Harnessing Unlabelled Data for Automatic Aerial Poacher Detection

Reducing Annotation Costs through Unsupervised and Self-Supervised Learning

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Samantha Ball
1603701

Supervised by Dr Richard Klein

May 26, 2023

A research report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Masters of Science in Data Science

Abstract

The recent escalation in wildlife poaching poses a major threat to the survival of several key species, with South Africa and Zimbabwe forming the epicentre of the poaching crisis. The application of emerging technology such as Unmanned Aerial Vehicles (UAVs) and object detection provides a novel way of tackling this issue through the use of aerial surveillance. However, despite pioneering studies into the practical use of computer vision for poacher detection, current models require detailed ground truth. Notably, the sparsity and small-scale of objects in poaching detection data renders the annotation process particularly expensive and time-consuming, posing a barrier to resource-constrained conservation organisations in real-world scenarios. To reduce the need for costly annotations, this study explores the use of the self-supervised DINO model and unsupervised anomaly detection network FastFlow to provide pseudo-labels for unlabelled data. The value of these alternative techniques is evaluated on real-world poaching detection data provided by a Southern African conservation NPO. The results indicate that a YOLOv5 detection model can be trained using pseudo-labels together with only a small fraction of manually-annotated ground truth for the most difficult training videos. The resulting models attain over 90% of the detection recall of a baseline model trained with original ground truth labels, while also maintaining real-time detection speeds. This reduction in annotation cost would allow current systems to harness large unlabelled datasets with greatly reduced annotation effort and time, while still meeting the efficiency constraints associated with the UAV platform.

Declaration

I, Samantha Ball, hereby declare the contents of this research report to be my own work. This report is submitted for the degree of Masters of Science in Data Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

 Samantha Ball 26 May 2023

Acknowledgements

I would like to thank my supervisor, Dr. Richard Klein, for providing consistent and valuable insight, guidance, encouragement and expertise over the course of the project. Secondly, I would like to acknowledge SPOTS (Strategic Protection Of Threatened Species) and FruitPunch AI for the collection of the SPOTS poaching detection dataset, as well as their real-world insights and collaboration on this research project. Lastly, I would like to thank God without whom none of this would be possible.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Wildlife poaching has emerged as an urgent issue in recent years with severe increases in demand for ivory and rhino horn casting poaching as a lucrative opportunity. This increase has driven species such as the African bush elephant (*Loxodonta africana*) to endangered status and the black rhino (*Diceros bicornis*) to near extinction [Moodley *et al.* 2017]. As reported by the wildlife trade monitoring network TRAFFIC [Emslie *et al.* 2016], South Africa forms the epicentre of the poaching crisis, containing 88% of all identified illegal killings of rhinos between 2010 and 2016.

According to the South African National Parks (SANParks) annual report in 2020 [SANParks 2020], the Kruger National Park, which contains the largest rhino population in the world, has only 3549 white rhinos and 268 black rhinos remaining. Despite numerous conservation efforts, the current poaching rate is approximately 1 rhino poached every 22 hours which remains too high to ensure sustained rhino population growth [SANParks 2020]. Thus, the severe and immediate threat of wildlife poaching necessitates effective solutions to aid the ability of manual patrols to timeously apprehend poachers and thus protect these at-risk species.

1.1.1 Automatic Poacher Detection

A myriad of sensor technologies have been employed to help combat wildlife poaching. Detection technologies can include motion sensors, radar, acoustic, optical, infrared and chemical techniques [Kamminga *et al.* 2018]. Stemming from the increasing availability of drone technology, aerial surveillance has been employed to detect poachers and relay poacher locations to ground response units. Several non-profit organisations such as SPOTS (Strategic Protection Of Threatened Species) and Air Shepherd have successfully deployed aerial surveillance in national reserves across Malawi, Zimbabwe, South Africa and Tanzania, indicating the value of this strategy [Air Shepherd 2021].

Despite these successes, current practice relies on manual monitoring of video footage from visible light (RGB) cameras and thermal infrared (TIR) sensors which requires significant manpower, ranger availability and attentiveness in order to accurately detect

poaching threats in time for rangers to respond [Bondi *et al.* 2018b]. Thus, emerging computer vision techniques such as object detection provide a promising alternative to manual surveillance with the potential to automatically process vast amounts of aerial data and locate possible poaching threats. Object detection networks are capable of locating and classifying all the objects in the image, therefore providing a potentially efficient solution to the issue of poacher detection.

1.1.2 Current Limitations

Pioneering work into the practical use of object detection to identify poachers in surveillance footage has been undertaken by several studies, such as Bondi *et al.* [2018b] in partnership with the Air Shepherd organisation. However, significant limitations remain in terms of the necessary annotation effort to provide detailed bounding boxes for extensive amounts of training data. Objects in aerial conservation data are both sparse and very small in the frame, leading to an arduous annotation process [Bondi *et al.* 2017]. Moreover, the thermal infrared video stream collected by the conservation UAVs poses specific challenges which increase the difficulty of annotation. Thus it is pertinent that unsupervised and self-supervised techniques are explored in the context of poaching detection to overcome the barriers associated with obtaining fully annotated ground truth.

An additional challenge associated with the development of an effective poaching detection system is the speed at which the object detection network can process image data. Real-time response is essential for manual patrols to successfully apprehend poachers, therefore requiring rapid processing of the available surveillance data. Consequently, any candidate object detection network must be able to perform in real-time in order to provide a realistic solution. Moreover, due to the intermittent internet coverage available in the field, it is often not viable to stream the thermal infrared video back to a base station for processing and therefore object detection must be performed onboard the UAV [Bondi *et al.* 2018b]. This brings about additional efficiency constraints associated with running the object detection model on a computationally constrained platform.

In light of these limitations, this work proposes the application of current advances in unsupervised and self-supervised learning to the context of poacher detection. The proposed techniques are evaluated on real-world, thermal infrared data from conservation drones in order to ascertain their effectiveness in terms of accuracy, detection speed and model efficiency. This provides crucial insight into how current systems can be improved by leveraging state-of-the-art unsupervised and self-supervised techniques for poaching detection.

1.1.3 Reducing the Annotation Cost of Poaching Detection

While several differing approaches could aid in reducing the time and annotation effort involved in training an effective poaching detection network, certain approaches are

unable to meet the speed and efficiency constraints associated with the task. Two major ways in which annotation cost could be decreased include either reducing the level of supervision with which the object detection network is trained, or directly reducing the time and manual effort needed to produce labels for a fully supervised poaching detection network.

While decreasing the level of supervision required to train the network may initially seem to be the better solution, the resulting models are often unsuitable to be run onboard the conservation UAV. An example of this is the field of weakly supervised object detection (WSOD), where object detection networks are trained using only image-level labels and therefore do not require detailed bounding boxes. Although the field of weakly supervised object detection has developed rapidly in recent years, many algorithms remain reliant on multiple pre-generated object proposals which would render them unsuitable for real-time poaching detection [Shao *et al.* 2022]. In addition, these networks often struggle with small-scale objects which are prevalent in poaching detection data [Shamsolmoali *et al.* 2021]. Therefore current methods of training object detection networks in a weakly supervised manner are not the best fit for the poaching detection task. Similarly, many object detection networks trained with alternative methods of supervision are unable to meet the strict real-time and size constraints of the working detection model or are incompatible with the hardware onboard the UAV platform.

Consequently, this study aims to directly reduce the effort and time required to produce labels for fully supervised poaching detection. Thus, we investigate alternative means to provide full annotations for aerial datasets through the use of state-of-the-art unsupervised and self-supervised techniques. The resulting pseudo-labels are then used to train a lightweight detection model in a fully supervised manner, which would then be able to run in real-time onboard the conservation drone. Thus the study aims to reduce the annotation cost of training of a fully supervised poaching detection model, while ensuring the practical use of the resulting system.

1.2 Problem Statement

The primary objective of the proposed study is to investigate the feasibility of using unsupervised and self-supervised techniques to produce pseudo-labels for unlabelled poaching detection data and thus enable the training of a poaching detection network with reduced annotation cost. Due to the expensive and time-consuming nature of obtaining fully annotated ground truth for large amounts of aerial data, harnessing recent advances in unsupervised and self-supervised learning to provide pseudo-labels could provide a sustainable solution to train effective poaching detection networks in the real world. The study focuses on thermal infrared data in favour of visible light (RGB) data due to the relative lack of research into the use of unsupervised and self supervised learning with thermal infrared images and the critical need for poacher detection in low light conditions.

1.3 Research Hypothesis

This study proposes that recently developed unsupervised and self-supervised techniques can be used to produce pseudo-labels for large amounts of unlabelled training data and thus reduce the annotation cost required to train a poaching detection network, while maintaining the same recall/detection rate and speed as current state-of-the-art models, as measured by recall (R) and frames per second (fps).

1.4 Research Objectives

Consequently, this study will explore the following research questions:

- Can self-attention maps from the self-supervised learning method DINO be used to semi-automate the annotation of poaching detection data by producing pseudo-labels?
- Similarly, can segmentation maps obtained from unsupervised anomaly detection algorithms produce pseudo-labels for unlabelled poaching detection data?
- To what extent can training a YOLOv5 object detection model with the resulting pseudo-labels match state of the art fully supervised detection performance?
- Can the generated pseudo-labels be combined with a small proportion of ground truth labels in a way that enhances overall detection performance?
- Will the resulting network be of practical use in the real poaching detection scenario in terms of real-time requirements and model efficiency on the UAV platform?
- Which of these approaches provides a better solution for the challenges of poaching detection?

1.5 Scope and Limitations

Due to the time constraints associated with the project, only a limited number of unsupervised and self supervised techniques were evaluated. The investigation covers the techniques which appear most promising for the poaching detection task. However, further methods within each class will require further study. Secondly, the project is limited to evaluation of different techniques on pre-recorded data rather than using a live system. Therefore some practicalities associated with the real-world system are unable to be represented. Nevertheless, this investigation aims to provide insight into the use of unsupervised and self-supervised techniques in the novel conservation context by utilising real-world, conservation-specific data.

1.6 Ethical Considerations

The dataset provided by the Strategic Protection of Threatened Species (SPOTS) organisation is closed due to the sensitive nature of the information and thus appropriate care must be taken to ensure the privacy of the data is maintained. Therefore none of the data used in this study may be released. The dataset contains no identifiable human participants since the small-scale of the poachers and thermal infrared nature of the image data renders them unrecognisable.

1.7 Structure of Report

The report is structured as follows. Firstly, in Chapter 2, relevant background regarding the mechanisms of object detection networks and the distinction between fully supervised, unsupervised and self-supervised approaches is provided. Subsequently, Chapter 3 provides a review of recent studies employing computer vision in conservation, as well as novel developments in self-supervision and unsupervised anomaly detection. Thereafter, Chapter 4 details the research methodology including the creation of the unsupervised and self-supervised pseudo-labelling pipelines, training of the detection network and evaluation against the fully supervised baseline. In Chapters 5 and 6, the results are presented and analysed in relation to the primary research questions. Lastly, Chapter 7 highlights the core aims and overall significance of the project.

Chapter 2

Background

A technical overview of object detection together with the distinctions between various levels of supervision is provided in this chapter as background to the concepts discussed in the subsequent literature presented in Chapter 3.

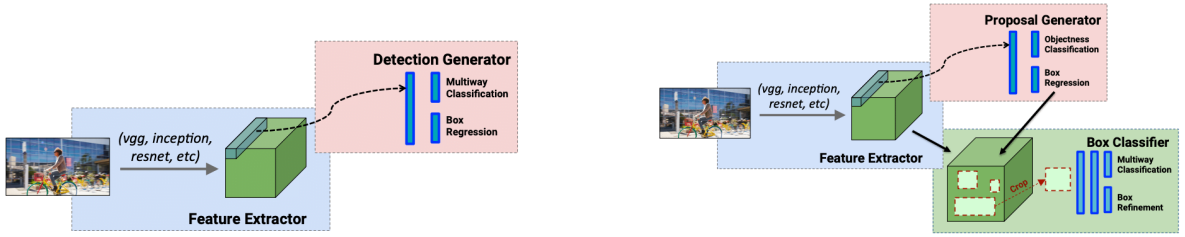
2.1 Object Detection

Object detection can be defined as the task of locating and categorizing objects in an image [Liu *et al.* 2021]. The object detection model must therefore determine the spatial location of each object in the image, usually represented by a bounding box surrounding the object, and classify the object as one of multiple predefined categories. Object detection constitutes a fundamental computer vision task, necessary for more complex processes such as semantic segmentation and object tracking [Liu *et al.* 2021].

2.1.1 Architecture

Stemming from early approaches based on hand-crafted features, the rise of deep learning has allowed for vast improvements in object detection accuracy. State-of-the-art techniques for object detection rely on deep networks for both classification and localization of objects. Recent techniques can be viewed in terms of two major categories based on their overall network architecture. Two-stage models such as Faster R-CNN have an initial region proposal stage which recommends areas of the given image that are likely to contain objects [Ren *et al.* 2016]. These estimated regions are then classified and refined in the second stage of the network. In contrast, one-stage models such as Single Shot Detector (SSD) [Liu *et al.* 2016] and You-Only-Look-Once (YOLO) [Redmon *et al.* 2016] do not utilise a separate stage for region proposal and instead present a unified framework which performs object classification and localization using a single feed-forward network.

While one-stage object detection models provide the advantage of balancing both accuracy and inference time [Liu *et al.* 2020], two-stage approaches still far outperform one-stage models with respect to detection accuracy.



(a) One-Stage

(b) Two-Stage

Figure 2.1: Basic architecture of the one-stage and two-stage object detection networks [Huang *et al.* 2017].

The basic architecture of one-stage and two-stage object detection networks is depicted in Fig. 2.1, indicating the feature extraction stage, region proposal network where applicable and the final classification stage.

2.1.2 Methods of Supervision

The differing methods of supervision are now outlined, together with their application to the task of object detection.

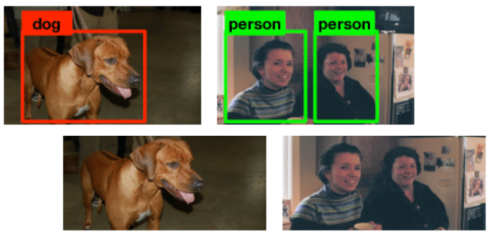
Fully Supervised

Fully supervised object detection networks learn to localise objects in an image using fully annotated ground truth data. These annotations may take the form of bounding box or point co-ordinates, as indicated in Fig. 2.2a which shows bounding box annotations and class labels for each object in the image. Obtaining detailed bounding box annotations for vast amounts of image data can be both costly and time-intensive, as annotation needs to be performed either manually or using automated tools [Tang *et al.* 2018]. Moreover, errors in the annotations may also lead to poor performance of the object detection model. Often, obtaining fully-labelled datasets may be impractical in a given setting. Consequently, techniques to decrease reliance on fully annotated data are vital for many real-world applications.

Semi-Supervised

In semi-supervised learning a small percentage of the training examples are fully annotated, with the remaining training examples having partial or no labels. An example of this in the object detection context is included in Fig. 2.2b. Semi-supervised learning can thus be viewed as a combination of supervised and unsupervised learning. Often, a pseudo-labelling technique is implemented where the model is first trained with the limited set of labelled data. The model is then utilised to provide pseudo-labels for the further unlabelled data [Kim *et al.* 2022]. Thus, semi-supervised learning differs from unsupervised learning in that a small amount of labelled data is still provided.

(a) Fully supervised object detection with bounding box annotations



(b) Semi-supervised object detection with a small percentage of fully labelled and large percentage of unlabelled data

(c) Weakly supervised object detection with image-level labels

(d) Self-supervision uses the underlying properties of unlabelled data to form a supervisory signal

Figure 2.2: Different levels of supervision for object detection [Yan *et al.* 2017].

Weakly Supervised

To alleviate the need for full supervision, weakly supervised techniques aims to learn from incomplete, partial or noisy labels. In practice this often translates to the label having less detail than the target. In the context of object detection, networks may learn from only image-level labels indicating the presence or absence of an object, or counts representing the number of the objects in the image [Kellenberger *et al.* 2019]. This can be seen in Fig. 2.2c showing image-level class labels. Therefore the ground truth is only partial as it lacks localization information. Weak supervision is particularly beneficial in cases where rapid adaptation or iteration is required.

Unsupervised

Unsupervised learning refers to scenarios in which patterns or trends are observed from completely unlabelled data. Common forms of unsupervised learning include hierarchical and probabilistic clustering, and dimensionality reduction. In the context of the study, the anomaly detection algorithms explored are described as unsupervised, as they are termed in the literature [Akcay *et al.* 2022; Yu *et al.* 2021]. However, strictly speaking the anomaly detection algorithms included in the paper could be considered to include a form of weak supervision as images are first classified as normal or abnormal before training of the anomaly detection model with only normal images.

Self-Supervised

Lastly, self-supervision refers to a situation in which the data is unlabelled but uses properties of the underlying structure of the data to form a supervisory signal from which to train. Most commonly, a pretext task is defined through which representations are learnt and then utilised in further downstream applications [Misra and Maaten 2020]. Thus self-supervised tasks learn to utilise data where no labels are provided,

as represented by Fig. 2.2d. Recently, novel self-supervision techniques have been developed using the properties of the Vision Transformer architecture, which is outlined in the following section.

2.2 Vision Transformers

Extensive success in Natural Language Processing stemming from the use of the Transformer architecture [Vaswani *et al.* 2017] have led to great interest in the use of Transformers for visual applications. Recently, the Vision Transformer (ViT), as introduced by Dosovitskiy *et al.* [2020], applied Transformers to the computer vision domain by feeding image patches as inputs to the Transformer encoder. The image patches are likened to word tokens in NLP tasks. The primary distinctive characteristic of the Transformer encoder is the use of layers of multi-headed self-attention. These core characteristics are summarised in the architectural diagram below in Fig 2.3.

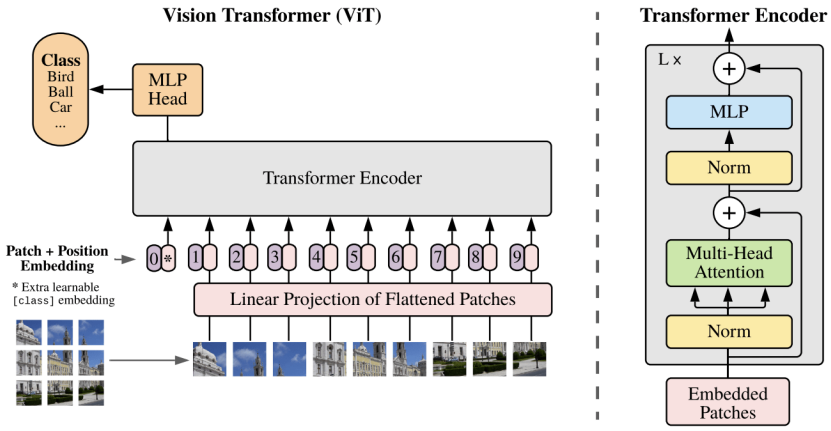


Figure 2.3: Architecture of the Vision Transformer, as proposed by Dosovitskiy *et al.* [2020].

Subsequently, a myriad of enhancements to the Vision Transformer have been proposed. Most notably, Data-Efficient Image Transformers (DEit) improve upon ViT by reducing the data and compute power required to train the model through the use of a teacher-student strategy [Touvron *et al.* 2021]. In addition, cross-covariance image transformers known as XCiT aim to combine the scalability of convolutional neural networks with the increased accuracy of the Transformer network, leading to high performance with fewer computational requirements [El-Nouby *et al.* 2021].

The properties of the Vision Transformer when combined with the self-supervised learning method DINO are of particular interest to this study. The attention maps generated have an inherent ability to pinpoint objects of interest in the image, thus leading to the possibility of transforming these attention maps into pseudo-labels. These concepts are further explored in the next section where we review existing literature on the use of computer vision for poaching detection, its limitations in terms of annotation cost and several alternative methods of supervision.

Chapter 3

Literature Review

3.1 Introduction

The ability to detect and apprehend poachers in vast national reserves is critical to the survival of many animal species facing extinction. Recently, emerging computer vision technologies have been applied to the task of poaching detection using imagery from camera traps or conservation drones together with object detection techniques. This review will cover the current uses of computer vision in conservation, with a focus on the identification of poachers through object detection and the specific challenges introduced by aerial data, speed and efficiency considerations. This is followed by an exploration of alternative methods of supervision within the aerial detection context. Finally, we look at the development of self-supervised pseudo-labelling and unsupervised anomaly detection as alternative approaches to detection with reduced annotation cost.

3.2 Computer Vision in Conservation

The application of computer vision in conservation has experienced a recent surge due to rapidly developing computer vision techniques. Specifically, computer vision has been used to augment traditional animal census techniques and wildlife tracking to provide vital information such as species location, population sizes and species interaction necessary for conservation efforts [Wearn and Glover-Kapfer 2017]. Recent work in computer vision for conservation has mainly focused on the use of either camera traps or aerial surveillance data.

