NTNU
Norwegian University of
Science and Technology

# Detecting Sheep in Drone Images by Deep Learning

Kari Meling Johannessen

20-12-2019

Specialization Project

Supervisors:

Professor Hongchao Fan
Professor Svein-Olaf Hvasshovd

# Abstract

This project examines the use of deep learning based object detection algorithms for automatic detection of sheep in drone images. Automatic detection of sheep can be useful to help farmers find their sheep at the end of the grazing season. Data of free ranging and fenced sheep is collected during three field trips to Storlidalen in Oppdal and Libra R-CNN object detection architecture is applied on the collected images. The model achieves an average precision of 93% with an IoU threshold of 0.5. At confidence threshold of 86%, a 91% precision and and 86% recall is achieved on the validation data set. This shows that automatic detection of sheep using deep learning can be a great complementary aid to existing solutions that help farmers keep track of their sheep.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Every summer, approximately 2.1 million sheep are released to graze freely on large pasture areas throughout Norway. When sheep farmers collect their sheep at the end of the grazing season, there are often a few stragglers that are extremely difficult to find. Electronic tracking solutions exist but they are expensive and dependent on mobile coverage. Some farmers use drones as a tool to search demanding terrain. However manual search for sheep is time consuming and prone to error.

With recent great advances in object detection using machine learning, automatic detection of sheep in drone images should be possible. This report outlines and tests how current state of the art deep learning based object detection algorithms can be used to detect sheep in visual drone images. Even though the drone used in this project captures both visual and infrared images, the scope of this project only examines the visual images.

# 2 Sheep grazing and roundup

This chapter gives some insight into how the sheep grazing process takes place today, which technologies currently exist and an overview of the solution envisaged by this project. The Norwegian agricultural cooperative Nortura, [4], recently released a thematic report, which gives information and recommendations on how to manage sheep during the grazing season. NTNU professor S-O. Hvasshovd, who has been researching modernisation of raising sheep for several years and actively carrying out research by participation has also given some insight into the processes surrounding sheep grazing and roundup.

## 2.1 The grazing season

Every summer, approximately 2.1 million sheep are released to graze freely on large pasture areas in Norway, [5]. On average, 40% of a sheep farmers feeding source comes from here. Outback sheep grazing also has the advantage that it prevents nature from overgrowing, which is important for maintaining the current biological diversity, [6].

During the grazing season, the farmers are required by law to inspect their sheep at least once a week and to keep documentation of these inspections, [6]. For this purpose, it is common that farmers form teams and share the inspection responsibilities. Since sheep do not always spread ideally over the grazing areas, injury and loss of oversight may occur. In dealing with this problem, some farmers hang posters to encourage people to share information about injured or lost sheep. Fences and strategically placed salt blocks may also help encourage sheep to graze in desired locations.

When lambs have reached a desired weight or when outdoor food resources are reduced in quality due to seasonal changes, it is time to collect the sheep. This is often a large event that requires a lot of manpower and time. Cooperation between farmers and help from family and neighbours is therefore crucial for a successful sheep collection. S-O. Hvasshovd describes the typical sheep round up as happening in three phases, [7]:

1. **Main roundup 1:** The first roundup phase often involves help from many people and sheep dogs and commonly runs over 1-2 weekends. In this time, approximately 90% of sheep are collected.
2. **Main roundup 2:** The farmer goes over the same area as in the first round up phase to collect the missed sheep. This phase can last a couple of weeks and the farmer has much less help. In this second round up 5% to 10% of sheep are collected.
3. **Collecting Stranglers:** Typically 10 to 20 sheep spread across 5-6 groups are left uncollected by this stage. These sheep have often wandered outside the typical grazing terrain and are especially hard to find. As a result, this phase can be very time consuming and a source of frustration for the farmer.

## 2.2 Existing Technology

There exist some technical solutions on the market today that farmers can use to keep track of their sheep. These are radio bells, electronic ear tags and drones.

