI is essential in helping create a "sustainable future" [33]. While the model developed in this project will not contribute directly to this goal, the insights gained from developing the model can be potentially used for the development of algorithms that drive a sustainable future.
External Stakeholders	
Other Researchers	The major concern is whether AI will replace human researchers entirely. DeepMind's AlphaFold [40] has shown promising results in predicting 3D architectures of proteins . While the future impacts of AI advancements are uncertain, the consensus for this project is: Due to the large amount of data generated by large sky surveys, Machine Learning is expected to help researchers better understand the data gathered. Models similar to these will reduce the time used on solving arbitrary tasks such as image labelling and classification. Similar research will also encourage multidisciplinary and multi-national collaboration for the pursuit of a common goal of galactic exploration.
General Public	The model developed will be accessible on GitHub for all interested parties, if additional funding is secured. The model can be transformed into an API, expanding access for all astronomy enthusiasts. This will promote AI in

	fundamental research and generate interest among pre-university and undergraduate students.