3.2.1 Camera Traps

Camera traps combine digital cameras together with passive infrared sensors to sense the presence of wildlife and trigger the camera in the absence of human operation [Wearn and Glover-Kapfer 2017]. Camera traps are a non-invasive method of estimating the diversity, distribution and location of different animal species and thus play a key role in conservation efforts. Recently, computer vision has been introduced into camera trap technology to automatically process the vast amounts of collected data to

detect animal species and even individuals. In a recent work, Schneider *et al.* [2018] compare the ability of the Faster-RCNN and YOLOv2 object detection networks to identify, locate and classify animal species in camera trap images, demonstrating the superiority of the Faster-RCNN network. Beery *et al.* [2019] further build on this work by designing an efficient pipeline to adapt a pretrained animal detection model to a new environment using a limited dataset. Furthermore, Schofield *et al.* [2019] extend camera trap analysis to include automatic re-identification of individual animals using the Single Shot Detector (SSD) architecture. Thus, several key studies have been conducted to overcome the challenges of manually processing increasing amounts of camera trap data, which would present a substantial benefit to the field of conservation.

3.2.2 Unmanned Aerial Vehicles (UAVs)

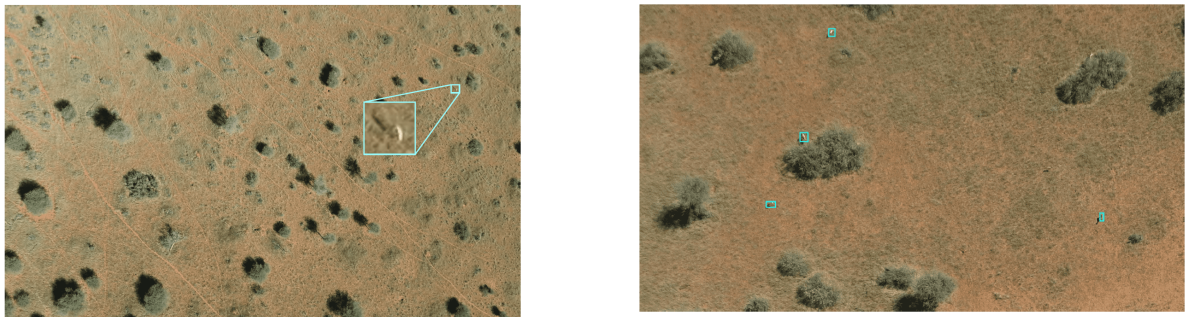
In addition to camera traps, computer vision has been employed alongside Unmanned Aerial Vehicles (UAVs) to perform tasks such as aerial animal population estimation [Kellenberger *et al.* 2018] and wildlife tracking [Bondi *et al.* 2020]. Object detection in aerial imagery presents significantly different challenges to those involved in analysing camera trap images. Recent studies either centre on the analysis of RGB images from visible light cameras or thermal imagery from heat-sensitive infrared cameras.

RGB Imagery

Several studies have focused on animal detection in RGB images obtained from conservation drones [Rey *et al.* 2017; Kellenberger *et al.* 2017 2018; Moreni *et al.* 2021]. A pioneering work by Rey *et al.* [2017] employs the use of an ensemble of exemplar support vector machines (EESVM) to identify large animals in vast reserves. In contrast to later approaches which focus on the use of deep learning, Rey *et al.* [2017] utilise hand-crafted features such as bag-of-visual-words and histogram of colours. The study utilises the SAVMAP dataset for training and analysis, containing an open-source collection of RGB images acquired using a fixed-wing drone in the Kuzikus wildlife reserve in Namibia [Reinhard *et al.* 2015]. Samples from the SAVMAP dataset are included in Fig. 3.1a and Fig. 3.1b, indicating the sparsity and small-scale of animals from the Unmanned Aerial Vehicle (UAV) perspective. The sparsity and small-scale of objects present in the data collected from conservation drones further escalates the complexity and time taken to obtain detailed annotations.

More recent work has employed advances in deep learning to animal detection in aerial imagery. Kellenberger *et al.* [2017] employ a convolutional neural network to improve upon the hand-crafted feature approach utilised by Rey *et al.* [2017]. Through the combination of a pretrained AlexNet backbone and a two-branch strategy for animal recognition and localization, Kellenberger *et al.* [2017] improve significantly upon the previous animal detection results on the SAVMAP dataset. Most notably the resulting model was able to process data in near real-time, processing over 72 images per second in comparison to the 3 images per second achieved by the baseline Faster-RCNN model. Real-time processing is an acute requirement for any practical animal

or poaching detection system [Kamminga *et al.* 2018], and thus any proposed network must meet real-time requirements in order to form a viable solution.



(a) Small scale of animals from aerial viewpoint

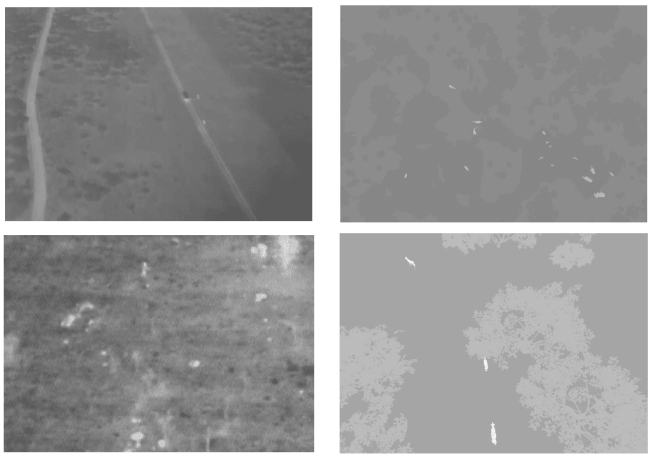
(b) Target objects appear sparsely in vast amounts of data

Figure 3.1: Locating large animals in the Kuzikus wildlife reserve in Namibia [Kellenberger *et al.* 2018].

Thermal (Infrared) Imagery (TIR)

Thermal infrared imagery provides a tremendous advantage over RGB images as it remains effective in the low light and nighttime conditions during which poaching activity experiences an increase [Bondi *et al.* 2018b]. However, thermal imagery introduces the added challenges of low contrast, blurry edges and noise, which can have a detrimental effect on detection performance [Li *et al.* 2020]. Thermal infrared cameras are based on the detection of infrared energy emitted from objects, known as heat signatures, and therefore function well in low-light conditions. A major obstacle to current animal and poaching detection is the manual monitoring of thermal video which requires significant concentration and is prone to error, while also being limited to staff availability in the reserve [Bondi *et al.* 2018b].

While most applications of computer vision to conservation have focused on animal population estimation, recent work has applied similar principles to the task of poaching detection. Most pertinent to this study is the development of the SPOT (Systematic POacher deTector) system by Bondi *et al.* [2018b] which automatically detects poachers in thermal imagery obtained from fixed wing conservation drones. Core advantages realised by the final design include the ability of the system to perform in near real-time as well as the ability of the system to adapt to poor network connectivity [Bondi *et al.* 2018b]. This pioneering study indicates the feasibility and value of applying state-of-the-art object detection methods to the task of poaching detection, leading to the adoption of the system in a reserve in Botswana [Bondi *et al.* 2018b]. Despite these successes, a significant barrier remains in terms of the time and cost involved in annotating the UAV data in order to train the poaching detection network. This is particularly concerning for the long-term practicality of such systems in which models may need to be updated with large amounts of incoming aerial data.



(a) Real (b) Synthetic

Figure 3.2: Real and synthetic thermal infrared imagery from the BIRDSAI dataset [Bondi *et al.* 2020].

A notable, recently released dataset consisting of thermal infrared images from fixed-wing conservation drones, is the Benchmarking IR Dataset for Surveillance with Aerial Intelligence (BIRDSAI) which aims to encourage further research into aerial poacher detection [Bondi *et al.* 2020]. Frames from both the real-world and synthetic thermal infrared videos collected as part of the BIRDSAI dataset are shown in Fig. 3.2 above. Most notably, while developing the poaching detection benchmark, Bondi *et al.* [2017] highlight several acute challenges with respect to annotating the aerial thermal infrared videos. Firstly, the small-scale of the objects from the aerial view-point together with the sparsity of the objects leads to an arduous and time-consuming annotation procedure. This is further exacerbated by multiple image resolutions due to varying drone altitude [Bondi *et al.* 2017]. Furthermore, due to the sensitive nature of poaching detection data, large-scale crowd-sourced annotation tools may not be a viable option in order to uphold data privacy. Thus the study indicates the vital need for improved labelling procedures in this domain. In response to these challenges, Bondi *et al.* [2017] developed VIOLA (VIdeO Labeling Application), aiming to assist labelling of poachers and animals in aerial conservation data through a workload distribution framework. However, since the proposed framework remains reliant upon manual annotators, it still requires significant labour and expertise. Consequently, further developments to improve current annotation processes for aerial conservation data are warranted in order to decrease both labour requirements and overall costs.

3.2.3 Alternative Supervision in Aerial Detection

Current studies investigating alternative methods of supervision for aerial detection remain limited. Notably, only one study has explored the use of weak supervision in the conservation context. Kellenberger *et al.* [2019] employ weak supervision on the RGB images from the Kuzikus dataset, exploring the use of image-level class labels and image-level object counts as contrasting forms of partial annotation. Specifically, Kellenberger *et al.* [2019] utilise a heatmap-based approach with a pretrained ResNet

backbone, and different loss functions appropriate for the image-level and count-based supervision respectively. A notable result found in the study is that the use of only 1% of fully supervised data together with the weakly supervised object count data can match full supervision performance. This result indicates the value of weak supervision for aerial conservation tasks, as effective models can be trained with ground truth that is significantly cheaper and faster to annotate. However, while Kellenberger *et al.* [2019] achieve promising results using count-based weak supervision, performance using only image-level class labels remained poor. Since the annotation cost of image-level counts is much higher than simple binary class labels or no labels, further development is required to significantly reduce manual annotation effort.

A complementary study focusing on the use of self-supervised pre-training was undertaken by Zheng *et al.* [2021], who utilise contrastive learning to pretrain the detection model without the need for annotations. Zheng *et al.* [2021] demonstrate the superiority of networks pretrained using Momentum Contrast (MoCo) over an ImageNet-pretrained model with full supervision, therefore indicating the value of self-supervised pretraining for the conservation task. Nevertheless, the use of self-supervised pre-training is limited to initialisation of the network and therefore cannot aid in continual improvement and updating of the model with incoming data. This limits the benefits of self-supervised pre-training as it cannot harness new batches of unlabelled thermal imagery to continuously update or fine-tune the model in a real-world pipeline. In addition, since both studies focused on animal detection in RGB data, the efficacy of weak and self-supervision techniques has not yet been explored for thermal infrared conservation applications. Visible light cameras are severely limited to daylight conditions. Therefore, for the purposes of poacher detection, operation in low light situations is crucial, thus requiring the use of an additional sensor modality.

Therefore, previous conservation literature has indicated the need for improved methods to harness vast amounts of unlabelled UAV data in order to train effective poaching detection models. Unsupervised and self-supervised techniques pose a unique opportunity to overcome the difficulties involved in annotating aerial conservation data. The next section explores the most recent developments in the unsupervised anomaly detection and self-supervision domains.

3.3 Self-supervised Learning

The fields of unsupervised and self supervised learning are rapidly expanding due to the variety of real-world applications that require cheaper means of data annotation. Self-supervision aims to utilise the inherent properties of unlabelled data to form a supervisory signal and thus learn useful embeddings for further tasks. Central classes of self-supervised techniques include contrastive and non-contrastive learning.

3.3.1 Contrastive Learning

Contrastive learning aims to learn an embedding such that similar images are placed closely together in the embedding space and dissimilar images or negative examples are far apart in the embedding space [Jaiswal *et al.* 2020]. More specifically, an anchor image, positive sample and negative sample are defined where the distance between the anchor-positive pair is minimized while the distance between the anchor-negative pair is maximised. Often a positive sample is created as an augmentation of the original image, as shown in Fig. 3.3.

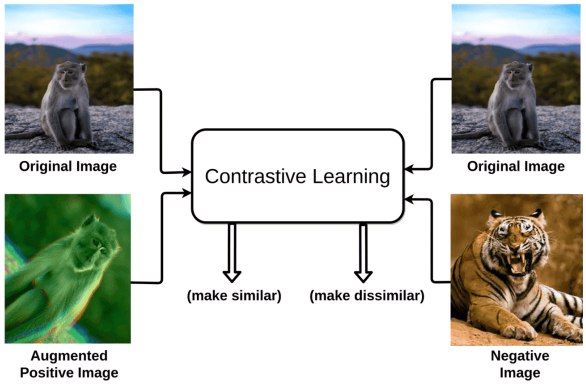


Figure 3.3: In contrastive learning, the distance between the original image and similar, positive images is minimised while the distance between the original image and dissimilar, negative examples is maximised [Jaiswal *et al.* 2020].

Accordingly, the contrastive loss is defined in relation to the chosen similarity metric, which often takes the form of the cosine similarity, defined as

$$cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.1}$$

A popular implementation of the contrastive learning approach, known as Momentum Contrast (MoCo), was proposed by He *et al.* [2020] who introduce a queue structure, allowing for larger, more consistent dictionaries of data samples and better performance on downstream tasks.

3.3.2 Non-contrastive Learning

Further methods eliminate the need for negative pairs, leading to increased resilience against changes in batch sizes and choice of image augmentation [Grill *et al.* 2020]. In particular, Grill *et al.* [2020] introduce Bootstrap Your Own Latent (BYOL) which utilises two interactive neural networks, known as an online and target network respectively. Stemming from this, a recently developed method known as self DIstillation with NO labels (DINO) proposes a simplified self-supervision strategy where a student network directly predicts the output of a teacher network using cross-entropy loss [Caron *et al.* 2021]. Similarly to BYOL, the self-distillation approach employs a teacher and student network with identical model architecture. In the DINO approach, Caron *et al.* [2021], different crops of each input image are fed to the student and teacher networks

respectively. After an additional centering step is applied to the teacher network to prevent mode collapse, the outputs of each network are then normalised using a softmax function [Caron *et al.* 2021]. The overall aim is that the distribution of the student network should match that of the teacher network. Accordingly, the weights of the student network are updated through back propagation with cross-entropy loss, while the weights of the teacher network are updated according to the exponential moving average (EMA) of the student's weights [Caron *et al.* 2021]. The overall process is summarised in Fig. 3.4.

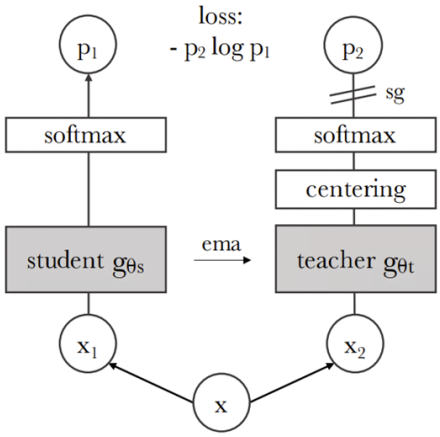


Figure 3.4: The self-supervised method self-DIstillation with NO labels (DINO) uses cross-entropy loss to train a student network to directly predict the output of a teacher network [Caron *et al.* 2021].

3.3.3 Self-supervised Pseudo Labelling

Recently, self-supervised approaches have been harnessed to provide a means of pseudo-labelling in scenarios where manual annotation is infeasible. Most recently, with the popularity of the vision transformer (ViT), the DINO (Self-DIstillation with NO labels) method was implemented using vision transformers for both the teacher and student networks [Caron *et al.* 2021]. This led to crucial results regarding the features of the resulting attention maps from the final layer of the network. As indicated in Fig. 3.5, outputs from the various attention heads focus on the objects of interest in the image, suggesting that the attention maps can be utilised to locate object boundaries in a self-supervised manner.

In this study, we explore the transformation of these self-attention maps into automated pseudo-labels, which can be used in place of time-consuming manual annotations. Localizing Objects with Self-Supervised Transformers and no labels (LOST) [Siméoni *et al.* 2021] used a similar approach for the non-aerial RGB dataset PASCAL VOC. However, we apply the use of Transformer attention maps to the challenges of aerial thermal infrared data and the small-scale objects involved in poaching detection. In the next section we explore recent developments in unsupervised anomaly detection as an alternative method of localising areas of interest in unlabelled data.

Figure 3.5: Attention maps obtained from the final layer of a Vision Transformer (ViT) trained using the self-supervised learning method DINO [Caron *et al.* 2021].

3.4 Unsupervised Anomaly Detection

Unsupervised anomaly detection aims to detect and localise anomalous areas of an image in situations where annotating sufficient anomaly training data is impractical. Current unsupervised anomaly detection approaches generally fall into two categories, namely representation-based and reconstruction-based methods. These classes of unsupervised anomaly detection are further explored below, along with their strengths and limitations.

3.4.1 Reconstruction-based Methods

Reconstruction-based methods employ the use of generative models such as Generative Adversarial Networks (GANs) and autoencoders (AEs) to reconstruct normal images, while anomalous regions cannot be reconstructed as they do not appear in the training samples.

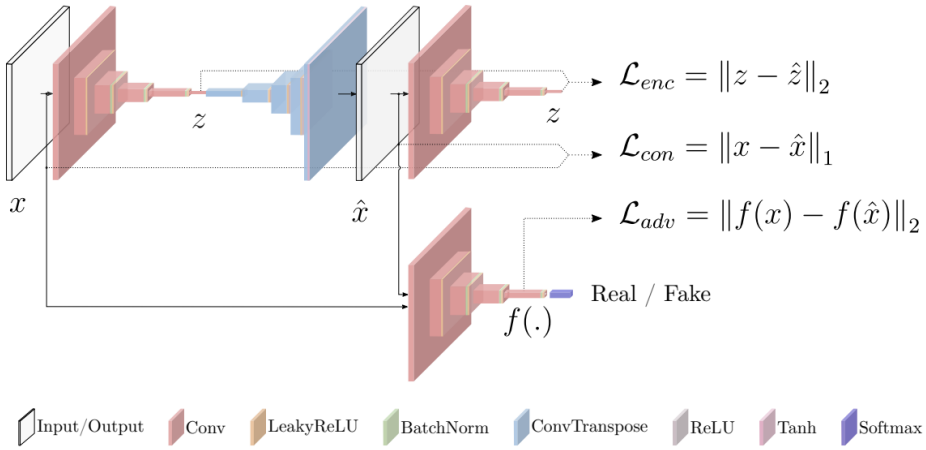


Figure 3.6: The architecture and loss functions used in the reconstruction-based GANomaly anomaly detection method [Akcay *et al.* 2018].

Anomalies can then be detected by performing a pixel-by-pixel comparison of the input and its corresponding reconstructed image [Bergmann *et al.* 2019]. Fig. 3.6 shows the architecture and loss functions used to train the reconstruction-based GANomaly model [Akcay *et al.* 2018], where the encoder loss L_{enc} is used to calculate the final anomaly score. This core principle has been adapted in a number of different ways [Bergmann *et al.* 2018; Akçay *et al.* 2019; Zavrtanik *et al.* 2021], and applied predominantly in the aerospace and medical domains [Baur *et al.* 2019]. These domains share a similar requirement to the task of poaching detection where the amount of annotation effort for detailed medical and aviation data is impractical, necessitating the use of unsupervised techniques.

3.4.2 Representation-based Methods

In contrast, traditional representation-based approaches involve using a deep convolutional neural network to extract features of a set of normal images X and estimate their distribution p_X using non-parametric methods [Yu *et al.* 2021]. This enables the model to detect out-of-distribution images or regions $\bar{X}$, i.e. images or regions which have a small likelihood given p_X and are therefore anomalous. This approach is indicated by the encoder-decoder structure of the CFlow architecture shown in Fig. 3.7. Significant milestones made using this approach include PaDiM [Defard *et al.* 2021], CFlow [Gudovskiy *et al.* 2022] and FastFlow [Yu *et al.* 2021]. As demonstrated in Fig. 3.7, the output heatmap is able to identify anomalous regions of the image without being trained with localisation information.

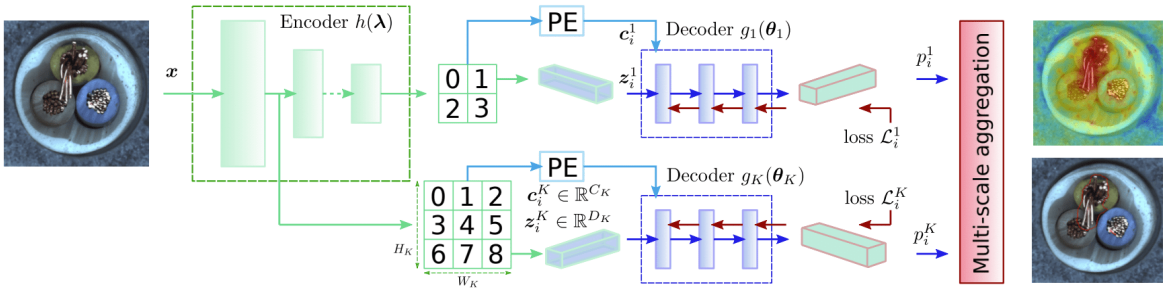


Figure 3.7: Architectural diagram depicting the representation-based unsupervised anomaly detection algorithm CFlow [Gudovskiy *et al.* 2022]. The CFlow model comprises an encoder and decoder for feature extraction and distribution estimation respectively.