### 2.2.1 Radio Bells

Radio Bells are commonly used to track sheep while they are grazing. Some examples of these products are *Smartbjella*, [8], *Findmy*, [9] and *Telespor*, [10]. A summary of the price and features of these products

| | Smartbjella | Telespor | Findmy |
|---|---|---|---|
| **Price per tracker (NOK)** | 990 | 1124 | 1590 |
| **Seasonal Subscription cost (NOK)** | 99 | 124 | 229 |
| **Battery life (years)** | 10 | 1 | 3 |
| **Tracking Technology** | NB-IoT,GPS | LTE-M, NB-IoT ,GPS | low orbit satelite, GPS |
| **GPS tracking** | ✓ | ✓ | ✓ |
| **Geofencing** | ✓ | ✓ | ✓ |
| **Movement alarm** | ✓ | ✓ | ✓ |
| **Stress warning** | | | ✓ |

**Table 1:** *Overview of existing radio bell tracking products. A checkmark indicates that the feature is offered by the product*

are given in table 1. These products work by having the sheep wear a GPS and radio tracking collar that transmits their location. Using this, sheep farmers can get an overview of their sheeps' current and historical locations via a website or application. Other features offered by these products include geofencing, movement alarm and stress warning.

Radio bell tracking of sheep can be very advantageous to farmers as long as they are within an area that has mobile coverage. However, as shown in table 1, the price per unit is quite high. Due to the high price, a common way to use these products is to only track a portion of the total sheep herd and assume that the sheep will stay in groups. Another limitation with this tracker is that it cannot be used on lam. This is because lam cannot wear traditional collars as they are growing.

#### Tracking Technology

*Smartbjella* and *Telespor* track sheep using Narrowband Internet of Things (NB-IoT), which the providers claim covers larger areas than regular mobile internet such as 4G. The network is currently being expanded in Norway but there are still large areas that are missing coverage. *Findmy* uses low orbit satellite technology for communication and as a result is not dependent on internet coverage. This technology will work as long as the sheep are outdoors with a free view to the sky.

#### Geofencing

Geofencing works by having the user draw a virtual fence on a map and if one of the tracked sheep enters or leaves this area then the farmer is notified.

#### Movement Alarm

Movement alarm will send a notification if a tracked sheep has been immobile over a longer period of time, typically 48 hours. This can be useful in order to detect sheep that are injured, stuck or dead.

#### Stress Warning

Stress warning is a feature only offered by *Findmy*. The product analyses movement patterns of the sheep and is able to detect stressful movement behaviour in a herd. This can be useful as it will give an indication to the farmer that something is scaring the sheep and as a result, extraordinary measures can be taken to protect the sheep.

### 2.2.2 Electronic Ear Tags

Another common aid that can be used to track sheep is an electronic ear tag. By placing readers of these ear tags at a strategical location such as a salt block, farmers can get information about which sheep have 'checked in' to these locations. These are very affordable, costing under NOK 20 per unit. In addition to the tags, it is also necessary to have a radio at the location to transmit the information to the farmer.

### 2.2.3 Drones

In an article by 'Norsk Sau og Gjeit' (NSG) [11], drones are pointed out as being a cheap and effective technology for observing sheep during the grazing season. Drones can easily fly over and capture images of challenging terrain. This can save the farmers valuable time and energy. However, currently the process of searching for sheep in drone images is manual, which can be time consuming, boring and error prone.

## 2.3 Envisioned Solution

Radio Bells are a useful technology to help farmers locate their sheep, however as discussed in section 2.2 their use is limited by mobile coverage requirements and a high price. Drones are also a useful technology being employed today, however detection of sheep in drone images is not yet automated. As a result, an alternative solution such as the one in figure 1 could potentially be useful in helping the sheep farmers to find the last of their sheep. This system involves three steps:

1. A drone is used to capture aerial images of the area of interest.
2. The images are processed by an object detection algorithm to locate potential sheep.
3. When processing is complete, the farmer can access the results through a graphical interface.