3.4.3 Speed and Efficiency

The CFlow [Gudovskiy *et al.* 2022] and FastFlow [Yu *et al.* 2021] models are of particular interest to this study due to their superior speed and efficiency. A significant challenge to current unsupervised anomaly detection techniques is ensuring efficiency and speed are maintained to provide a realistic solution to real-world problems. While the majority of unsupervised anomaly detection techniques struggle to meet real-time requirements, notable strides have been made in terms of inference speed and overall algorithm efficiency. This notable change was pioneered by the CFlow model which

introduced conditional normalizing flows, allowing the resulting model to be significantly faster and more compact in size than prior state-of-the-art models [Gudovskiy *et al.* 2022]. Building on this, FastFlow extend the use of normalizing flow to two-dimensional space [Yu *et al.* 2021]. Notably, FastFlow eliminates the need for the sliding window method commonly used in unsupervised anomaly detection and instead allows for end-to-end inference of the image as a whole, greatly improving inference speed and efficiency. Furthermore, the FastFlow model utilises a very lightweight network structure, further improving its efficiency [Yu *et al.* 2021]. Consequently, these recent developments in unsupervised anomaly detection algorithms indicate their potential to be used either as a means of pseudo-labelling or as the main detector onboard the computationally constrained UAV.

3.5 Conclusion

Current literature regarding the use of computer vision for conservation points to the urgent need for the ability to train accurate poaching detection networks using cheaper annotations. This is particularly significant for real-world systems which require a means of updating current models with new data or where annotating large amounts of incoming data with detailed ground truth may be prohibitively expensive or time-consuming. Moreover, the small-scale and sparsity of the targets found in aerial poaching data together with the low contrast of thermal-infrared imagery renders annotation particularly challenging, motivating the need for the development of an efficient and less costly means of annotation and training.

While pioneering research into the use of alternative levels of supervision in aerial conservation tasks has been conducted, this has been limited to studies using RGB data only. Furthermore, investigations into weak supervision using cheaper image-level class labels have yielded poor performance, with adequate performance only being achieved using more expensive image-level counts. In addition, research into self-supervision in aerial conservation has been limited to the effects of pre-training only. Therefore further exploration into the use of self-supervision is necessary to ensure that large amounts of incoming data can be harnessed in a continual manner, allowing for maintenance of existing models and more accessible training of new models.

Furthermore, recent developments such as the self-distillation method DINO indicate the potential of self-supervision and Vision Transformers to locate image boundaries without manual annotation. This presents a novel opportunity to utilise the resulting attention maps to generate pseudo-labels for poaching detection data. Similarly, the recent advances in unsupervised anomaly detection suggest that utilising these algorithms directly as detectors, or to provide pseudo-labels, may provide an effective solution to decrease annotation expense for poaching tasks. Accordingly, unsupervised and self-supervised techniques present a unique opportunity to decrease the annotation cost associated with labelling aerial, thermal-infrared data, thus providing a more realistic and effective poaching detection system.

Chapter 4

Research Methodology

The next section describes and motivates the chosen research framework, methods and implementation process. Firstly, the research design is outlined, together with the chosen methods and dataset. This is followed by an exploration of the dataset, data preparation, creation of the pseudo-labelling pipelines and training and evaluation of the final detection network.

4.1 Investigative Framework

4.1.1 Research Design

The project followed a hypothesis-led structure, aiming to prove or disprove the stated research hypothesis. Accordingly, two pipelines were constructed to produce pseudo-labels from the self-supervised method DINO and an unsupervised anomaly detection network respectively. The pseudo-labelled data was then used to train a YOLOv5 object detection model in a fully supervised manner, for each set of pseudo-labels respectively. The resulting networks were evaluated on real-life thermal imagery collected from drone flights by the local conservation organisation, SPOTS. The proposed approaches were evaluated according to their detection accuracy, speed and overall model size in order to take real-time and efficiency requirements into account. In addition, the performance of the unsupervised and self supervised approaches was benchmarked against that of a fully supervised equivalent.

4.1.2 Dataset

To provide an authentic representation of object detection accuracy in a practical, aerial poaching detection context, real-world data from a local wildlife organisation was used for training and analysis.

SPOTS

The dataset provided by the local conservation organisation SPOTS (Strategic Protection of Threatened Species) contains a variety of environments, flight altitudes and objects, thus providing a realistic representation of the conditions a real-world poaching

detection system would encounter. The dataset contains individual frames of thermal infrared data taken using a fixed-wing drone in wildlife reserves in Southern Africa. The dataset was collected by the SPOTS in collaboration with FruitPunch AI, a challenge-based AI education platform which focuses on solving problems related to the sustainable development goals [FruitPunch AI 2022]. The resulting images are not fixed in dimension. The dataset contains real-world thermal imagery taken from various drone flights at different times, amounting to over 40GB of data. Notably, ground truth annotations detailing object location and species are present for each frame, allowing for the training of the fully supervised baseline. The dataset was labelled by a local company specialising in data annotation, using a combination of manual labelling and annotation models.

4.1.3 Model Selection

In order to select a suitable approach to decreasing annotation cost for the task of poaching detection, the size and speed of the model were crucial considerations. This is particularly pertinent as poaching detection requires real-time processing of the thermal infrared video stream, as well as an efficient architecture to allow the model to run onboard the UAV. Moreover, power efficiency is a core concern for conservation drone tasks as flight time is directly impacted if the battery power is drained too quickly. Current methods of alternative supervision vary in terms of both efficiency and speed, promoting careful choice of technique for applications running on edge hardware.

Selection of Self-supervised and Unsupervised Techniques

Accordingly, the field of weakly supervised object detection was deemed less suitable for the task as most weakly supervised object detection networks require pre-computed object proposals using methods such as EdgeBoxes [Zitnick and Dollár 2014] or Selective Search [Uijlings *et al.* 2013] and thus cannot meet the real-time requirements of the poaching detection system. Similarly, the self-supervised learning method DINO would be too large to run onboard the UAV to provide direct predictions. However, the potential of the DINO model as a means of pseudo-labelling unlabelled data could then allow for training a lightweight detection model to run onboard the UAV, therefore utilising the benefits of self-supervision but maintaining efficiency and speed.

From an unsupervised perspective, unsupervised anomaly detection algorithms could be of benefit to the poaching detection task as either a means of pseudo-labelling unlabelled data or by running the algorithm directly onboard the UAV to generate real-time predictions. In this study we have focused on the first use case in order to provide a suitable comparator to the DINO method. However, the chosen unsupervised anomaly detection algorithm was selected due to its speed and efficiency. This not only allows for an efficient pseudo-labelling procedure but also retains the potential for the model to be used as a direct detector. In order to select the unsupervised anomaly detection technique, representation-based approaches were considered in favour of reconstruction-based approaches due to their relatively simpler complexity. Despite this, many previously developed representation-based anomaly detection algorithms have not been

able to meet real-time requirements, as discussed in the literature. Therefore, only certain unsupervised anomaly detection networks have the potential to meet the speed and efficiency requirements of poaching detection. Thus, the FastFlow algorithm was selected due to its superior speed and efficiency in comparison to other unsupervised anomaly detection networks, allowing for investigation into its use as a means of providing either direct predictions or pseudo labels.

Selection of Detection Network

In order to select an appropriate detection network for training and evaluation, considerable weight was given to the practical realities of running a detection algorithm onboard the conservation UAV. Stemming from previous experience and experimentation undertaken by the SPOTS conservation organisation [SPOTS 2022] in partnership with FruitPunch AI [FruitPunch AI 2022], the YOLOv5-small (YOLOv5s) [Jocher 2020] model is a suitable choice for the drone hardware. Several motivating factors include the smaller model size, which allows the model to run on an NVIDIA Jetson Nano without overheating, as well as the ease of converting the YOLOv5 model to TensorRT, which allows for further optimization and speed up. While smaller models such as the YOLOv5-nano are available, detection accuracy is decreased. Thus, the YOLOv5-small model provides a better trade-off of accuracy, speed and model size.

4.1.4 Performance Analysis

To provide an accurate representation of model performance on the test datasets, several differing metrics were measured to evaluate each network, namely precision (P), recall (R), and mean Average Precision (mAP).

Mean Average Precision (mAP)

The Average Precision (AP) is one of the most common metrics utilised in literature to compare object detection performance. The Average Precision (AP) measures the detection accuracy by finding the average of the ratios of true positive detections to all positive detections, across all recall values [Kisantal *et al.* 2019]. Closely linked to the Average Precision is the concept of the Intersection over Union (IoU). The Intersection over Union (IoU) is a measure of the overlapping area between the predicted bounding box and true bounding box of an object in the image divided by the union of the two bounding boxes.

$$IoU = \frac{bb_{predicted} \cap bb_{true}}{bb_{predicted} \cup bb_{true}} \tag{4.1}$$

The IoU gives a quantitative score for how close the predicted bounding box is to the true bounding box. The model then classifies the prediction as correct if this IoU is above a certain threshold. The Average Precision (AP) is often reported at a variety of IoU threshold values in order to help determine the best choice of threshold. In this study, the mean Average Precision (mAP) averaged over all object categories is reported at an IoU threshold of 0.5.

Recall

In terms of the poaching detection task the most notable metric is recall (R), the proportion of targets that were identified, as it is most crucial to correctly recognise poaching detection threats. Thus for this use case, a low rate of false negatives is desirable. Hence recall will be taken to be the most significant measure in each evaluation. It should be noted that all metrics will be reported at the confidence threshold which maximises the $F1$ score and thus at the optimal balance between precision and recall. Since each bounding box prediction is reported with an associated confidence, the confidence threshold determines which predictions will be scored as true predictions and which will be discarded. Any candidate bounding boxes with confidence lower than the confidence threshold are discarded and therefore not included in the performance metric calculations. While a lower confidence threshold allows for higher recall and a higher confidence threshold results in higher precision, a balance between the two is required for sensible detection performance. The $F1$ score can be used as a metric for balancing recall and precision as it is defined as the harmonic mean of recall and precision. Therefore the confidence threshold is set at the value which maximises the $F1$ score, and thus maximises this balance.

Frames Per Second (FPS)

Lastly, in order to incorporate practical considerations such as detection speed, model speed was evaluated according to frames per second (FPS) in order to provide an indication of whether the model would be of practical use in a real-time application. Frames per second (FPS) gives an indication of the number of images that can be processed and classified by the detection model per second. Moreover, the resulting model size in MB was recorded in order to provide an indication of model efficiency for deployment on the computationally constrained UAV. Therefore both the FPS and model size were used as the measure of whether the models would be of practical use in the live poaching detection scenario.

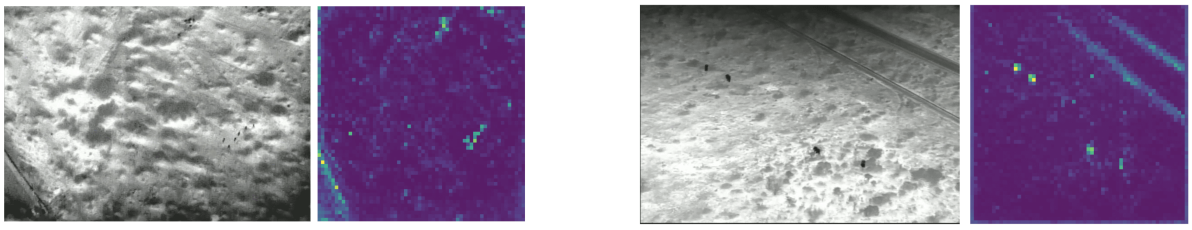
4.1.5 Plan of Development

The project is partitioned into two major phases, focusing on the use of self-supervised and unsupervised techniques respectively. An overview of each is provided below.

Phase One: Generating pseudo-labels from attention maps and self-supervision

In the first phase, the self-supervised learning method DINO (Self-DIstillation with NO Labels) [Caron *et al.* 2021] was utilised to provide pseudo-labels for the unlabelled poaching detection data. To accomplish this, attention maps were extracted from the final layer of the DINO network, indicating the parts of the image drawing the most focus from the network. Notably, this approach includes an assumption that the target objects contrast in some way with the background of the image. Since thermal imaging renders the objects of interest in black against grey and white imagery, the use of DINO attention maps seems appropriate for the task at hand. While previous studies have

shown the ability of the attention maps to focus on large, predominant objects in the frame, initial experimentation was performed to verify whether a similar effect could be achieved for small-scale objects.



(a) Small-scale objects of interest are identified

(b) Attention maps could provide a basis for pseudo-labels

Figure 4.1: Initial experiments using the DINO model to obtain attention maps for SPOTS data indicate the potential of the proposed pseudo-labelling methodology.

As indicated by Fig. 4.1, the attention maps were able to focus on small objects of interest from an aerial perspective. This encouraging result further motivated the potential to create pseudo-labels for the poaching detection task using the DINO model. Therefore, the first phase of the project focused on a self-supervised approach where the DINO framework is used to extract pseudo-labels for the dataset, without the need for tedious or manual annotation.

Phase Two: Generating pseudo-labels from unsupervised anomaly detection and segmentation maps

In the second phase of the investigation, unsupervised techniques were explored as an alternative to self-supervised approaches. Accordingly, an unsupervised anomaly detection network was selected and trained, with the resulting segmentation maps being utilised to generate pseudo-labels. To accomplish this, the anomaly detection algorithm was first trained on background-only images from the SPOTS dataset with no localization information. The trained anomaly detection network was then used to calculate segmentation maps for the SPOTS data, and the resulting segmentation maps were then converted to pseudo-labels in a similar manner to the DINO-based pseudo-labels. This approach allows the poaching detection network to train with annotations that are easier and faster to obtain as detailed bounding boxes are not required to train the unsupervised anomaly detection network.

Evaluation and Comparison

The resulting pseudo-labels from both techniques were then used as ground truth labels to train a YOLOv5-small (YOLOv5s) model in a fully supervised manner. In addition, the performance of the YOLOv5 model trained on the pseudo-labelled data was compared to that of the same model trained on the original ground truth as a baseline comparator. All selected techniques were tested using the same pipeline, poaching detection dataset and data preparation to ensure a fair comparison.

In the following sections, each of these phases is presented in more depth, detailing the characteristics and associated challenges of the dataset as well as the mechanisms of each pseudo-labelling pipeline.

4.2 Data Exploration and Preparation

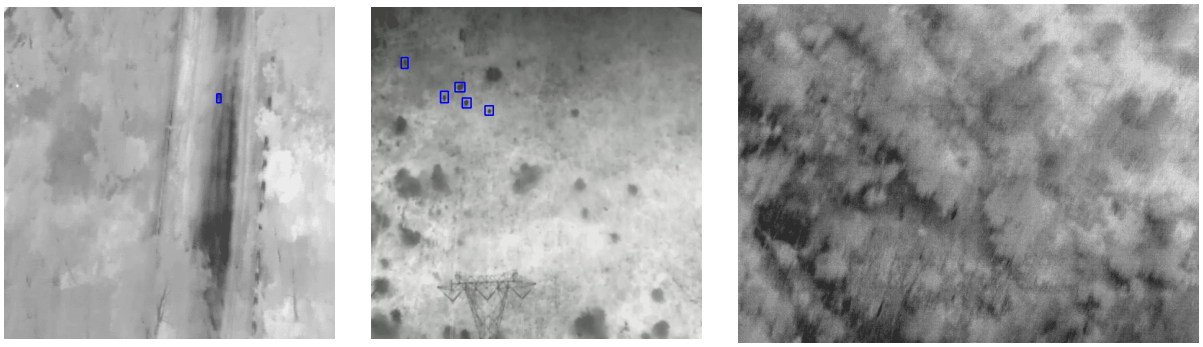
4.2.1 Dataset Composition

Dataset Structure

The full SPOTS dataset spans over 40GB of aerial video footage, comprised of 12 thermal infrared videos split into individual frames. Each video corresponds to a single drone flight spanning a particular terrain. Due to this, the terrains present in the dataset are quite varied, ranging from open savanna to dense bushland. Due to the format and size of the dataset, considerable processing was required to create a smaller dataset suitable for training and analysis. Most significantly, since the data contained video footage with high correlations between frames, the videos were sampled in two contrasting ways in order to form suitable train, validation and test sets.

Image Characteristics and Challenges

Through visualisation of several example images in the SPOTS dataset, the difficulties involved in the aerial poaching detection task can be better understood. As indicated in Fig. 4.2a, thermal infrared imagery poses the challenges of low contrast and low resolution. Moreover, as indicated by the bounding boxes in Fig. 4.2b, the objects requiring detection appear on a very small scale to the human eye. This increases the difficulty of the object detection task while simultaneously further motivating the need for automatic monitoring of thermal infrared drone footage. Furthermore, the sample images indicate the variation in drone altitude that can occur during a flight, causing objects to appear at different sizes in each frame. This is coupled with a non-uniform image size, each adding to the complexity of the object detection task. Lastly, Fig. 4.2c shows evidence of motion blur due to drone speed.



(a) Low contrast and low resolution. (b) Small-scale targets. (c) Motion blur.

Figure 4.2: Sample images from the SPOTS dataset [FruitPunch AI 2022] indicating the diverse challenges associated with aerial thermal infrared imagery.

Class Imbalance

In addition to an analysis of the visual characteristics of the dataset, further insights into the associated annotations can be gathered. The SPOTS data contains two major data sources, Dataset V2 and Dataset V3, each containing several videos with differing characteristics and annotation formats. To understand the prevalence of each object class in the data sources, the class distribution of each source is visualised in Fig. 4.3.

(a) Dataset v2.

(b) Dataset v3.

Figure 4.3: Proportion of frames containing background vs object classes for data source v2 and data source v3 respectively.

As indicated in Fig. 4.3a, Dataset V2 contains a relatively even balance between background-only frames and frames containing poachers. In contrast, Fig. 4.3b shows the dominance of background-only images in Dataset V3 in comparison to objects identified in the human, animal and vehicle categories. For the purposes of this study, frames containing only vehicles are categorised as background-only frames, with both animal and human objects being considered target objects for detection. Accordingly, the ratio of background-only frames to frames containing target objects when both data sources are combined is indicated in Fig. 4.4.

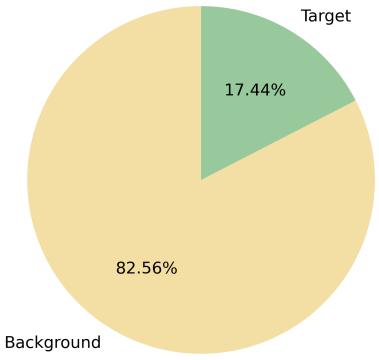


Figure 4.4: Proportion of frames containing background vs target classes after collation of the two data sources. A clear class imbalance is observed in favour of background-only images.

Notably, it can be observed from Fig. 4.4 that there is a severe class imbalance present in the raw data, with 82.56% of frames containing background only while only 17.44% of the frames in the dataset contain target objects. While this is representative of a real-world scenario, it leads to under-representation of the targets of the poaching detection system. Thus, specific methods to handle this class imbalance are required.

4.2.2 Dataset Curation

In order to curate a suitable dataset for training and evaluation, the two data sources were collated and standardised, followed by sampling to reduce the effects of high correlation between frames. Their respective annotations were then reformatted into a common form and converted to YOLO annotations. Lastly, the resulting dataset was split into train, validation and test splits in two different manners. This process is summarised in Fig. 4.5 below.

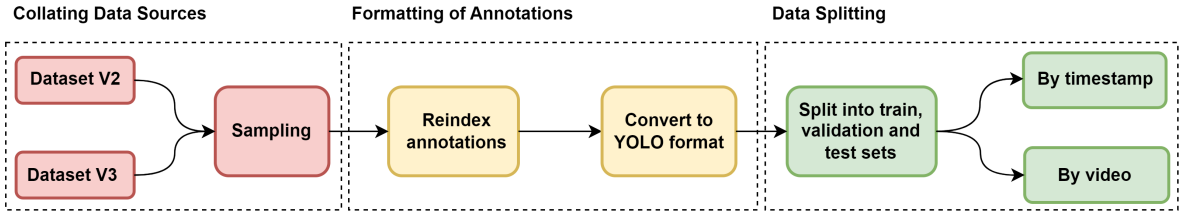


Figure 4.5: Pipeline for collating raw SPOTS data sources V2 and V3 into a suitable dataset for analysis. The pipeline includes three major phases: collation and sampling, formatting of annotations and data splitting.

Collation and Sampling

In order to combine the two data sources and reduce the similarity between adjacent frames, each of the thermal infrared videos from the two data sources was sampled at a fixed interval. Additionally, frames containing targets were sampled at a higher rate than background frames in order to reduce the large class imbalance in favour of the background-only images.

Formatting of Annotations

Two different formatting, naming and labelling schemes were present in the annotations of the SPOTS data, necessitating a collation process to standardize the annotations and convert them into YOLO annotation format. Moreover, due to the different classes present in each data source, all annotations were reindexed with all human and animal objects referenced as *class 0* (target object) and all frames with only vehicles classed as background-only frames with no annotations. The reindexed annotations were then converted to YOLO format, specified as $< class, x_{cen}, y_{cen}, w, h >$ in normalized form, where each image containing targets corresponds to a matching *.txt* file.

Data Splitting

Due to the nature of the dataset as thermal infrared video streams converted to individual frames, very high correlations were present between the frames in the original data sources. Therefore two different sampling methods were utilised to create the train, validation and test sets. Firstly, each video was split by timestamp into three timespans from which the train, validation and test sets were then sampled. This approach aimed to reduce data leakage while still following a more traditional machine learning approach to splitting the dataset. Secondly, in order to evaluate the generalisation abilities of each detection model and provide a more realistic experimental framework, the data was split into train, validation and test sets by video or flight.

To split by timeframe, each video was first separated into three sections according to timestamp. The respective sets were then sampled from only a particular section of the video. In contrast, to split by video, certain videos were designated for training, validation or testing. The train, validation and test sets were then sampled from the specified videos accordingly. This ensured that certain videos were completely unseen to the model during evaluation. These differences are summarised in the diagram in Fig. 4.6 below, portraying an example with six videos.

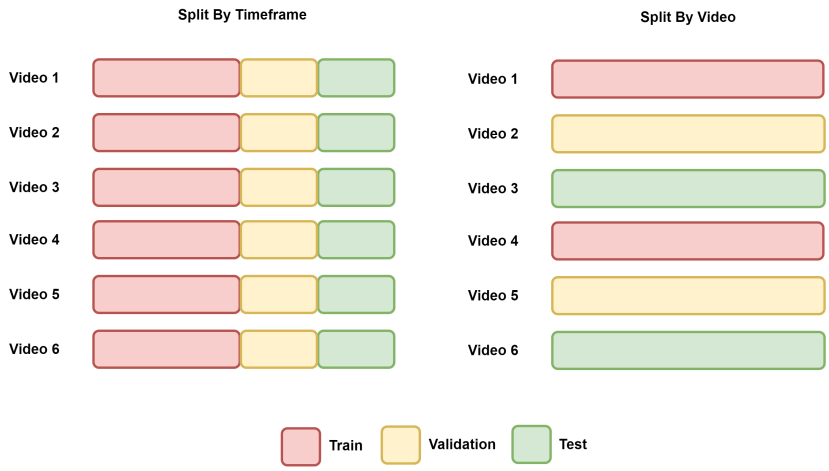


Figure 4.6: Visualisation of the two sampling techniques used to create the training (red), validation (orange) and test (green) sets.

4.3 Self-supervised Learning: DINO

Following dataset preparation, an investigation into the optimal use of the self-supervised DINO model in the conservation context was undertaken.

4.3.1 Development of the Pseudo-labelling Pipeline

Firstly, a pseudo-labelling pipeline was developed to convert the attention maps from DINO to pseudo-labels for the SPOTS dataset. Each pseudo-labelling pipeline involved

the application of the respective unsupervised or self-supervised technique to the input image, followed by a conversion from a heatmap to a binary mask from which bounding boxes were detected. Finally, post-processing steps such as filtering of noise and artefacts were applied to generate the final pseudo labels. A summary of the pseudo-labelling pipeline using the self-attention maps generated from DINO is shown in Fig. 4.7.

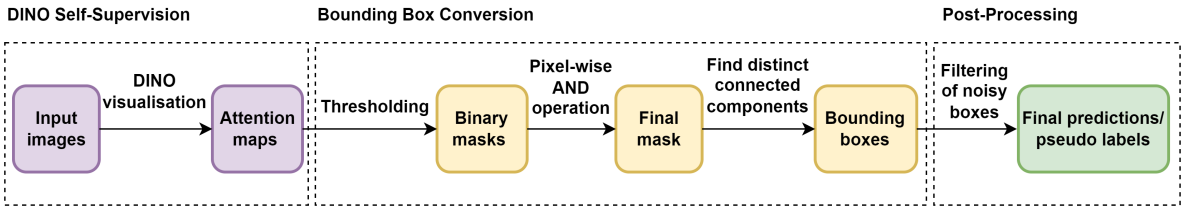
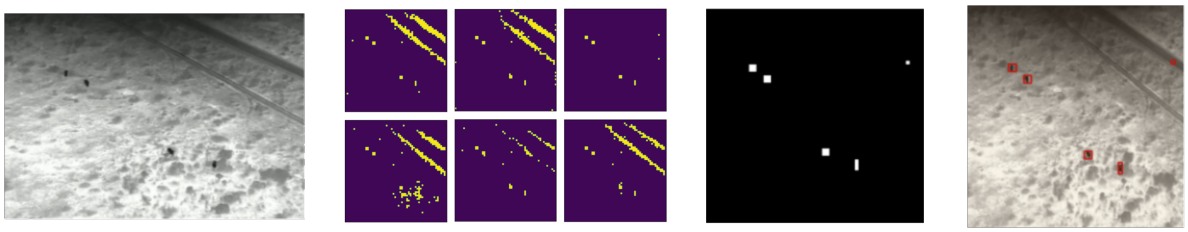


Figure 4.7: Pipeline for transforming DINO self-attention maps into pseudo labels used for object detection. The pipeline includes three major phases: generation of attention maps from DINO, bounding box conversion and post-processing and refinement.

Generating Attention Maps using DINO

Firstly, batches of images were processed using DINO *DeiT-S* [Touvron *et al.* 2021] with patch size 8 to visualise the self-attention maps. The six attention heads from the underlying Transformer model produced six corresponding attention maps each highlighting slightly different features. This is demonstrated in Fig. 4.8b.



(a) Input image. (b) Attention maps. (c) Combined mask. (d) Final pseudo label.

Figure 4.8: Example of the progression from the input thermal image, thresholded attention maps from the six attention heads, final combined mask and resulting bounding box pseudo-labels.

Converting Attention Maps to Binary Masks

The attention heatmaps generated by each of the six attention heads of the Transformer model were then thresholded to form binary masks, where white pixels above the threshold indicate possible objects or areas of interests. Careful threshold selection was pivotal as a lower threshold increases the ability of the pseudo labels to recall objects of interests but simultaneously increases the noise of the resulting pseudo labels. A range of thresholds from 60-100 were evaluated with a threshold of 80 being utilised unless otherwise stated. As indicated in Fig. 4.9, while a threshold of 60 was found to result in noisy pseudo-labels, a threshold of 100 resulted in the target object

being missed altogether. Therefore a threshold of 80 allowed for identification of target objects while preventing noisy labels which would obscure effective training of the YOLOv5 detection network. This threshold value would need to be tuned for a given dataset and could be determined visually in a similar manner.

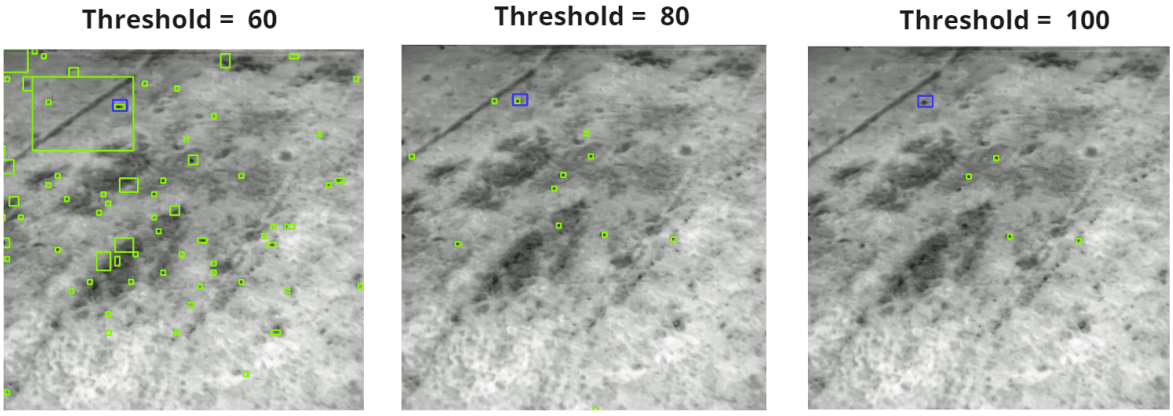


Figure 4.9: Effects of the threshold used when converting attention maps to binary masks. A threshold that is too low results in noisy labels which obscure correct identification of targets while thresholds that are too high omit real targets of interest.

Combining Multiple Attention Maps

In order to combine the insights from the six resulting masks, the masks were passed through a pixel-wise AND operation to generate the final combined mask. This method of combination was chosen to select out the areas of the image that all the attention heads focused on and therefore were the most likely to belong to objects of interest in the image. An example of the final combined mask is shown in Fig. 4.8c.

Extracting Bounding Boxes from Binary Masks

Lastly, bounding boxes were extracted from the combined mask by finding distinct connected components of adjacent white pixels using *scikit-image*. The resulting bounding boxes were then converted to YOLO annotation format and written to individual *.txt* files. A visualisation of the extracted bounding boxes superimposed on the original input image is included in Fig. 4.8d. Several post-processing steps were considered to refine the resulting bounding boxes and are discussed further in the next section.

4.3.2 Investigating Post-Processing Modules

Several different strategies were considered to refine the bounding boxes yielded by the pseudo-labelling pipeline. Notably, any post-processing technique needed to be universal in order to improve the quality of all the pseudo-labels across all included environments. Therefore care was taken to ensure the pipeline remained as general as possible to be able to cater for new data, environments and use cases.

Filtering

Firstly, filtering of very large and very small bounding boxes was considered. The filtering of large boxes could provide benefits by removing the identification of large background structures as objects of interests, such as segments of road. However, since the altitude of the conservation drone is variable, this could result in larger valid target objects being omitted. Due to the importance of identifying all targets, filtering of large objects was not included in the final pipeline. In contrast, filtering of abnormally small bounding boxes helped to remove noise from the resulting labels. Therefore filtering these boxes as a post-processing step resulted in better quality pseudo-labels.

Padding

Secondly, the mechanism of the pseudo-labelling procedure results in the final bounding boxes fitting very closely around the objects of interest. While this is a desirable quality, it can cause a discrepancy between the manually annotated labels and the pseudo-labels, with possible effects on later evaluation scores. Thus, the width and height of each bounding box was padded slightly to allow the pseudo-labels to more closely match that of the ground truth and therefore produce a better training signal for the object detection model.

In order to select the relevant hyperparameters, a range of filtering thresholds between 0.10-0.020 and padding sizes between 0-0.02 were evaluated. The effects of each hyperparameter choice is further explored in Section 5.3.2 in the results.

4.4 Unsupervised Learning: FastFlow

The second phase of the investigation centred on the use of unsupervised anomaly detection as an alternative to the self-supervised DINO model.

4.4.1 Training the Unsupervised Anomaly Detection Model

While the FastFlow pseudo-labelling pipeline follows similar stages to the DINO pipeline, a major difference between the two approaches includes the training or fine-tuning of the FastFlow network. Unlike the DINO pipeline which relied solely upon the pre-trained weights for the *DeiT-S* network, the FastFlow network was fine-tuned on the normal images of the training set before calculating the anomaly detection heatmaps used for the pseudo-labels.

Anomalib

Anomalib is an unsupervised anomaly detection library written in PyTorch that includes an extensive model zoo covering several state-of-the-art unsupervised anomaly detection networks [Akcay *et al.* 2022]. The Anomalib library was used to facilitate the training of the FastFlow algorithm and subsequent segmentation map calculations for the pseudo-labelling pipeline.

FastFlow

As discussed in the literature, FastFlow makes use of 2D normalizing flow together with a lightweight network structure, leading to state of the art accuracy and speed [Yu *et al.* 2021]. An overview of the FastFlow architecture is included in Fig. 4.10. The Anomalib library contains an implementation of the FastFlow algorithm, and can be utilised to provide segmentation results rather than just classification of the image as normal or anomalous. Since the pseudo-labels require localization information, the segmentation task was specified during FastFlow training. For all experiments, ResNet-18 was utilised as the feature extractor.

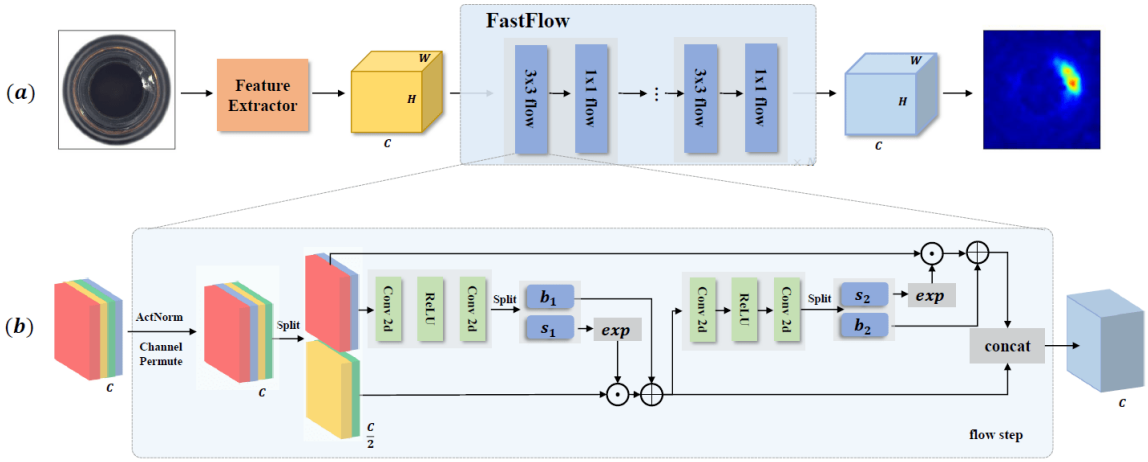


Figure 4.10: Architecture of the FastFlow anomaly detection algorithm [Yu *et al.* 2021]. The FastFlow model incorporates a two-dimensional loss function together with alternating large and small convolutional kernels, forming a lightweight structure. The resulting model can be used in combination with a variety of suitable feature extractors [Yu *et al.* 2021].

Data Preparation

In the poaching detection context, we consider normal images to be those without any target objects, while anomalous images contain such objects. Thus, since the FastFlow algorithm is trained on normal images only, the training set was split into background-only images and images containing targets. Background-only images are used during training in order to extract features for the modelling of the normal image distribution. Following training, passing normal or anomalous images through the FastFlow algorithm outputs a heatmap highlighting potential anomalous areas. This heatmap can then be transformed into a binary segmentation mask using thresholding.

FastFlow Training

In order to fine-tune the FastFlow network, the model was trained for 20 epochs using the background-only images from the training set. The model was then validated on both the anomalous and normal images from the validation set. Thus, it should be noted that this approach does require some manual manipulation to split the training

set into normal background-only images and abnormal images containing targets. In addition, a few images may be hand-annotated with localisation information in order to calculate additional validation metrics that measure the accuracy of the segmentation, such as the pixel-wise F1 score, however this is not required for the functioning of the pipeline.

4.4.2 Development of the Pseudo-Labelling Pipeline

Similarly to the DINO pipeline, the unsupervised FastFlow pipeline followed a progression from anomaly detection heatmap through to the final pseudo labels. However, several modules were tailored to the specific characteristics of the FastFlow segmentation heatmaps. An overview is included in Fig. 4.11.

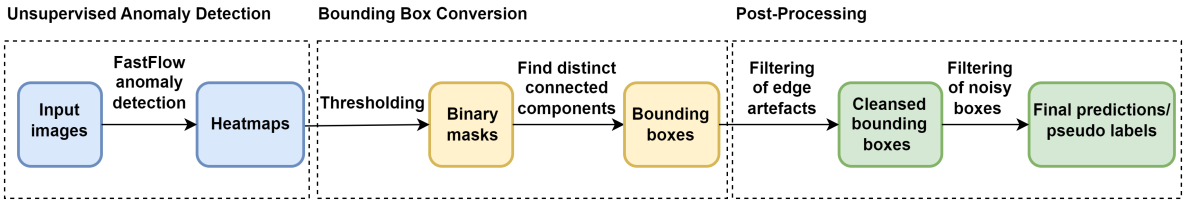
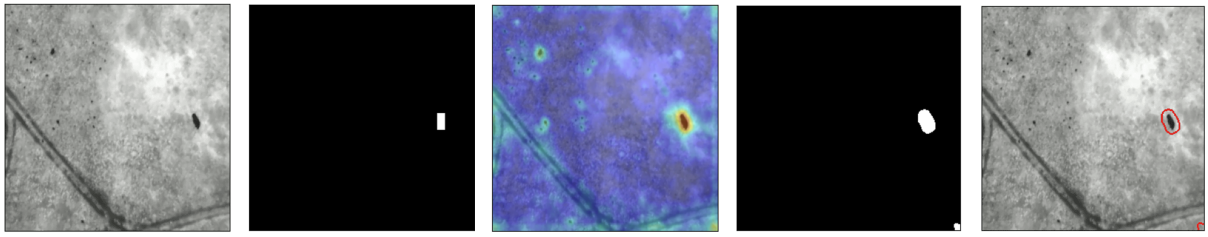


Figure 4.11: Pipeline for transforming FastFlow anomaly detection predictions into pseudo labels used for object detection. The pipeline includes three major phases: unsupervised anomaly detection, bounding box conversion and post-processing and refinement.

Converting Segmentation Maps to Binary Masks

During FastFlow training and validation, the threshold at which the heatmap (Fig. 4.12c) is transformed into the corresponding segmentation mask (Fig. 4.12d) is automatically calculated with respect to the validation set. However, in order to select out only the most important targets in the image, 0.1 was added to the auto-calculated threshold. This resulted in better identification of targets of interests and reduced noise in the final pseudo-labels.



(a) Input image. (b) Ground truth mask. (c) FastFlow heatmap. (d) FastFlow mask. (e) Segmentation result.

Figure 4.12: Comparison of the input image, ground truth mask, predicted heat map from FastFlow, and resulting predicted mask and detections. The mask obtained through unsupervised anomaly detection closely resembles that of the manually annotated ground truth.

Filtering of Edge Artefacts

A unique characteristic of the segmentation masks from the FastFlow algorithm is the presence of artefacts around the edges and corners of the frame. The FastFlow algorithm often detects edges as anomalous regions, resulting in edge artefacts and thus incorrect bounding boxes. To alleviate this, all bounding boxes connected to an edge were filtered, resulting in cleaner pseudo-labels. Fig. 4.13 indicates both the need for and results following the filtering of edge artefacts.

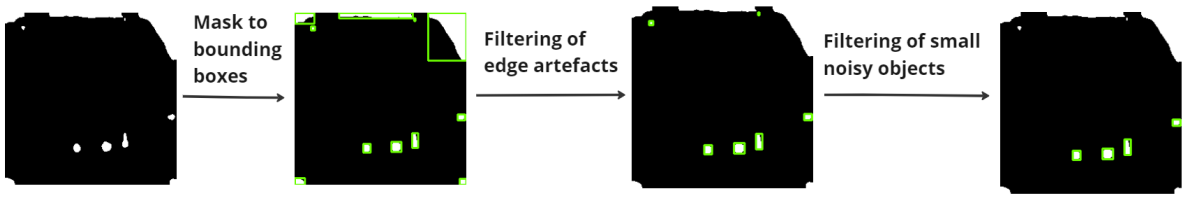


Figure 4.13: Progression and refinement of the FastFlow segmentation mask through each pipeline stage. The need for the filtering of edge artefacts is clearly shown by the incorrect detections at the edges and corners of the frame.

Post-Processing

As in the case of the DINO pipeline, the last stage included the filtering of very small noisy boxes and padding of all remaining boxes. Specifically, following hyperparameter tuning, bounding boxes with a width or height smaller than the chosen filtering threshold were filtered out in order to avoid large numbers of noisy predictions. Lastly, padding was implemented by adding a small amount to the width and height of each bounding box. It should be noted that the chosen filtering threshold and padding amount were standardized across both the DINO and FastFlow pipelines for each experiment, to ensure fairness of the post-processing effects. A full analysis of the effects of varying the post-processing hyperparameters is included in subsection 5.3.2.

4.5 Evaluation of Pseudo-Labels

In order to gauge the accuracy of the pseudo-labels during the pipeline development process, two different methods of evaluation were utilised to compare the pseudo-labels against the original ground truth bounding boxes.

4.5.1 Intersection-over-Union (IOU)

Firstly, the intersection-over-union (IOU) between the ground truth masks and pseudo label masks was calculated on an image-level. This gave an estimate of how closely the given pseudo-label matched its ground truth counterpart. Furthermore, the training images were clustered according to IOU bracket to visualise which images resulted in good quality pseudo-labels and what type of images were more challenging for the pipeline. For example, images that performed well with $IOU > 0.5$ included frames

with clearly defined and distinct target objects while frames that performed poorly with $IOU < 0.1$ often contained background-only images where pseudo-labels picked up false positives. The image-level IOU also aided in decisions regarding hyperparameters such as thresholds, maximum size of objects to be filtered out and the effects of padding discussed in the previous section.

4.5.2 Bounding Box Visualisation

Secondly, in order to further understand the characteristics, strengths and weaknesses of the generated pseudo-labels, the pseudo-labels from each pipeline were visualised and compared. The pseudo-label bounding boxes were plotted on the original input images and compared to the ground truth bounding boxes to verify the legitimacy of the pseudo labels. A subset of the resulting visualisations are showcased in section 5.1 in the results chapter.