This system has some advantages over current technologies. Firstly, drones are relatively cheap. As an example, the DJI Mavic 2 Dual Enterprise(regular and Infrared camera), [12] that is used in this project can be purchased for approximately NOK 30000. The same drone without the infrared (IR) camera can be bought for approximately NOK 12000. Other advantages are that this system does not rely on mobile coverage and that the sheep do not have to wear an expensive tracker.

On the other hand, this system does have some vulnerabilities. One issue is that detection likelihood is reduced if sheep are occluded by other objects such as trees. Moreover, image quality is dependent on weather conditions. If it is too sunny or too dark then this technique will not work. Object detection algorithms are quite good but they are not perfect so there is a risk of finding false positives (for instance mistakenly classifying a rock as a sheep).

The strength of this system compensates for some weaknesses of existing solutions, however it does have some of it's own weaknesses. As a result, it is imagined that this solution could be used in combination with existing technologies.

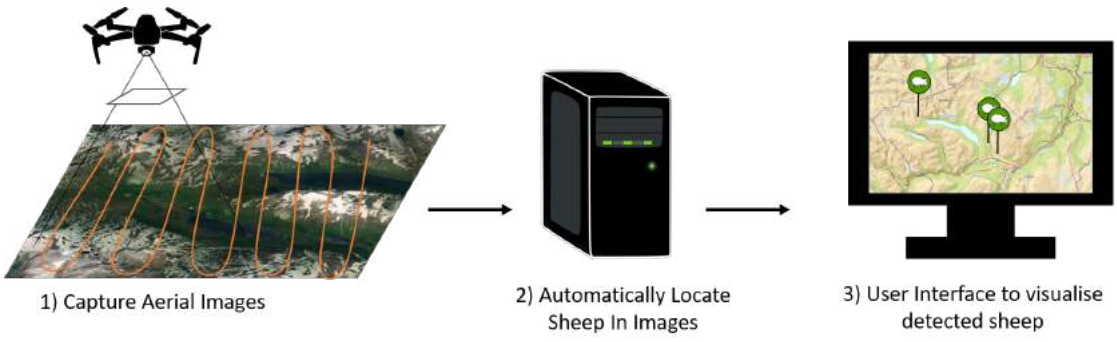


**Figure 1:** *Envisioned system to find sheep.*

# 3 Theory - Object detection

Object detection is the process of locating and classifying objects in an image, [13]. Figure 2 shows an example of the desired result of applying an object detection algorithm on an image to detect sheep. An object detection algorithm takes an image as input and outputs bounding boxes for all the instances found in the image. For each output bounding box, the detector will also predict which class the object belongs to. In other words, object detection algorithms answer the question: *What objects and where?*, [14]. Object detection has a wide range of use cases. Facial detection, self driving cars and visual search engines are just some examples of where object detection is being used, [15].

Object detection progress can be classified in two historical periods “traditional object detection period (before 2014)” and “deep learning based detection period (after 2014)”, [14]. Traditional object detection methods are greatly based on sophisticated hand crafted feature detection. By 2012, performance using these methods had begun to stagnate, [14][16]. In 2012, A. Krizhevsky *et al.* designed Alex-net, which drastically improved performance in image classification by using a convolutional neural network (CNN). CNNs are a form of deep learning and are effective at learning robust, high level features of an image. In 2014, R. Girshick *et al.* proposed the R-CNN algorithm for object detection, [16], which was among the first methods to utilise the feature detecting abilities of CNNs for object detection. This was extremely effective and as a result, all modern state of the art object detection algorithms now utilise deep learning.

In this chapter, the theory behind deep learning based object detection algorithms is discussed. In addition, some of the existing state of the art (SoTA) methods are presented.

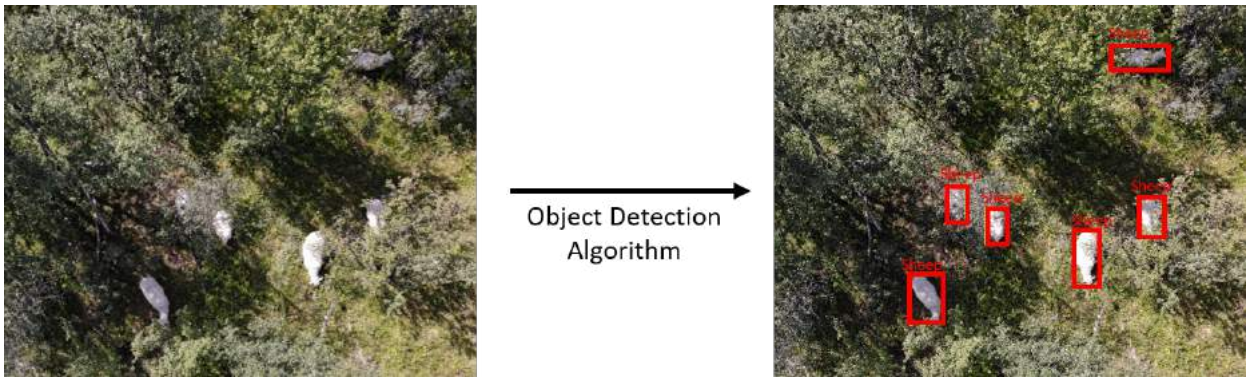


**Figure 2:** *Object Detection to identify and locate sheep.*

## 3.1 Object detection Metrics

In order to evaluate and compare the performance of object detection algorithms, it is necessary to use a common metric. Average Precision is most commonly used and is defined as the average precision over a set of evenly spaced recall values, [14]. Average precision (AP) is a good metric because it takes both precision and recall into account.

The quality of a predicted bounding box is judged by its degree of overlap with a given ground truth bounding box. This is called intersection over union (IoU) and is simply the intersection area divided by the union area as shown in figure 3.

The IoU can be used to classify each predicted bounding box as either a true positive (TP) or a false positive (FP). A predicted bounding box is a TP if it has an IoU of more than a given threshold value with a

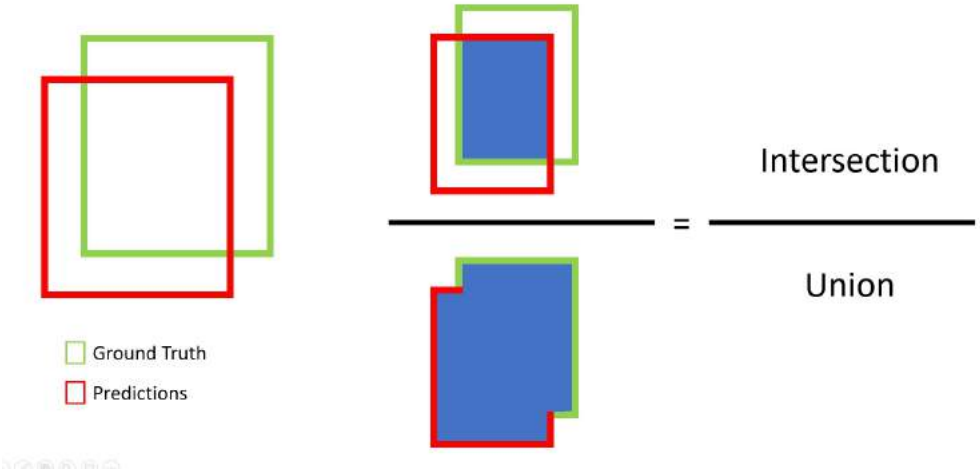


**Figure 3:** *Intersection over Union (IoU).*

ground truth box and it is a FP if it has an IoU of less than the threshold. A false negative (FN) occurs if the detector fails to detect a ground truth bounding box. Figure 4 shows some examples of occurences of TPs, FPs and FNs. True negatives (TN) is the case where the detector correctly did not predict a bounding box. This is not considered because it is a difficult thing to quantify.