4.6 Training the YOLOv5 Detection Model

Once the pseudo-labels were generated for the train and validation datasets, a YOLOv5s model was fine-tuned for 50 epochs using the pseudo-labels from DINO and FastFlow respectively. The resulting models were then evaluated on a separate test set and compared to the original ground truth test set labels. In order to provide a baseline for comparison the YOLOv5s detection network was trained in a fully supervised manner using the original ground truth labels for the SPOTS dataset and evaluated on the same test set. All of the images input to the pseudo-labelling pipelines and subsequent YOLOv5 training were resized to 640×640.

Hyperparameter	**Value**
Batch Size	32
Base Learning Rate	0.01
No. of Epochs	50
Image Size	640

Table 4.1: Selected hyperparameters while training the YOLOv5s network on the SPOTS training dataset.

The YOLOv5-small object detection network was utilised for the training and evaluation of all investigated techniques with a custom *YAML* configuration file for the SPOTS dataset. The majority of the experimental work performed was undertaken in a Google Colab environment in order to ensure access to sufficient computing power such as Graphics Processing Units (GPUs). All models were initialised using the same weights from a YOLOv5s model pre-trained on the ImageNet dataset. In addition, data augmentation techniques including random flips and resizing were applied to the input data during training for all networks.

4.7 Evaluation on SPOTS Dataset

Evaluation of the proposed techniques was performed by considering both the quality of the pseudo-label in isolation as well as the results of the YOLOv5 model when trained on the pseudo-labels.

4.7.1 Qualitative

As discussed above, in order to determine the quality of the pseudo-labels generated during development of the pipeline, the resulting bounding boxes were overlayed on the original images together with the ground truth bounding boxes. Similarly, following the training of the YOLOv5 detection model, the resulting predictions on the test set were filtered according to the confidence threshold and then plotted on the original input images together with the ground truth bounding boxes. This allows us to determine whether the predictions made by the networks trained on the pseudo-labels are viable and useful for the task of poaching detection. An analysis of the resulting predictions from a qualitative perspective is included in subsection 5.2.3.

4.7.2 Quantitative

Furthermore, following fine-tuning of the YOLOv5 detection network on each set of labels, each model was then quantitatively evaluated on two different test sets. Notably, two different test sets were utilised in order to measure the performance of the pseudo-labels under different conditions. A major distinction was drawn between the majority of frames containing natural, bush-land or open field environments and a minority group of images containing buildings. These environmental differences were observed to play a major role in the performance of the models. Thus to quantify this difference, two test sets were constructed, one including a large number of frames containing cluttered scenes and buildings, and the other with a reduced number of complex frames. Approximately 30% of the former set comprised of complex frames, while the latter set contained less than 1% of frames with buildings. This test set structure was repeated for both the split-by-timestamp experiments and the split-by-video experiments, where the test sets were sampled as detailed in subsection 4.2.2 and then the number of complex building frames was reduced accordingly to form the second test set in each case.

In addition, in order to investigate whether the generated pseudo-labels can be combined with a small portion of ground truth labels to improve performance, two different models were trained for each type of pseudo-label. Firstly the YOLOv5 model was fine-tuned with the full set of pseudo labels to ascertain the model performance trained on pseudo-labels alone. Then the YOLOv5 model was trained using the same set of pseudo labels but with ground truth labels substituted in for the complex building frames only. In practice this could allow the majority of videos to be annotated using the pseudo-labelling pipeline with only the most difficult training videos requiring manual annotation. Once again this approach was followed for both the split-by-timestamp and split-by-video data splits. An average of 10 runs was taken for each experiment.

Chapter 5

Results

The results for each experiment are now presented in order to evaluate the efficacy of the pseudo-labelling procedures from both a qualitative and quantitative perspective.

5.1 Quality of Pseudo-Labels

Firstly, the accuracy of the pseudo-labels is assessed from a qualitative perspective by comparing the ground truth and pseudo-label segmentation masks and corresponding bounding boxes. The ground truth bounding boxes are depicted in blue, while the DINO and FastFlow boxes are indicated in red and green respectively. Since the complete pseudo-label set cannot be represented, three subsets displaying the core characteristics of the pseudo-labels have been selected to portray the major trends observed in the fuller dataset. Firstly, Fig. 5.1 showcases a set of frames where the pseudo-labels perform well in a diversity of conditions. Secondly, Fig. 5.2 displays common shortcomings such as missed targets or grouping multiple targets into one bounding box. Lastly, Fig. 5.3 indicates the difference between the FastFlow and DINO pseudo-labels in terms of the main cause of their respective false positives.

5.1.1 Self-supervised: DINO

From the visualised bounding boxes and masks shown in Fig. 5.1, we can observe the ability of the DINO attention maps and resulting DINO pseudo-labels (red) to correctly identify objects of interest in the frame. Furthermore, Fig. 5.1 encompasses a wide variety of different environments in which the pseudo-labelling pipeline functions effectively with the notable exception of the final frame, where the pseudo-label deviates greatly from the ground truth. Thus, a significant distinction can be noticed in the quality of the pseudo-labels for natural environments in comparison to frames containing cluttered building scenes. This trend will be discussed further in later sections.

In addition, from Figures 5.2 and 5.3 we can observe several trends and shortcomings in the DINO pseudo-label set. In some cases a focus on distracting background features causes missed targets or additional false positives. While representing the full extent of the differences between the DINO and FastFlow pseudo-labels is intractable,

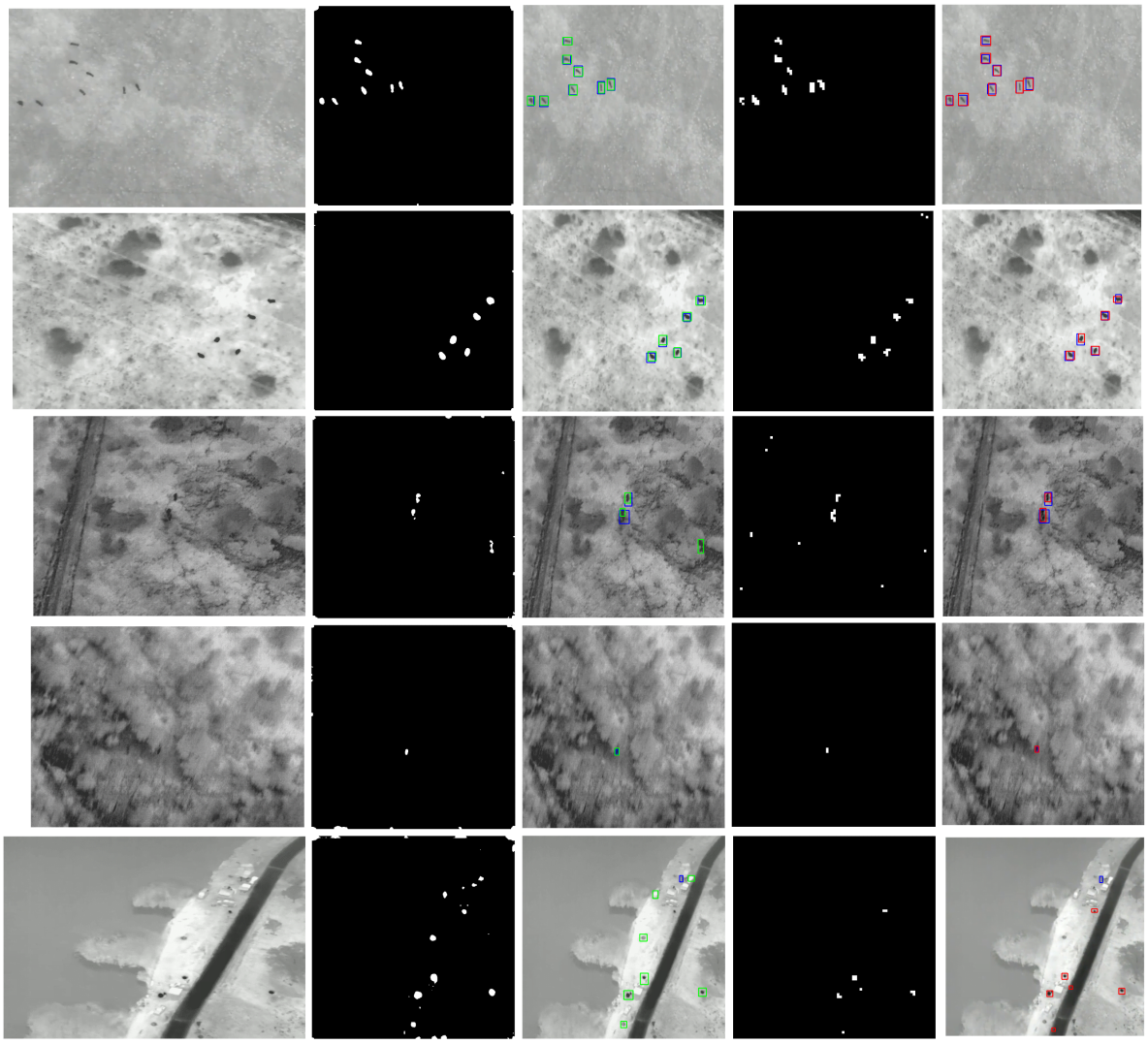
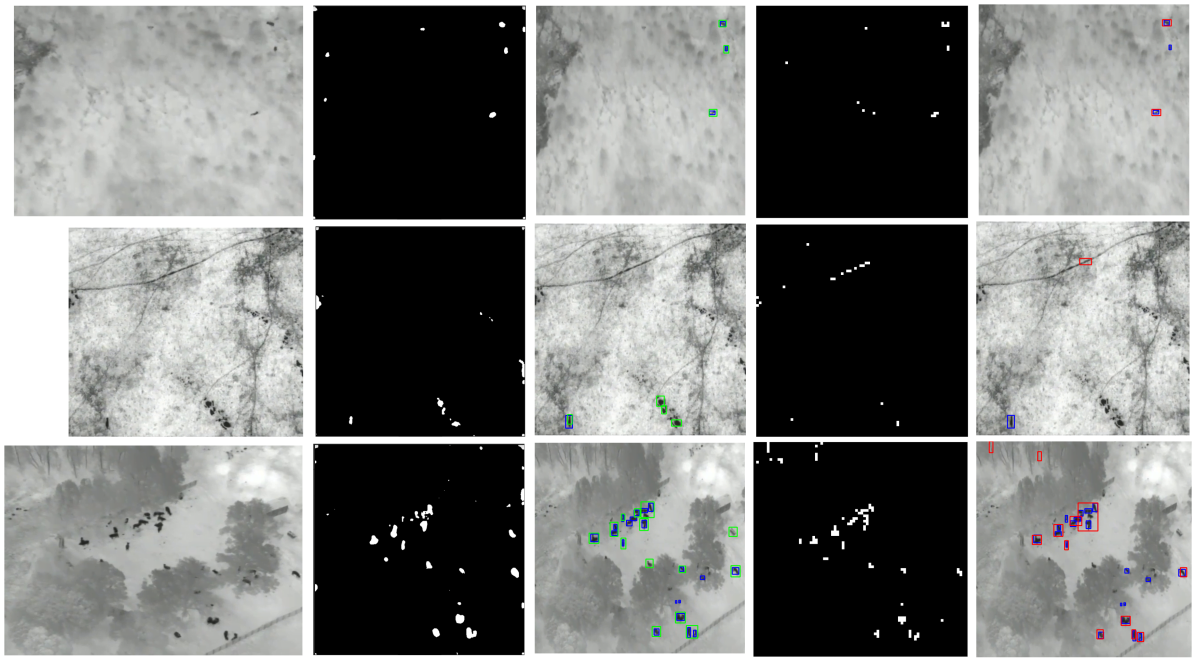


Figure 5.1: Qualitative comparison of the segmentation masks from FastFlow and DINO, and the resulting pseudo-labels for FastFlow (green) and DINO (red) compared against the ground truth bounding boxes (blue). It can be observed that the pseudo-label pipelines are able to detect and localise objects in a range of different natural environments, from open plains to dense forest terrain (frames 1-4). In contrast, both pipelines perform poorly for cluttered building environments (frame 5).

a notable trend includes the tendency of the DINO pseudo-labels to focus on strong background features such as roads, fences or lines. This results in false positives in these regions while often leading to missed targets within the same frame. This phenomenon is shown most clearly in Fig. 5.3 where the DINO segmentation masks clearly highlight lines and roads as areas of interest in the image, and thus detract from the true targets in some cases. Despite these weaknesses, the majority of the DINO-based pseudo-labels retain a strong ability to identify and localise the targets of interest with well-fitted bounding boxes.

5.1.2 Unsupervised: FastFlow

Furthermore, from Fig. 5.1, we can observe that the FastFlow pseudo labels (green) are also able to correctly detect and enclose single and multiple targets. The resulting pseudo-labelled bounding boxes closely resemble ground truth boxes in most frames. Once again a notable exception is observed for the cluttered building scene in frame 5 where several distracting objects are highlighted while real target objects are missed.



(a) Input image. (b) FastFlow mask. (c) GT (blue) vs FastFlow (green). (d) DINO mask. (e) GT (blue) vs DINO (red).

Figure 5.2: Qualitative comparison of original input image, segmentation masks from FastFlow and DINO, and the resulting pseudo-labels from FastFlow (green) and DINO (red) compared against the ground truth bounding boxes (blue). Common shortcomings include missing targets (frame 1), false positives (frame 2) and labelling multiple objects as one target (frame 3).

As can be observed in Figures 5.2 and 5.3, while the FastFlow pseudo-labels also include false positives, this does not stem from a misplaced focus on large background structures. Instead, the FastFlow pseudo-labels often label environmental elements such as rocks which bear a very similar resemblance to target objects. This is showcased in both Fig. 5.2 (frame 2) and Fig. 5.3 (frame 3).

A source of error common to both the FastFlow and DINO pseudo-labels occurs in cases where multiple objects are marked as a singular object or some targets remain undetected. This is common in frames with targets clustered closely together such as in Fig. 5.2 (frame 3). Despite the observed shortcomings, the resulting pseudo-labels are largely able to correctly identify and localise the targets in each frame, having received little to no supervision.

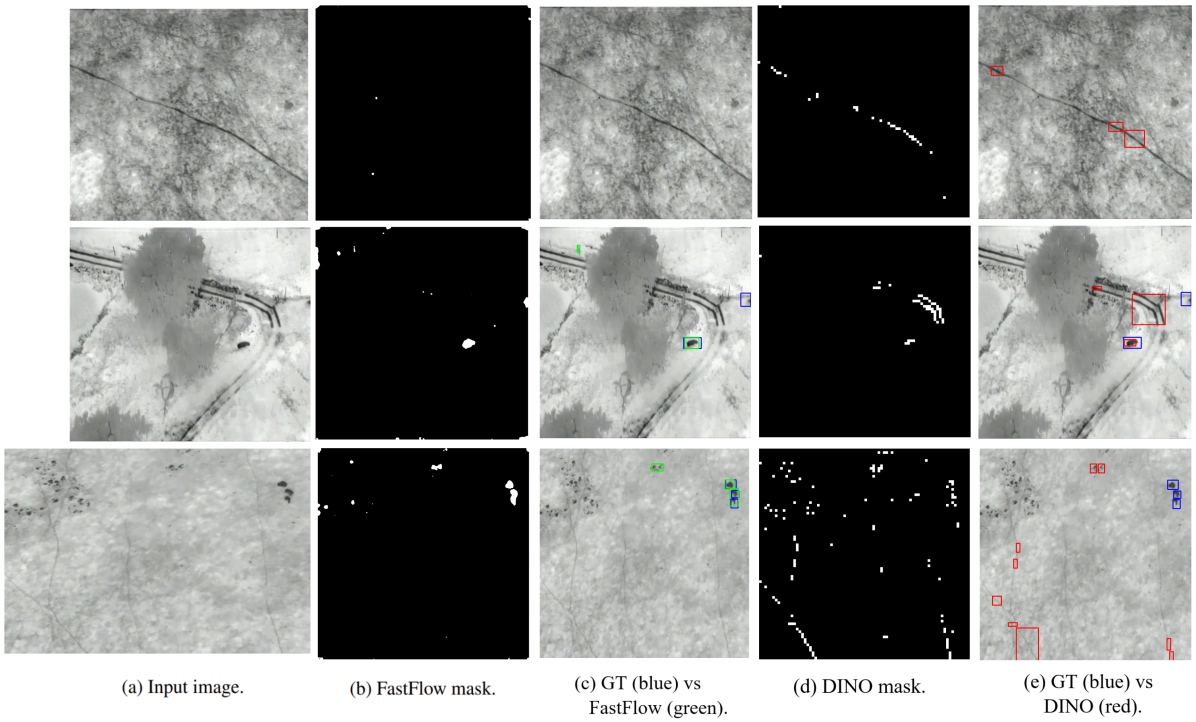


Figure 5.3: Qualitative comparison of the segmentation masks from FastFlow and DINO, and the resulting pseudo-labels from FastFlow (green) and DINO (red) compared against the ground truth bounding boxes (blue). It can be observed that while FastFlow (green) indicates a strong ability to correctly label target objects, DINO (red) is often misled by other dominant items in the frames such as roads, lines and rivers.

5.2 Poaching Detection Performance

Secondly, in order to determine how the differences in the respective pseudo-labels affect the resulting performance of the detection network, the predictions from each trained model were examined both qualitatively and quantitatively. Following fine-tuning of the YOLOv5 using the ground truth labels and pseudo-labels respectively, the precision, recall and mean Average Precision (mAP) were measured in order to evaluate the resulting performance of the poaching detection model. It should be noted that the precision, recall and mAP values are reported at the confidence threshold which maximises the $F1$ score for the model.

5.2.1 Split By Timestamp

Firstly, we consider the set of experiments where the dataset was split into train, validation and test sets by sampling according to timestamp. This formed the initial experimentation for each pipeline, with a post-processing filtering threshold of 0.020 and a padding amount of 0.005.

Self-supervised

The first set of experiments involved training the YOLOv5 detection model on the DINO-based pseudo-labels and evaluating the model on the split-by-timestamp test set, with

and without cluttered building images. This included both the model trained on the full DINO pseudo-label set *(DINO All)* and the model trained on the DINO pseudo-labels with certain ground truth labels substituted in for frames containing buildings *(DINO + GT for Buildings)*. The results are included in Tables 5.1 and 5.2 respectively.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	**0.484**	**0.438**	**0.344**
DINO All	0.179	0.149	0.0665
DINO + GT for Buildings	0.261	0.175	0.11

Table 5.1: Detection results for the models trained with DINO pseudo-labels, evaluated on the split-by-timestamp test set with $\approx$ 30% of frames containing buildings. A noticeable increase in performance is observed by substituting ground truth labels into the pseudo-label set for difficult building scenes.

As indicated by the results in Table 5.1, the YOLOv5 model trained on the full set of DINO pseudo labels obtains fairly poor performance in comparison to the baseline model trained on the ground truth labels. A noticeable improvement across all metrics is observed by substituting the ground truth labels in for the frames including buildings during training. This affirms that the noisy pseudo labels generated for the complex building frames negatively affect the overall performance of the YOLOv5 detection model. Therefore to counteract this, ground truth labels were substituted in for only the select few complex frames, leading to an observed increase in performance. Therefore the careful selection of better quality pseudo-labels together with certain ground truth labels can have a significant impact on the final results.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	**0.511**	**0.473**	**0.368**
DINO All	0.284	0.166	0.108
DINO + GT for Buildings	0.327	0.182	0.13

Table 5.2: Detection results for the models trained with DINO pseudo-labels, evaluated on the split-by-timestamp test set with $<$ 1% of frames containing buildings. All models exhibit increased performance when evaluated on a test set with reduced building scenes.

Moreover, from Table 5.2 we observe improved performance across all metrics when the models are evaluated on the test set with the majority of cluttered building scenes removed. This underlines the difference in performance observed between most natural environments and complex scenes containing buildings. This is particularly noticeable by the change in precision and mAP values between Tables 5.1 and 5.2. Furthermore, the performance of the ground truth-trained model also experiences an increase, suggesting that even the baseline model performs worse on complex building scenes.

Unsupervised

The same experimental set up was then repeated for the FastFlow-based pseudo-labels, with the first model trained on the full set *(FF All)* and the second model trained on the

FastFlow pseudo-labels together with ground truth for the building scenes *(FF + GT for Buildings)*. As indicated by the results in Table 5.3, the models perform better than the DINO equivalents but still do not approach the accuracy of the baseline model. Notably, the YOLOv5 model trained on the FastFlow pseudo labels show a similar improvement in performance following the substitution of frames containing buildings for ground truth in the training set. Therefore the FastFlow results further support the ability to boost performance by combining high quality pseudo labels together with a low number of carefully chosen ground truth labels.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	**0.484**	**0.438**	**0.344**
FF All	0.223	0.252	0.102
FF + GT for Buildings	0.298	0.282	0.187

Table 5.3: Detection results for the models trained with FastFlow pseudo-labels, evaluated on the split-by-timestamp test set with a large amount ($\approx$ 30%) of frames containing buildings. Similarly to the DINO-trained models, performance increases when ground truth is substituted in for complex building frames.