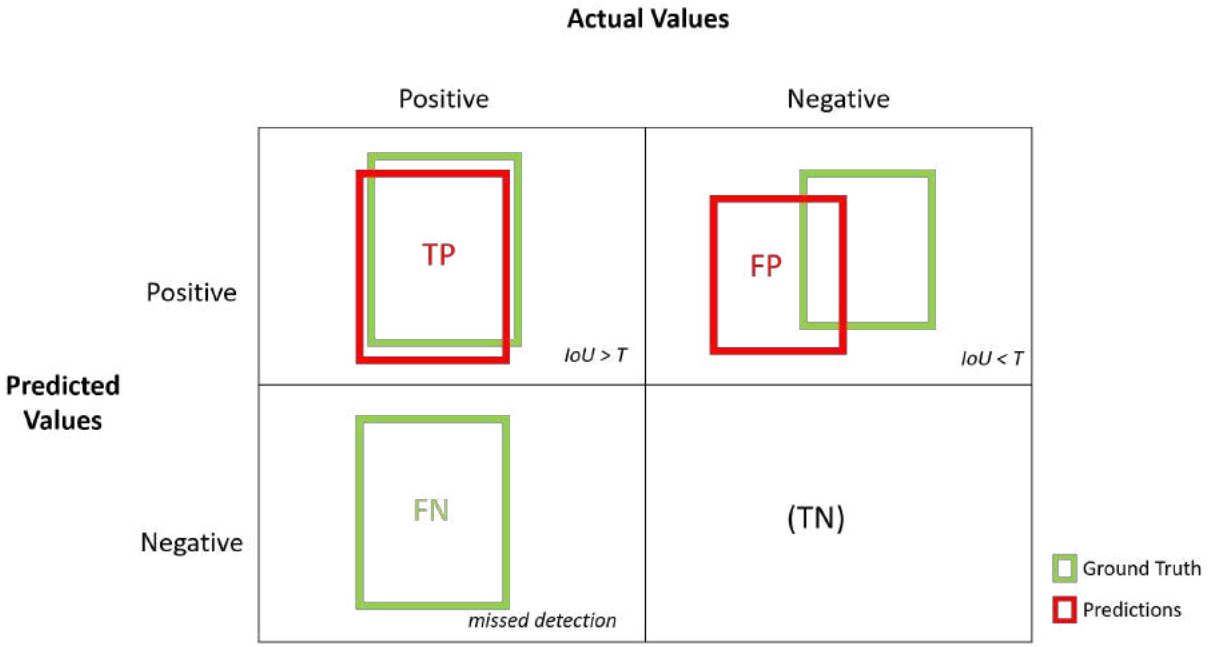


**Figure 4:** *Examples of true positive (TP), false positive (FP) and false negative (FN).*

Precision is expressed by equation 3.1 and reveals the proportion of predictions that were correct. However, a weakness with precision as a detection metric is that it does not take into account all the detections that were missed by the object detector (FNs).

$$Precision = \frac{TP}{TP + FP} = \frac{Correct\ predictions}{All\ predictions} \tag{3.1}$$

In comparison, recall does take FNs into account and is expressed in equation 3.2. Recall reveals the proportion of objects that the detector was able to detect. In order to improve recall, one could simply increase the number of predictions since this would increase the chance of detecting an object. However,

making more predictions would likely increase the number of FPs, which would be detrimental to the precision value. This trade off between precision and recall is clearly demonstrated in figure 5.

$$Recall = \frac{TP}{TP + FN} = \frac{Correct\ predictions}{All\ ground\ truths} \tag{3.2}$$