Once more, as shown in Table 5.4, when tested on the set with less than 1% of building frames, the overall performance of the models improves greatly. The effects of adding, removing and substituting the pseudo labels for the difficult frames follows a similar pattern to a lesser extent as expected but overall all metrics show considerable improvement. This suggests that the FastFlow pseudo-labelling technique could be used successfully on ordinary, natural terrains which make up the majority of poaching data, while rarer video footage containing buildings or other obstructions could be manually labelled, thus greatly reducing annotation costs.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	**0.511**	**0.473**	**0.368**
FF All	0.391	0.285	0.207
FF + GT for Buildings	0.389	0.297	0.23

Table 5.4: Detection results for the models trained on the FastFlow pseudo-labels, evaluated on the split-by-timestamp test set with a small amount ($<$ 1%) of frames containing buildings. As expected, the improvement in detection performance gained by substituting in certain ground truth labels is lessened when evaluated on a test set with reduced buildings.

Cumulative Results

The cumulative results for the split-by-timestamp experimentation are included below, indicating the level of supervision required for the labels, the label set used to train the YOLOv5 network and the corresponding performance metrics. From Table 5.5 we can observe that during this phase of initial experimentation, the models trained on the pseudo-labels were able to successfully perform detection but were only able to reach 60-80% of the performance of the ground truth-trained model in the best case.

Furthermore, in this phase of experimentation, the FastFlow-trained model outperforms the DINO-trained model. This may be due to the tendency of the DINO attention maps to focus on environmental features such as roads, rivers and lines. Lastly, as exhibited in Table 5.6, all models indicated a significant improvement in detection performance for the test set without building frames, suggesting that certain environments are more difficult for the detection model.

Supervision	**Labels**	**P**	**R**	**mAP**$_{@0.5}$
Full	Ground Truth	**0.484**	**0.438**	**0.344**
Self	DINO All	0.179	0.149	0.0665
	DINO + GT for Buildings	0.261	0.175	0.11
Unsupervised	FastFlow All	0.223	0.252	0.102
	FastFlow + GT for Buildings	0.298	0.282	0.187

Table 5.5: Final quantitative results obtained on the split-by-timestamp SPOTS test set, with $\approx$ 30% of frames containing buildings. The model closest in performance to the baseline is the model trained on a combination of FastFlow pseudo-labels and ground truth labels for difficult building scenes.

Supervision	**Labels**	**P**	**R**	**mAP**$_{@0.5}$
Full	Ground Truth	**0.511**	**0.473**	**0.368**
Self	DINO All	0.284	0.166	0.108
	DINO + GT for Buildings	0.327	0.182	0.13
Unsupervised	FastFlow All	0.391	0.285	0.207
	FastFlow + GT for Buildings	0.389	0.297	0.23

Table 5.6: Final quantitative results obtained on the split-by-timestamp SPOTS test set, with $<$ 1% of frames containing buildings. All models show a significant improvement in performance when evaluated on a test set with reduced building frames.

5.2.2 Split By Video

In order to determine the generalisation abilities of the networks trained with pseudo-labelled data, the dataset was then partitioned by designating certain flight videos to the train, validation and test sets respectively. Notably, this provides a much more realistic evaluation of the networks as in practice the poaching detection model would be trained on a set of past videos and then required to perform on an entirely new flight. The corresponding pseudo-labels were generated using a post-processing filtering threshold of 0.015 and a padding amount of 0.01.

It can be observed from Tables 5.7 and 5.8 that the detection performance for the network trained on ground-truth only is lower when the data is split by video. This is as expected since the split by video framework provides a more realistic measure of detection performance and is closer to how training would work in practice. In contrast, the ground-truth only results from the split-by-timeframe experiments may be

artificially inflated due to the similar characteristics and temporal correlation between frames from the same video. Therefore the split by video framework also provides a better indication of the network's generalisation abilities as the test videos and their specific characteristics have not yet been encountered by the network.

Self-supervised

The performance of the models trained with DINO pseudo-labels for the split-by-video training set is shown in Tables 5.7 and 5.8. It can be observed that while the performance of the ground truth-trained model has decreased, the performance of the DINO-trained models have improved significantly. A particularly notable result is that the model trained using the DINO pseudo-labels alone *(DINO All)* is now able to reach greater than 90% of the baseline precision and recall. It can be noted that substituting in a small portion of ground truth for the most difficult building frames *(DINO + GT for Buildings)* further improves the results, especially with respect to the precision and mAP values, thus indicating a reduction in false positives.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	0.298	**0.313**	**0.190**
DINO All	0.284	0.293	0.154
DINO + GT for Buildings	**0.315**	0.299	0.185

Table 5.7: Detection results for the models trained with DINO pseudo-labels, evaluated on the split-by-video test set with $\approx$ 30% of frames containing buildings. Notably, the model trained on the DINO pseudo-labels alone is able to attain approximately 90% of baseline precision and recall.

Remarkably, on the test set devoid of complex building frames (Table 5.8), the model trained on the DINO pseudo-labels alone *(DINO All)* outperforms the model trained on the ground truth labels, across all metrics. This promising result suggests that training a poacher detection model with auto-generated DINO pseudo-labels can match or even outperform baseline performance in certain environments.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	0.322	0.325	0.212
DINO All	0.372	**0.378**	0.236
DINO + GT for Buildings	**0.372**	0.370	**0.239**

Table 5.8: Detection results for the models trained with DINO pseudo-labels, evaluated on the split-by-video test set with $<$1% of frames containing buildings. A remarkable result is observed on the test set without buildings, as the model trained only on DINO pseudo-labels is able to match baseline performance.

Therefore the results for the self-supervised method DINO indicate a very promising ability to match the recall of the baseline network, and thus support pseudo-labels as a viable technique to decrease annotation cost.

Unsupervised

Lastly, the detection model was trained on the FastFlow pseudo-labels using the split-by-video data splits. The detection performance evaluated on the full test set and test set without building frames is shown in Tables 5.9 and 5.10 respectively. From both tables we can observe that the models trained with the FastFlow pseudo-labels are also able to achieve recall values close to that of the baseline model.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	**0.298**	**0.313**	**0.190**
FF All	0.213	0.288	0.110
FF + GT for Buildings	0.287	0.299	0.167

Table 5.9: Detection results for the models trained on the FastFlow pseudo-labels, evaluated on the split-by-video test set with a large amount ($\approx$ 30%) of frames containing buildings. The models trained with FastFlow-based pseudo-labels are also able to attain close to baseline recall but exhibit lower precision and mAP values than the DINO equivalents.

In comparison to the DINO-trained models, the inclusion of certain ground truth labels during training has a much greater impact on the performance of the FastFlow models when tested on the full test set containing difficult building frames. Therefore the inclusion of ground truth for building frames is even more valuable to improve precision and mAP in the FastFlow case, as indicated in Table 5.9.

Labels	**P**	**R**	**mAP**$_{@0.5}$
Ground Truth	0.322	0.325	0.212
FF All	**0.355**	0.366	0.223
FF + GT for Buildings	0.342	**0.381**	**0.226**

Table 5.10: Detection results for the models trained on the FastFlow pseudo-labels, evaluated on the split-by-video test set with a small amount ($<$ 1%) of frames containing buildings. The FastFlow-trained models no longer exhibit significantly lower precision and mAP than DINO when evaluated on the test set without buildings.

In contrast, when evaluated on the test set without building scenes, the improvement gained by substituting in the ground truth labels is negligible, as shown in Table 5.10. This suggests that the lower precision and mAP values observed in Table 5.9 are linked to poorer performance of the FastFlow models on building scenes specifically. On inspection, a possible reason for this could be that the FastFlow pseudo-labels contain more false positives than the DINO pseudo-labels for images containing buildings. This leads to similar false positives in the corresponding predictions for building scenes, and thus lower precision and mAP values on test sets including these frames. Overall, in a similar manner to the DINO pseudo-labels, the FastFlow-trained models are shown to achieve remarkable performance on the test set without buildings, matching and outperforming the baseline across all metrics.

Cumulative Results

The final results for all split-by-video experiments are summarised in Tables 5.11 and 5.12. Notably, when evaluated on the full split-by-video test set, all pseudo label-based models are able to achieve over 90% of the recall of the baseline ground truth model. This is particularly significant since recall is the most important metric for poaching detection. The use of selective ground truth labels can also be leveraged to further improve the results, particularly in the case of the FastFlow pseudo-labels.

Supervision	**Labels**	**P**	**R**	**mAP**$_{@0.5}$
Full	Ground Truth	0.298	**0.313**	**0.190**
Self	DINO All	0.284	0.293	0.154
	DINO + GT for Buildings	**0.315**	0.299	0.185
Unsupervised	FastFlow All	0.213	0.288	0.110
	FastFlow + GT for Buildings	0.287	0.299	0.167

Table 5.11: Final quantitative results obtained on the split-by-video SPOTs test set, with $\approx$ 30% of frames containing buildings. The models trained with pseudo-labels are shown to reach $\approx$ 90% of the recall of the model trained with ground truth, with the DINO-based models achieving significantly better precision and mAP scores than the FastFlow-based models when evaluated on the full test set.

Moreover, as indicated in Table 5.12, when the building scenes are removed from the test set, both pseudo label-based models are able to match and outperform the baseline across all metrics, when trained on pseudo-labels alone. Thus, for a diversity of natural scenes, very promising performance can be attained using only pseudo-labels with no manual annotations.

Supervision	**Labels**	**P**	**R**	**mAP**$_{@0.5}$
Full	Ground Truth	0.322	0.325	0.212
Self	DINO All	0.372	0.378	0.236
	DINO + GT for Buildings	**0.372**	0.370	**0.239**
Unsupervised	FastFlow All	0.355	0.366	0.223
	FastFlow + GT for Buildings	0.342	**0.381**	0.226

Table 5.12: Final quantitative results obtained on the split-by-video SPOTs test set, with $<$ 1% of frames containing buildings. Notably, both models trained with pseudo-labels alone are able to match and even outperform the detection performance of the ground truth-trained model across all metrics when evaluated on the test set with reduced building frames.

Overall, the split-by-video experiments indicate very promising results, with both pseudo label-trained models able to achieve more than 90% of the recall of the ground-truth model.

5.2.3 Qualitative Analysis

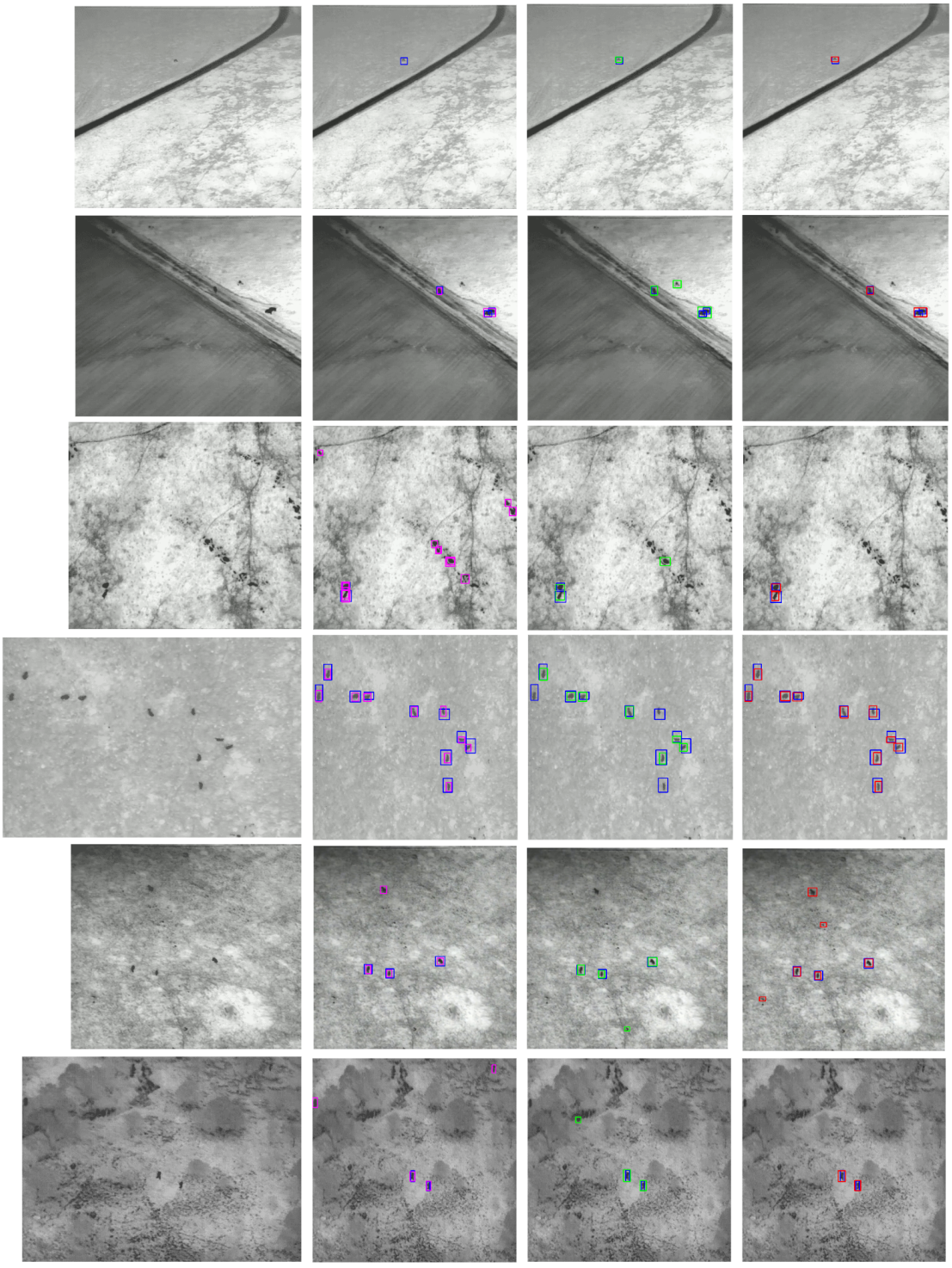
While quantitative scores are able to give an indication of the detection performance of each model, qualitative evaluation allows us to verify whether the system would give valuable results in practice. To accomplish this, the predictions made by each model when evaluated on the test set were visualised as bounding boxes on the original image, and compared to the ground truth bounding boxes. This is pivotal to understand whether the predictions give sensible results in practice. In a similar manner to the plotted pseudo-labels, each model's predictions are colour-coded while the ground truth bounding boxes appear in blue. It should also be noted that the predictions are visualised by plotting the bounding boxes with confidence scores above the confidence threshold which yields the highest $F1$ score. This is the same confidence threshold at which the earlier Precision, Recall and mAP values were reported.

In order to analyse the differences between the predictions of each model, two visuals containing a subset of frames are included in Fig. 5.4 and 5.5 below. The predictions represent performance on the full split-by-video test set. While Fig. 5.4 exhibits the ability of the pseudo-label trained models to perform well in a variety of environments, Fig. 5.5 represents several challenging cases and shortcomings. We consider only the models trained fully on the pseudo-label sets, without any added ground truth substitutions. Firstly, the model trained with only DINO-based pseudo-labels is considered.

DINO-trained Predictions

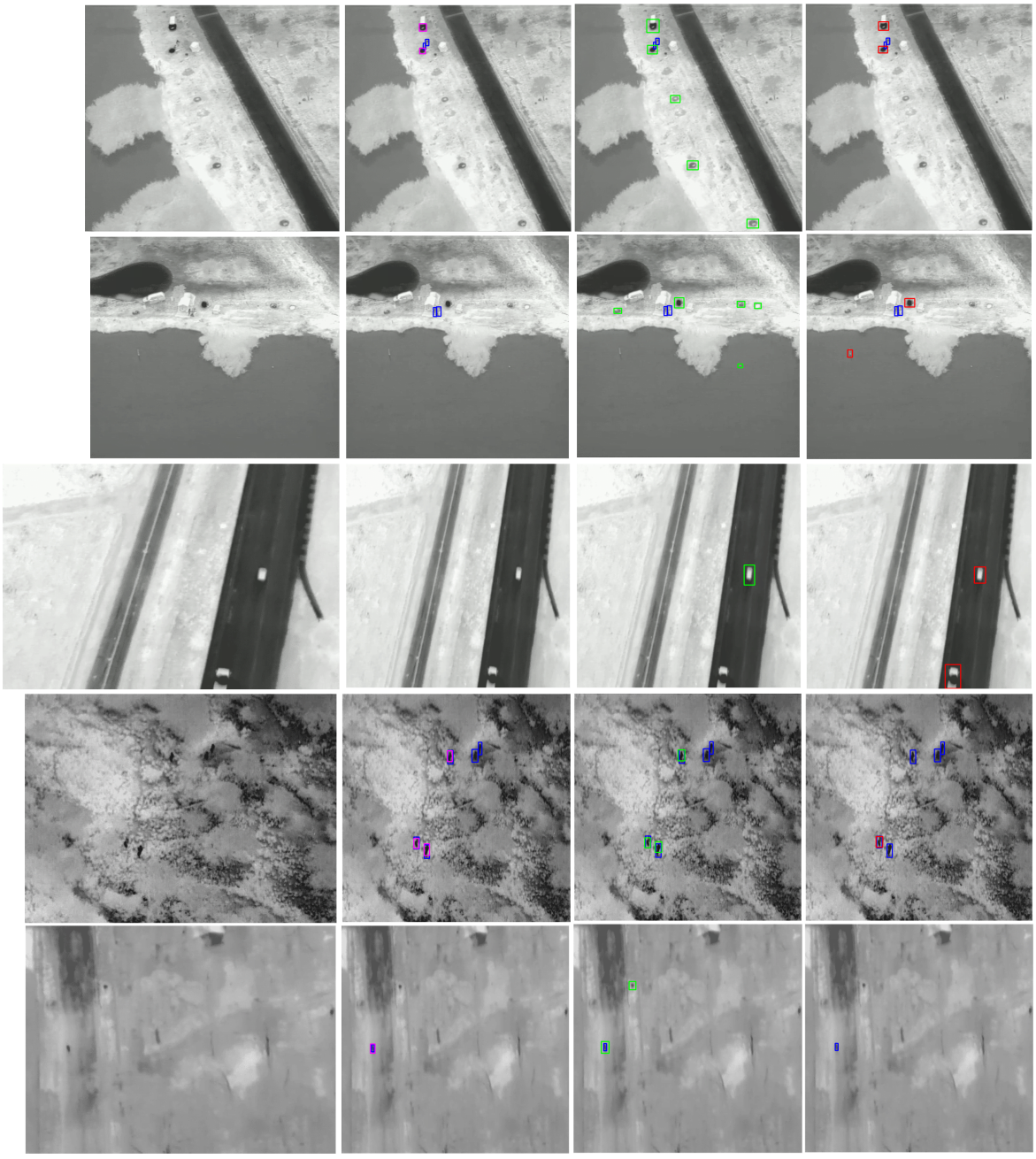
As indicated in Fig. 5.4, the model trained with the DINO-based pseudo-labels (red) displays an ability to produce sensible predictions for target objects across a wide variety of environments. Thus the core aim of being able to capture objects of interest with a bounding box tightly enclosing the object is achieved. This indicates that the proposed method could indeed provide a viable alternative to manual annotation in low-resource scenarios, as the overall purpose of the detection network is retained.

Nevertheless, several shortcomings of the DINO pseudo label-trained model are apparent in Fig. 5.5. Firstly, the first two frames depicting complex scenarios containing buildings and other distracting objects cause the real objects of interest to be omitted from the predicted bounding boxes. However it should be noted that none of the models are able to detect these targets. While this behaviour is undesirable, it is not unexpected as several objects appear in the frame that bear resemblance to real targets, and thus these objects are detected by the models rather than the true targets. While in certain cases all models struggle to identify the true objects of interest, there are some scenarios in which the ground truth-trained model provides better predictions. In frame 4 and 5 of Fig. 5.5, it can be observed that the ground truth-trained and FastFlow-trained models are able to detect significantly more objects of interest than the DINO equivalent. It is important to note that these discrepancies vary from frame to frame, where each model may in turn to slightly better. Moreover, due to the nature of the application as a video stream in practice, the detector may miss an object in one frame but correctly identify it in the next.



(a) Input image. (b) GT-trained predictions (pink) vs labels (blue). (c) FF-trained predictions (green) vs labels (blue). (d) DINO-trained predictions (red) vs labels (blue).

Figure 5.4: Qualitative comparison of the predictions made on the test set for the models trained with ground truth labels (pink), FastFlow-based pseudo-labels (green) and DINO-based pseudo-labels respectively (red). Ground truth labels are shown in blue. It can be observed that the pseudo label-trained models are largely able to detect objects of interest in a similar manner to the ground truth-trained model, across a variety of environments.



(a) Input image. (b) GT-trained predictions (pink) vs labels (blue). (c) FF-trained predictions (green) vs labels (blue). (d) DINO-trained predictions (red) vs labels (blue).

Figure 5.5: Qualitative comparison of the predictions made on the test set for the models trained with ground truth labels (pink), FastFlow-based pseudo-labels (green) and DINO-based pseudo-labels respectively (red). Ground truth labels are shown in blue. Several challenging cases are depicted, showing variable performance across the three models.

FastFlow-trained Predictions

Similarly to the DINO-trained model, the model trained on the FastFlow-based pseudo-labels is largely able to identify the objects of interest in each frame in Fig. 5.4. Thus this technique also shows immense potential to produce useful detection models.

From Fig. 5.5 we observe that the predictions from the FastFlow-trained model exhibit similar shortcomings to the DINO equivalent such as missing certain targets and the presence of false positives. However, as can be seen in both visuals, the observed false positives for both the FastFlow and DINO-trained models often enclose objects closely resembling the characteristics of objects of interest, rather than purely noisy predictions. This is particularly interesting as the noisy bounding boxes observed in the DINO pseudo-labels due to background structures are not present in the resulting predictions, and therefore the false positives present often focus on objects with similar characteristics to the true targets. This suggests that the YOLOv5 detection model is tolerant of some noise in the training labels. In addition, since false negatives are more undesirable in the poaching detection system, a few false positives that do not overwhelm the purpose of the detector are acceptable.

Lastly, a clear difference between the ground truth-based and pseudo label-based models is their treatment of vehicles. Both pseudo label-based models identify vehicles as objects of interest, as indicated by frame 3 in Fig. 5.5. While this does not detract from the aim of the network if the vehicles appear as the only objects in the frame, it may distract the networks if vehicles appear alongside real objects of interest. Overall, the visualised predictions verify the value of the pseudo-labelling techniques and indicate the feasibility of training a poaching detection network with pseudo-labelled data. Notably, both models trained with pseudo-labels only are able to correctly detect single or multiple objects in the frame, thus fulfilling the main role of the system. Therefore, although imperfect, pseudo-labels can be used to train a poaching detection network with practical outcomes.

5.3 Further Exploration

In addition to the qualitative and quantitative detection results, additional factors such as the inference speed, efficiency and model response to changes in pipeline hyperparameters were explored in order to provide a more complete analysis.

5.3.1 Speed and Efficiency

In order to evaluate model speed and efficiency, the total size of each model was recorded and several experiments were run to measure detection speed. The models trained on the ground truth labels, full DINO pseudo-labels and full FastFlow pseudo-labels were evaluated on the full split-by-video test set and their detection speeds were measured on both a CPU and GPU. Furthermore, each model speed was tested by calculating predictions in batch as well as for single frames. Lastly, the inference speed including saving the predictions to text file was measured. The subsequent results are tabulated in Table 5.13.

	GPU			CPU		
Labels	*bs 32*	*bs 1*	*bs 1 + save*	*bs 32*	*bs 1*	*bs 1 + save*
Ground Truth	69.33	36.36	1.98	3.17	3.35	0.93
DINO	66.56	39.31	1.32	3.04	3.36	0.84
FastFlow	67.2	38.55	3.08	3.03	3.27	1.52

Table 5.13: Detection speed measured in FPS for batch size 32, batch size 1 and batch size 1 including saving the results to file.

The total size of each model was found to be 14MB as expected. Furthermore, from Table 5.13 we can observe that the final YOLOv5s model operates at approximately 66-69 FPS (GPU) in batch and 36-39 FPS (GPU) when evaluating single frames in sequence. Since in practice the model will evaluate a single frame at a time, the FPS at a batch size of 1 is most significant. The models are all able to attain greater than 35 FPS, which is sufficient to fulfil the real-time requirements of the poacher detection network.

The differences in inference speed between the models was found to be nominal for both batch size 1 and a batch size of 32. However, it should be noted that saving the detection results to individual *.txt* files significantly hampers the speed of the model. This is particularly significant for the GPU speeds which are reduced by more than a factor of 10. Thus it is imperative that an efficient way of communicating detection information back to the base station is developed in order to prevent slow inference speed. It can also be noted that in real-world systems, every n_{th} frame will be processed for detections in order to reduce the load.

5.3.2 Robustness to Changes in Hyperparameters

A further investigation involves an exploration of each method's tolerance to changes in hyperparameters. Accordingly, the response of each model to changes in the filtering and padding applied to the pseudo-labels during post-processing is evaluated. Specifically, each set of pseudo-labels is post-processed using a range of hyperparameters and the resulting models are evaluated on the full split-by-video test set including complex building frames. For each experiment, the best combination of hyperparameters for the DINO and FastFlow pseudo-labels is highlighted, while the performance of the ground-truth trained model is included for comparison.

Effects of Filtering Threshold

Firstly, the amount of filtering applied to the pseudo-labels during post-processing is varied by changing the filtering threshold. In post-processing, any bounding boxes with both width and height less than the filtering threshold are filtered out to remove small noisy bounding boxes. Thus as the threshold increases, larger bounding boxes are filtered out.

Labels	Filtering	Padding	P	R	mAP$_{@0.5}$
Ground Truth	N/A	N/A	0.298	0.313	0.190
DINO	0.010	0.01	0.259	0.268	0.135
DINO	0.015	0.01	**0.284**	**0.293**	**0.154**
DINO	0.020	0.01	0.275	0.280	0.144
FastFlow	0.010	0.01	0.203	0.287	0.104
FastFlow	0.015	0.01	0.213	0.288	0.110
FastFlow	0.020	0.01	**0.219**	**0.292**	**0.110**

Table 5.14: Effects of filtering on detection performance, with padding kept constant.

From Table 5.14 we can observe that the DINO-trained models seem more susceptible to changes in the amount of filtering applied to the pseudo-labels. While the performance of the FastFlow-trained models remains relatively constant, the performance of the DINO-trained models varies slightly with the change in the filtering hyperparameter. Thus it is important to apply enough filtering to filter out the small noisy boxes present in the DINO pseudo-labels but not to filter too much such that it hurts recall. It can also be noted that while the original choice of filtering hyperparameter, 0.015, seems optimal for the DINO pseudo-labels, the FastFlow pseudo-labels maintain almost consistent performance across the investigated filtering threshold values and thus using a value of 0.020 only marginally outperforms the other hyperparameter combinations.

Notably, there remains a significant discrepancy in precision and mAP score between the DINO-trained and FF-trained model performance across all filtering hyperparameter combinations, with the DINO-trained models attaining better outcomes. This is in line with the previous observations from the main results. Therefore this observed difference in performance cannot be attributed to a sub-optimal hyperparameter choice.

Effects of Padding Amount

Secondly, the amount of padding added to the width and height of each pseudo-label bounding box is varied. To determine the effects of padding on the detection results, the model was trained on pseudo-labels with no padding (padding of 0.0), padding of 0.01 and padding of 0.02. The results are summarised in Table 5.15 below.

Labels	Filtering	Padding	P	R	mAP$_{@0.5}$
Ground Truth	N/A	N/A	0.298	0.313	0.190
DINO	0.015	0.0	0.174	0.236	0.078
DINO	0.015	0.01	0.284	**0.293**	0.154
DINO	0.015	0.02	**0.296**	0.284	**0.159**
FastFlow	0.015	0.0	0.174	0.256	0.081
FastFlow	0.015	0.01	**0.213**	**0.288**	**0.110**
FastFlow	0.015	0.02	0.189	0.251	0.089

Table 5.15: Effects of padding on detection performance, with filtering kept constant.

From Table 5.15 it appears that the DINO-trained models remain more sensitive to changes in hyperparameters. The addition of padding is shown to have a significant impact on the performance of the DINO-trained models, especially in terms of precision and mAP score. Across the board for both DINO-trained and FastFlow-trained models, including padding in the post-processing steps is shown to improve the detection results. While FastFlow appears to perform best at the previously selected padding amount of 0.01, the results indicate that slightly more padding may benefit the DINO pseudo-labels. However the difference in performance gained by doubling the padding is nominal in comparison to the difference between some padding and no padding at all. This supports the intention of adding padding to more closely match the bounding boxes of the labels.

Overall the investigation yields positive results, indicating that the original choice of hyperparameters was sensible and generalises well to both the DINO and FastFlow pseudo-labels. While some variability in performace is observed from changes in hyperparameters, the integrity of the detection system is maintained. General heuristics such as being careful not to filter out too many bounding boxes and not neglecting to pad the pseudo-labels can ensure that good detection performance can be maintained without extensive hyperparameter tuning.

Chapter 6

Discussion

We now present an analysis of the results obtained in light of the research objectives outlined at the start of the investigation.

6.1 Reducing Annotation Cost

The main objective of the study involved investigating the feasibility of using unsupervised or self-supervised techniques to reduce the annotation cost associated with training a poaching detection network.

6.1.1 Feasibility of Techniques

From the results presented we have observed that both the DINO and FastFlow pipelines are able to produce pseudo-labels which can identify and locate objects of interest. The resulting pseudo label-trained models perform well in most natural environments that are likely to be encountered in practice. Furthermore, both pipelines were able to generalise to a large variety of environments without specific pre- or post-processing modules or manual intervention. Since only generalised post-processing principles were used, it is likely that the pipeline would be able to perform well on similar thermal infrared or aerial datasets without significant change to the pipeline modules. However, as demonstrated by the results, the pipelines would be less suitable for use on datasets containing very busy frames filled with distracting objects.

6.1.2 Detection Accuracy

In terms of detection performance, we have observed that both techniques are able to reach approximately 90% of baseline recall, with the DINO model reaching close to baseline performance across all metrics. Notably, a large difference in performance was observed depending on whether the data was split according to timestamp or according to video. This considerable difference is further explored below.

Split-by-Timestamp vs Split-by-Video

There are multiple potential causes of the difference between the results in the split-by-timestamp and split-by-video experiments. Firstly, the performance of the ground truth-trained model in the split-by-video experiments is significantly lower than the performance of the split-by-timestamp equivalent. This may be due to better prevention of data leakage in the split-by-video experiments by withholding entire videos for testing. Accordingly, the ground truth-trained model would not be able to learn the environment and its corresponding targets so closely, leading to more realistic results. Furthermore, during the split-by-timestamp experiments, it is possible that any sources of error in the pseudo-labels were more closely reflected in the predictions since the test set was made up of the same videos only later in the timespan.

For the split-by-video experiments, the performance of the ground truth-trained model is lower as the test videos are completely unseen. Thus since this is a better test of generalisation, these results are likely to be closer to realistic performance. Both pseudo label-trained models are seen to perform better in the split-by-video experiments, almost matching the recall of the baseline model. It is possible that the network learns more general features from the pseudo-labels and therefore does not memorise the specific sources of error in each pseudo-label set, leading to better results and greater tolerance to noise in the training set. In addition, the training set used in the split-by-video experiments may have included a better representation of the types of target objects, leading to improved recall from both pseudo-label based models. Therefore the composition of the training set plays an important role in the final predictions.

Detection Score Comparison

The results can additionally be examined in light of the literature and previous benchmark results. The study that is of most relevance to this investigation is that of Bondi *et al.* [2020] who benchmark four detection models on the thermal-infrared aerial poaching detection dataset, BIRDSAI. The best performing model, Faster-RCNN with Weighted Cross Entropy (WCE) loss, obtained an overall mAP of 0.438, while the worst performing model was YOLOv2 with a mAP of 0.304. These results indicate similar performance to the results obtained on the SPOTS dataset in this study. In addition, it can be noted that existing literature on poacher detection or animal detection from aerial UAV imagery reports reasonably low accuracy in comparison to non-aerial detection. This points to the wider challenge present in the field of object detection regarding poorer performance on small-scale targets, which particularly impacts aerial imagery. Overall, a comparison to external results indicates that the results of the study are in line with expected performance in the poaching detection domain.

6.1.3 Cost Reduction

The central question of the investigation centred on whether a significant reduction in the annotation cost associated with poaching detection can be obtained through the application of unsupervised and self-supervised methods. From a cost reduction perspec-

tive, both techniques require very little manual input. The DINO-based pipeline uses a pre-trained DINO model to calculate the attention maps followed by an automatic post-processing pipeline. The FastFlow-based pipeline requires some hand sorting to create the training set of background-only images on which the FastFlow network is fine-tuned, however no hand-annotated localisation information is required. Moreover, the FastFlow network could also be used without fine-tuning, at the cost of an increase in noise. Therefore both techniques substantially reduce annotation cost. Following the creation of the pseudo-labels, the process to train the detection model for use onboard the conservation drone remains unchanged, therefore ensuring simplicity of the overall system. Each pipeline can therefore be used as a substitute for time-consuming manual annotation of the thermal infrared videos.

Overall the investigation has demonstrated that both techniques provide a promising strategy to reduce annotation costs. Despite some noise present in the pseudo-labels, the resulting models achieve close to baseline recall. This is particularly significant for organisations who may not have sufficient resources to obtain detailed ground truth for large amounts of data, or those that are planning a rapid development of a poaching detection network and wish to avoid time-consuming and expensive annotation. Both the pseudo-labels and predictions are able to fulfil the inherent requirements of the poaching detection task by identifying and localising targets of interest in the image.

6.2 Pseudo-Label Quality

A supporting research objective involved whether a small proportion of ground truth could be used together with the generated pseudo-labels such that detection performance might be improved. Hence in this section we explore the quality of each pseudo-label set, the impact of including certain ground truth labels as well as external sources of error leading to mismatches between the predicted and ground truth bounding boxes.

6.2.1 Performance of Pseudo-Labels in Different Scenes

As indicated by the visualised pseudo-labels in section 5.1 of the results, both pipelines are able to localise target objects in a variety of challenging environments. As indicated in section 5.1, poaching detection systems may need to operate in scenes varying from open plains to sparse vegetation to dense bushland. Therefore, Fig. 5.1 in the results demonstrates the ability of the pipelines to localise targets in these differing conditions. In contrast, a central limitation of the proposed system is observed when faced with crowded scenes containing buildings. As discussed in the results, both the FastFlow and DINO pseudo-labels fail to recognise the target objects and instead locate other irrelevant objects. Due to this poor performance in cluttered building environments, replacing this subset of pseudo labels with ground truth increases performance as the detection network is able to learn the correct target features. Overall, the qualitative and quantitative results presented indicate the ability of the pseudo-labels to detect

single and multiple targets for a variety of natural environments. Moreover, although the pseudo-labelling procedure contains error, the resulting pseudo-labels allow the YOLOv5 detection model to learn useful features for detection.

6.2.2 Selective Inclusion of Ground Truth Labels

A notable result is the importance of training on fewer, more accurate pseudo labels rather than a full set of pseudo labels containing noisy or erroneous examples. Due to the clear distinction in performance between natural bush and building scenes, videos containing largely natural scenes could benefit from the pseudo-labelling pipeline while a minority of videos containing cluttered building scenes could be hand-annotated. As indicated by the results of both the DINO and FastFlow pipelines, this combination of a majority of pseudo-labels together with a small number of specific ground truth labels can be leveraged to produce significant improvements in detection performance. This becomes increasingly important if the model is evaluated on a high proportion of difficult frames.

Since the pseudo label-trained models are able to match or exceed baseline performance in natural scenes, selective ground truth labels were considered for building scenes only in order to improve performance on these scenes. It can be noted that the selected ground truth labels formed approximately 10-15% of the training data for each case, namely *DINO + GT for Buildings* and *FF + GT for Buildings*. In further studies, the impact of the addition of a small proportion of ground truth across all scenes could be considered. However, due to the high levels of performance from the pseudo-labels alone, together with the observed improvement from the ground truth for building scenes, this was not included in the scope of this study.

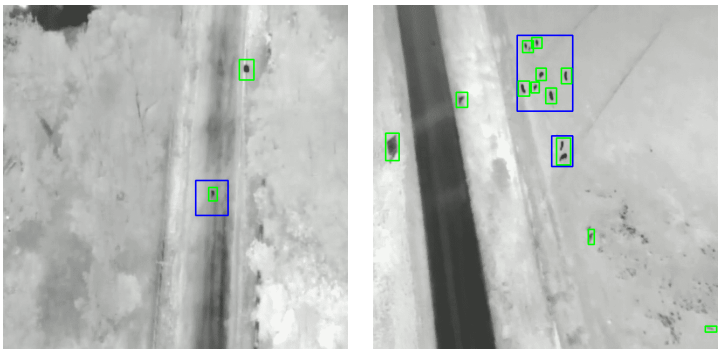
Another practical recommendation involves ensuring that the training images clearly show the types of objects of interest such that the network learns to focus on the most important features of the target objects. This factor may have played a role in the difference in performance between the split-by-timeframe and split-by-video results. Therefore, a smaller set of clear training examples with good quality labels leads to better results than large amounts of ambiguous training data. Accordingly, using the pseudo-labelling pipelines to produce annotations for videos with clear targets and training mostly on these frames will lead to better results than applying the pipelines to all incoming video data without discrimination.

6.2.3 External Sources of Bounding Box Error

While some errors arise from the pseudo-labelling pipeline themselves, in some cases external factors lead to the disparity between the pseudo-labels and ground truth annotations. Lower mAP values in some cases may be attributed in part to a low IOU score between the predicted bounding box and the ground truth. In addition to the sources of error stemming from the pseudo-labels themselves, several other factors contribute to mismatches between the predicted and ground truth bounding boxes.

Differences in bounding box size

As indicated in Fig. 6.1a, in certain cases the ground truth data may be annotated such that the object of interest is not tightly enclosed. Thus since the anomaly detection mask or self-attention map outlines the object of interest more closely, this leads to a poorer IOU score even in cases where the object has been detected and localized correctly, and perhaps more accurately.



(a) Difference in bounding box size. (b) One vs multiple targets.

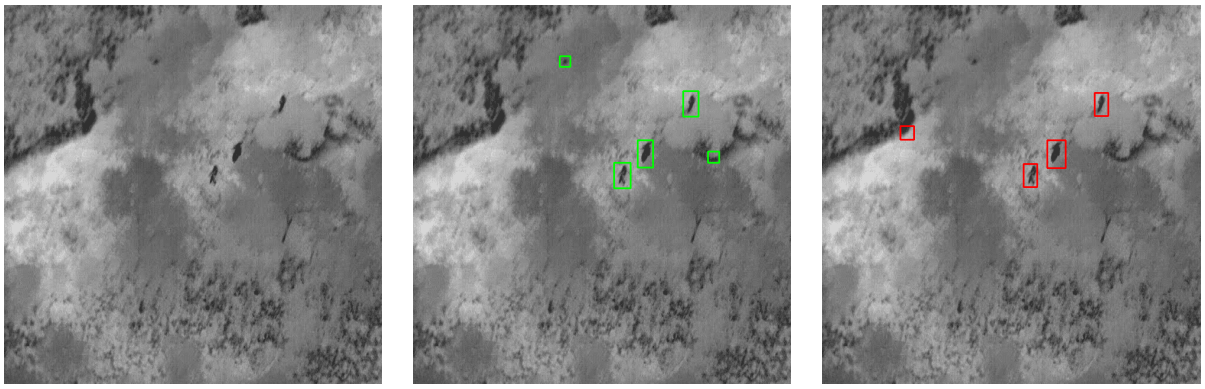
Figure 6.1: Sources of difference between ground truth (blue) and FastFlow pseudo-labels (green) independent of pseudo label error.

One vs multiple targets

A further source of bounding box mismatch stems from cases where the ground truth data may annotate multiple targets as one target, while the pseudo-labelled data may perform the same error in a different case. For example in Fig. 6.1b the ground truth labels enclose several small objects in one bounding box, which would negatively affect the overall metrics despite the correct recovery of the objects of interest.

Missing or incorrect annotations

Most significantly, in certain cases, the ground truth annotations appear incomplete or completely omit certain target objects.



(a) Ground truth with missing targets. (b) Poachers identified by FastFlow. (c) Poachers identified by DINO.

Figure 6.2: Example of missing ground truth annotations.

An example of this is included in Fig. 6.2 where the ground truth does not provide labels for the poacher target objects but both the DINO and FastFlow pseudo labels are able to identify the poachers. Overall, these additional sources of mismatch between the ground truth pseudo bounding boxes can contribute to the observed lower mAP scores in some cases even when recall and precision are higher.

6.3 Detection Speed and Efficiency

A central question of this investigation is whether it is realistic to incorporate unsupervised and self-supervised techniques into real-world poaching detection systems, and whether any benefits gained in reducing annotation cost come at the cost of a decrease in detection speed or efficiency.

6.3.1 Inference Speed

A particularly desirable result is the ability of the proposed self-supervised and unsupervised strategies to maintain a very similar speed and model size to their fully supervised equivalent. While the inference speed of the baseline is able to meet real-time requirements at 36 FPS, the models trained on the pseudo-labels attain similar speeds a 39 and 38 FPS for DINO and FastFlow respectively. While almost double the inference speed can be attained using a batch size of 32, in practice the images will be processed in single frames and thus the inference speed with batch size 1 is taken as most significant. An even further improvement in speed may be gained by converting the YOLOv5s model to TensorRT to run onboard the conservation drone. Therefore while the trade-off between detection accuracy and inference speed remains a central challenge to automatic poaching detection, it is encouraging to observe that the introduction of unsupervised and self-supervised techniques can allow for a reduction in annotation cost while maintaining detection speed. This suggests that the addition of self-supervised or unsupervised techniques in this manner could be viable in a real-world scenario.

6.3.2 Model Size

Furthermore, from an efficiency perspective, the YOLOv5s network was selected for its compact size at 14MB, allowing it to run effectively on the drone platform. A potential benefit of the proposed self-supervised approach is that heavyweight models such as Vision Transformers can be utilised to provide semi-automatically labelling while leaving the lightweight detection model unaffected and able to be deployed on the computationally constrained UAV. Since the pseudo-labels do not affect the resulting model size, the efficiency constraints associated with the conservation drone can be met. On the other hand, while the FastFlow model is able to produce useful pseudo-labels, the model may also be able to run directly on the UAV, and is recommended for further investigation.

6.4 Selecting an Optimal Technique

The final research question pertained to which of the investigated methods of reducing annotation cost would be the most suitable for the task at hand. Several notable factors can be considered when comparing the two pipelines, including ease of creation, time taken to generate the pseudo-labels, detection performance, sensitivity to hyperparameters and generalisation ability.

6.4.1 Ease of Pseudo-labelling Procedure

While both pseudo-labelling techniques make use of publicly available and well-maintained code bases, both methods have specific benefits and drawbacks. Firstly, the DINO pipeline makes use of a pre-trained DINO *DeiT-S* model to calculate the six attention maps, and thus requires no fine-tuning or change to the available existing model. In contrast, the FastFlow model benefits from being fine-tuned on a set of background-only images in order to produce cleaner segmentation masks. Therefore additional effort is needed to first fine-tune the FastFlow model that is not required for the DINO equivalent. However, an advantage of the FastFlow network is its quick training time and fast evaluation speed. Therefore, the aforementioned fine-tuning process does not require long training times. In addition, generating pseudo-labels with the FastFlow method is a much faster process than the DINO equivalent since the FastFlow segmentation masks are calculated more rapidly than the DINO attention maps.

6.4.2 Pseudo-Label Comparison

As discussed in section 5.1 of the results, the focus of the DINO attention maps on background objects often leads to noisy false positives, and in some cases missed objects. In addition, since the DINO attention maps always focus on something in the frame, the resulting pseudo-labels are much more likely to detect a false object in every frame, and therefore struggle to identify background-only images. On the other hand, the filtering required to remove edge artefacts from the FastFlow pseudo-labels can result in missed targets on the edges of the frame. Thus both sets of pseudo-labels contain sources of error specific to their mechanisms.

6.4.3 Accuracy of Predictions

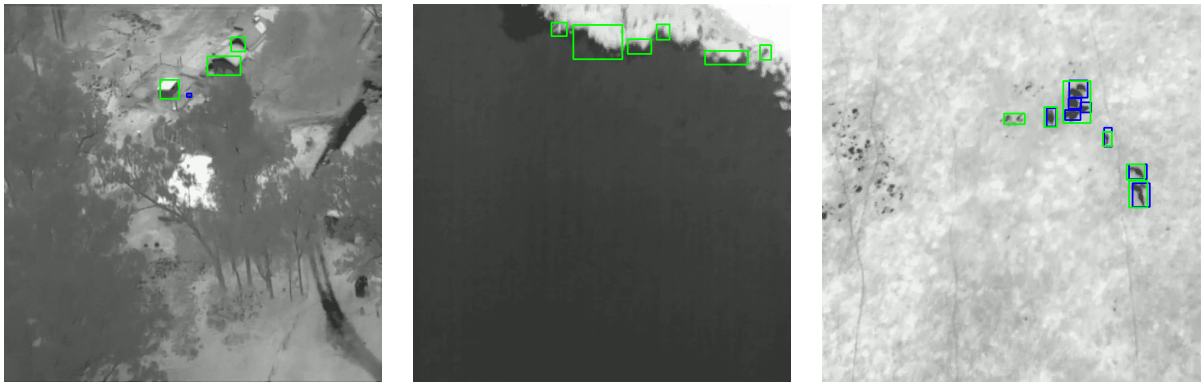
Despite the clear differences in the respective pseudo-labels, the resulting effects on detection model performance are not as clear cut. For example, although the DINO pseudo-labels contain false positives due to distracting background features such as roads, these are not present to the same extent in the predictions. A possible reason for this is that the false positives created from background structures in the DINO pseudo-labels are dissimilar to one another and thus may not be learnt as features of interest. This suggests that the YOLOv5 model is able to tolerate a certain level of noise in the training labels. Hence the remaining false positives largely focus on objects with similar characteristics to real targets.

In addition, a particularly interesting result is that better performance is observed for the FastFlow-trained models in the split-by-timestamp experiments, while superior performance is obtained by the DINO-trained models in the split-by-video evaluation. On analysis, this discrepancy is mainly due to the respective models under-performing on one particular type of image in the respective test sets. For example, in the split-by-timestamp experiments, the DINO-trained models perform worse on frames containing tightly clustered targets due to the presence of only one poorly labelled example of clustered targets in the training set. This result points to the impact of the training set composition. For optimal results the training set should comprise mainly clear, distinct targets with more than one frame of each type of target object to ensure that the correct features of interest are learnt. This is corroborated by the difference in the split-by-timestamp and split-by-video performance as a whole, where the split-by-video data splits resulted in a more representative training set. Therefore, composition of a representative and clear training set will aid in achieving optimal performance for both techniques.

Lastly, the DINO pseudo-labels seem slightly more sensitive to changes in pipeline hyperparameters. However this can be mitigated through a few simple heuristics and is not significant enough to distinguish one technique as clearly better than the other. Therefore, on consideration of the benefits and limitations of each technique, neither of the proposed techniques clearly outperforms the other. Instead, the drawbacks of each approach stemming from their intrinsic mechanisms should be considered. A further exploration could be undertaken to investigate whether the resulting pseudo-labels from the two techniques could be combined to enhance performance.

6.5 Remaining Challenges

Finally, the limitations of the investigation and remaining challenges are highlighted for further study. A clear distinction can be drawn between the scenarios and environments in which the pseudo-labelling procedure works successfully and where it is less suited. Specifically, as demonstrated in the results, the pseudo-labelling pipeline works well in cases where the target can be clearly distinguished from the background, where there are multiple distinct targets and the background environment contains a variety of natural, open or bush-like terrains without buildings. In contrast, the pseudo-labels perform poorly for complex or cluttered frames containing buildings or other distracting items. Other scenarios which cause inadequate results include frames containing the border between land and a body of water and frames where targets are tightly packed together and therefore difficult to distinguish. These challenging cases are depicted in Fig. 6.3.



(a) Cluttered frames with buildings. (b) Border of land and water body. (c) Closely packed targets.

Figure 6.3: Challenging cases leading to poor quality pseudo labels.

It should be noted that quantitative performance does depend largely on the composition of the test set, as observed in the differences in performance between test set with building frames and without. Therefore it is likely that all models would struggle on a video composed mostly of building scenes, while all models would perform favourably on videos with distinct targets in a natural environment. In further work, specific exploration into improving performance on more difficult scenes is warranted to avoid inconsistent performance. However, it should be noted that the pipelines are still able to perform well in frames that are cluttered with natural objects such as bushes or trees. Therefore, due to the vast majority of game reserve terrains comprising open bushland, the pseudo-labelling techniques presented could offer much value to conservation organisations as buildings and complex scenes would appear only very seldom in the collected thermal data.

Chapter 7

Conclusion

In this study we have identified a central limitation of current aerial poaching detection systems which rely heavily on detailed ground truth. The expensive and time-consuming methods required to obtain the annotations together with the limited ability of low-resource conservation organisations to annotate vast amounts of thermal data often pose a barrier to effectively utilising available thermal imagery for training. Due to the urgent nature of the poaching crisis and the promising strategy of using Unmanned Aerial Vehicles (UAVs) to detect poachers in large reserves, the ability to train detection networks with lower annotation costs could provide a vital solution.

Consequently, this study explored the construction of two pseudo-labelling pipelines using self-supervised and unsupervised techniques respectively to semi-autonomously label thermal data and thus allow for the detection of poachers with cheaper annotations. The accuracy, speed and efficiency of the proposed solutions were evaluated in a poaching detection-specific context through the use of a real-world dataset from the local conservation organisation SPOTS. The main contributions of the study are summarised below:

- We demonstrate the feasibility of generating pseudo-labels from the unsupervised anomaly detection algorithm FastFlow and self-supervised learning method DINO to train a YOLOv5s detection network known to fulfil the real-time and efficiency constraints of the UAV.
- We show that using the pseudo-labels produced from each pipeline, we can attain 90% or more of the baseline recall, with no manual annotation.
- Furthermore, we demonstrate that both pseudo-labelling techniques can match and even exceed the performance of the baseline model across all metrics in natural scenes without buildings.
- We illustrate the importance of selecting a smaller set of higher-quality pseudo-labels with clear targets, and the advantages of further improving performance by manually annotating only the most difficult training videos.

Therefore in reference to the central research hypothesis we have shown that both methods can be used to create pseudo-labels for poaching detection and achieve up to

90% of the recall of a fully supervised baseline. Moreover, the final YOLOv5 model is able to meet the speed and efficiency requirements associated with the drone platform with a model size of 14MB and inference speed of +30 FPS. A notable observation is that the final performance is affected by both the composition of the training set as well as the characteristics of the environments present in the test set. Therefore use of the pseudo-labelling pipelines on non-cluttered scenes and careful composition of a training set containing clear examples of target objects is recommended for optimal results.

Despite these limitations, both techniques are shown to perform well on the natural environments that make up the majority of the poaching detection landscape. Overall, by generating pseudo-labels in a semi-automatic manner, large amounts of unlabelled thermal data can be harnessed without extensive and costly manual annotation effort. Thus, unsupervised and self-supervised pseudo-labelling are shown to be promising tools to aid in annotation of poacher detection video data, therefore lowering the barrier to entry for many resource-constrained conservation organisations.

7.1 Recommendations for Future Work

While this study serves as an initial exploration into reducing the annotation cost associated with the task of poaching detection, several aspects could be further improved or expanded. Four principle ways in which the study could be further advanced are presented below.

7.1.1 FastFlow as the Primary Detection Algorithm

It should be noted that the FastFlow anomaly detection algorithm could be considered as a substitute for the YOLOv5 object detection model onboard the UAV due to its speed and overall compact model size. This would greatly reduce the need for annotation as the training set needed to train the FastFlow algorithm is comprised only of normal background images with no annotated targets. Moreover, the resulting bounding boxes from the developed pipeline closely resemble that of the ground truth, thus demonstrating the potential value of this technique. However, experimentation would be required to determine whether the FastFlow algorithm would be able to run successfully on the GPU onboard the UAV platform as well as if the detection results would be sufficient in practice. Consequently, the feasibility of running the FastFlow algorithm as the primary detection model onboard the UAV is recommended for further study.

7.1.2 Distinguishing between Animal and Human Targets

Since this study focuses primarily on investigating the feasibility of using self-supervised and unsupervised pseudo-labelling techniques in place of manual annotation, the detection problem was simplified to detect both animals and humans as one target class. To extend the investigation, a distinction could be made between animal and human classes while training the YOLOv5 detection model. In order to augment the existing

pseudo-labelling pipelines, a method of classifying the identified objects could be developed. Several approaches such as an unsupervised clustering algorithm or separate classification step could be added to provide a class label for each bounding box in the pseudo-label. This may aid further in distinguishing between target objects such as poachers and surrounding objects such as rocks or vegetation.

7.1.3 Further Increasing Recall

In addition, due to the significance of recall as the defining metric for the poaching detection task, specific methods to increase recall should be further explored. Several possibilities include varying the proportions of background-only frames included in the training set and further hyperparameter tuning for both the pseudo-labelling pipeline hyperparameters and YOLOv5 training hyperparameters. In addition, the confidence threshold used to filter out bounding box predictions could be lowered at the expense of gaining more false positives. Given a manageable amount of false positives, this trade-off could result in increased recall of target objects and thus better detection performance in practice. Lastly, the poor performance of small object detection remains an on-going problem in the field. Current literature on aerial animal and poacher detection such as Bondi *et al.* [2018b] indicates low accuracy in comparison to non-aerial applications. Thus further development of small-scale object detection would greatly benefit conservation applications.

7.1.4 Exploring Alternative Methods of Reducing Annotation Cost

While this study has examined the possible benefits of using self-supervised and unsupervised techniques in the context of pseudo-labelling, several other approaches could be considered to reduce annotation cost. Although the techniques presented were carefully selected as the methods that seemed most applicable to the given problem, the study was limited to only one method from each supervision class. Thus, other unsupervised or self-supervised algorithms are recommended for further study in subsequent analyses. Moreover, training in a semi-supervised manner may provide significant benefit if a small amount of ground truth data is available. The detection model could thus be trained with the small amount of ground truth labels, and the trained network could then be used in turn to label more data. Therefore, further exploration into alternative methods of supervision is warranted.

7.1.5 Testing in Real-World Conditions

Lastly, since the analysis is restricted to pre-recorded poaching detection data, additional challenges associated with the live system could not be accounted for. A specific investigation into whether the proposed approaches would work well in the real use case would be of immense benefit to ensure that the problem brief is truly met.

In conclusion, the complexity of aerial poaching detection necessitates further exploration and analysis which extends beyond the bounds of this study. Nevertheless,

our findings indicate the value of unsupervised and self-supervised techniques for conservation and motivate that the introduction of unsupervised or self-supervised learning into current systems could provide a more robust and sustainable solution to the critical problem of poaching detection.

References

[Air Shepherd 2021] Air Shepherd. *Air shepherd: The lindbergh foundation*. https://airshepherd.org/, 2021. Accessed: 2021-05-15.

[Akcay *et al.* 2018] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pages 622–637. Springer, 2018.

[Akçay *et al.* 2019] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[Akcay *et al.* 2022] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. Anomalib: A deep learning library for anomaly detection. *arXiv preprint arXiv:2202.08341*, 2022.

[Baur *et al.* 2019] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI brainlesion workshop*, pages 161–169. Springer, 2019.

[Beery *et al.* 2019] Sara Beery, Dan Morris, and Siyu Yang. Efficient pipeline for camera trap image review. *arXiv preprint arXiv:1907.06772*, 2019.

[Bergmann *et al.* 2018] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

[Bergmann *et al.* 2019] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[Bondi *et al.* 2017] Elizabeth Bondi, Fei Fang, Debarun Kar, Venil Noronha, Donnabell Dmello, Milind Tambe, Arvind Iyer, and Robert Hannaford. Viola: Video labeling application for security domains. In *International conference on decision and game theory for security*, pages 377–396. Springer, 2017.

[Bondi *et al.* 2018a] Elizabeth Bondi, Debadeepta Dey, Ashish Kapoor, Jim Piavis, Shital Shah, Fei Fang, Bistra Dilkina, Robert Hannaford, Arvind Iyer, Lucas Joppa,

et al. Airsim-w: A simulation environment for wildlife conservation with uavs. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–12, 2018.

[Bondi *et al.* 2018b] Elizabeth Bondi, Fei Fang, Mark Hamilton, Debarun Kar, Donnabell Dmello, Jongmoo Choi, Robert Hannaford, Arvind Iyer, Lucas Joppa, Milind Tambe, et al. Spot poachers in action: Augmenting conservation drones with automatic detection in near real time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Bondi *et al.* 2020] Elizabeth Bondi, Raghav Jain, Palash Aggrawal, Saket Anand, Robert Hannaford, Ashish Kapoor, Jim Piavis, Shital Shah, Lucas Joppa, Bistra Dilkina, et al. Birdsai: A dataset for detection and tracking in aerial thermal infrared videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1747–1756, 2020.

[Caron *et al.* 2020] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[Caron *et al.* 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[Defard *et al.* 2021] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021.

[Dosovitskiy *et al.* 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[El-Nouby *et al.* 2021] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34, 2021.

[Emslie *et al.* 2016] Richard Emslie, Tom Milliken, Bibhab Talukdar, Susie Ellis, Keryn Adcock, and Michael Knight. *African and Asian Rhinoceroses – Status, Conservation and Trade*. Technical report, IUCN Species Survival Commission (IUCN SSC) African and Asian Rhino Specialist Groups and TRAFFIC, 2016.

[FruitPunch AI 2022] FruitPunch AI. *FruitPunch AI: AI for Wildlife Challenge*. https://fruitpunch.ai/ai-for-wildlife/, 2022. Accessed: 2022-04-01.

[Grill *et al.* 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[Gudovskiy *et al.* 2022] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022.

[He *et al.* 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[Huang *et al.* 2017] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7310–7311, 2017.

[Jaiswal *et al.* 2020] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[Jocher 2020] Glenn Jocher. *YOLOv5 by Ultralytics*, 5 2020.

[Kamminga *et al.* 2018] Jacob Kamminga, Eyuel Ayele, Nirvana Meratnia, and Paul Havinga. Poaching detection technologies—a survey. *Sensors*, 18(5):1474, 2018.

[Kellenberger *et al.* 2017] Benjamin Kellenberger, Michele Volpi, and Devis Tuia. Fast animal detection in uav images using convolutional neural networks. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 866–869. IEEE, 2017.

[Kellenberger *et al.* 2018] Benjamin Kellenberger, Diego Marcos, and Devis Tuia. Detecting mammals in uav images: Best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, 216:139–153, 2018.

[Kellenberger *et al.* 2019] Benjamin Kellenberger, Diego Marcos, and Devis Tuia. When a few clicks make all the difference: Improving weakly-supervised wildlife detection in uav images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[Kim *et al.* 2022] Jiwon Kim, Kwangrok Ryoo, Gyuseong Lee, Seokju Cho, Junyoung Seo, Daehwan Kim, Hansang Cho, and Seungryong Kim. Aggmatch: Aggregating pseudo labels for semi-supervised learning. *arXiv preprint arXiv:2201.10444*, 2022.

[Kisantal *et al.* 2019] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.

[Li *et al.* 2020] Minglei Li, Xingke Zhao, Jiasong Li, and Liangliang Nan. Comnet: Combinational neural network for object detection in uav-borne thermal images. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[Liu *et al.* 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[Liu *et al.* 2020] Mingjie Liu, Xianhao Wang, Anjian Zhou, Xiuyuan Fu, Yiwei Ma, and Changhao Piao. Uav-yolo: Small object detection on unmanned aerial vehicle perspective. *Sensors*, 20(8):2238, 2020.

[Liu *et al.* 2021] Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang. A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, page 114602, 2021.

[Misra and Maaten 2020] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[Moodley *et al.* 2017] Yoshan Moodley, Isa-Rita M Russo, Desiré L Dalton, Antoinette Kotzé, Shadrack Muya, Patricia Haubensak, Boglárka Bálint, Gopi K Munimanda, Caroline Deimel, Andrea Setzer, et al. Extinctions, genetic erosion and conservation options for the black rhinoceros (diceros bicornis). *Scientific Reports*, 7(1):1–16, 2017.

[Moreni *et al.* 2021] Mael Moreni, Jerome Theau, and Samuel Foucher. Train fast while reducing false positives: Improving animal classification performance using convolutional neural networks. *Geomatics*, 1(1):34–49, 2021.

[Redmon *et al.* 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[Reinhard *et al.* 2015] Friedrich Reinhard, Matthew Parkan, Timothée Produit, Sonja Betschart, Beatrice Bacchilega, Morgan Hauptfleisch, Patrick Meier, Stéphane Joost, and Devis Tuia. *Near real-time ultrahigh-resolution imaging from unmanned aerial vehicles for sustainable land use management and biodiversity conservation in semi-arid savanna under regional and global change (SAVMAP)*, 2015.

[Ren *et al.* 2016] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

[Rey *et al.* 2017] Nicolas Rey, Michele Volpi, Stéphane Joost, and Devis Tuia. Detecting animals in african savanna with uavs and the crowds. *Remote Sensing of Environment*, 200:341–351, 2017.

[SANParks 2020] SANParks. *South African National Parks Annual Report 2019/2020*. Technical report, 2020.

[Schneider *et al.* 2018] Stefan Schneider, Graham W Taylor, and Stefan Kremer. Deep learning object detection methods for ecological camera trap data. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 321–328. IEEE, 2018.

[Schofield *et al.* 2019] Daniel Schofield, Arsha Nagrani, Andrew Zisserman, Misato Hayashi, Tetsuro Matsuzawa, Dora Biro, and Susana Carvalho. Chimpanzee face recognition from videos in the wild using deep learning. *Science Advances*, 5(9), 2019.

[Shamsolmoali *et al.* 2021] Pourya Shamsolmoali, Jocelyn Chanussot, Masoumeh Zareapoor, Huiyu Zhou, and Jie Yang. Multipatch feature pyramid network for weakly supervised object detection in optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.

[Shao *et al.* 2022] Feifei Shao, Long Chen, Jian Shao, Wei Ji, Shaoning Xiao, Lu Ye, Yueting Zhuang, and Jun Xiao. Deep learning for weakly-supervised object detection and localization: A survey. *Neurocomputing*, 2022.

[Siméoni *et al.* 2021] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021.

[SPOTS 2022] SPOTS. *Strategic Protection Of Threatened Species*. http://www.spots.org.za//, 2022. Accessed: 2022-03-15.

[Tang *et al.* 2018] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):176–191, 2018.

[Touvron *et al.* 2021] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[Uijlings *et al.* 2013] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[Vaswani *et al.* 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wearn and Glover-Kapfer 2017] Oliver Wearn and Paul Glover-Kapfer. *Camera-trapping for conservation: a guide to best practices*. Technical report, WWF-UK, 2017.

[Yan *et al.* 2017] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, and Changshui Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint arXiv:1702.08740*, 2017.

[Yu *et al.* 2021] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021.

[Zavrtanik *et al.* 2021] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021.

[Zheng *et al.* 2021] Xiaochen Zheng, Benjamin Kellenberger, Rui Gong, Irena Hajnsek, and Devis Tuia. Self-supervised pretraining and controlled augmentation improve rare wildlife recognition in uav images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 732–741, 2021.

[Zitnick and Dollár 2014] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.