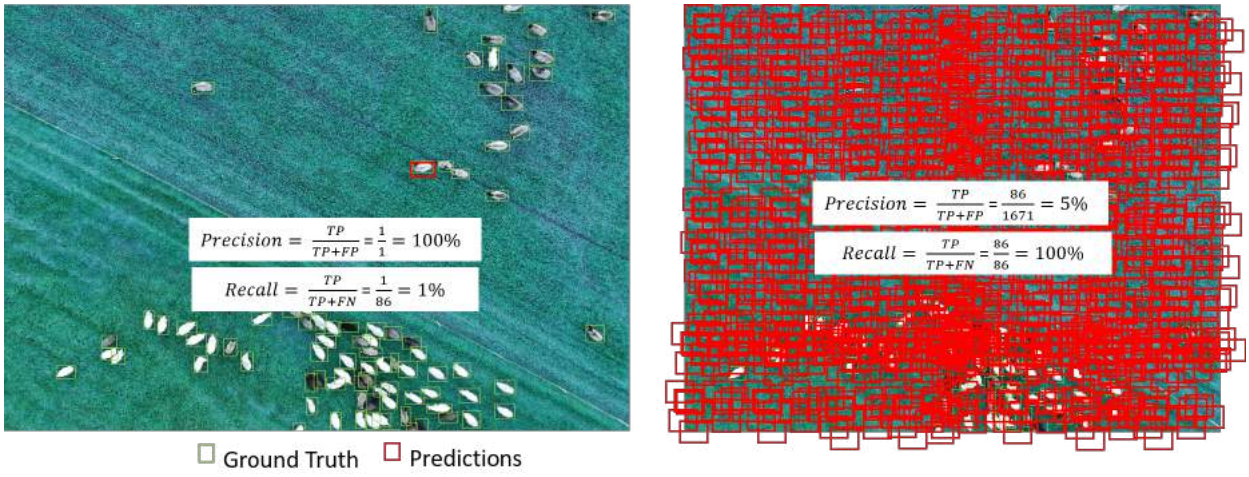


**Figure 5:** *Tradeoff between precison and recall. Image on the left has high precision but low recall. Image on the right has low precision but high recall.*

A good object detector should perform well on both precision and recall. As a result, Average precision is a good metric as it is average precision values over a range of recall values. A common IoU threshold value to use is 0.5. When the IoU threshold is set to 0.5, the AP is referred to as $AP_{50}$. Recently, it is also common to also consider $AP_{50:95}$, which is the average AP over IoU thresholds ranging from 0.5 (course localization) to 0.95 (perfect localisation). Either metric is suitable, depending on how accurate the bounding boxes are required to be. If an object detector detects multiple classes, then a common metric used to describe detection is mean average precision (mAP), which is simply the average AP of all the classes.

### 3.1.1 Detection data sets

There are three data sets that are commonly used to benchmark object detection results. These are Pascal VOC2007, Pascal VOC2012, [17] and Microsoft COCO (MSCOCO), [18].

#### VOC2012

The VOC2012 data set has 20 categories and images are divided into training, validation and test splits with 2501, 2510 and 5011 images respectively. $AP_{50}$ is used as the detection metric.

#### VOC2017

The VOC2017 data set has 20 categories and as with VOC2012, the images are divided into training, validation and test splits with 5717, 5823 and 10991 images respectively. Annotations for the test data set are not publicly available. $AP_{50}$ is used as the detection metric.

#### Microsoft COCO

The COCO data set is a large data set with 80 object categories. The data set consists of 118287, 5000 and 40670 labelled images in the training, validation and test sets respectively. Annotations for the test data set are not publicly available. The coco detection challenge uses a range of AP variants as detection metrics:

- $AP_{50:95}$: mAP averaged over ten IoU thresholds: 0.5 : 0.95
- $AP_{50}$: mAP at 0.50 IoU threshold
- $AP_{75}$: mAP at 0.75 IoU threshold
- $AP_S$: $AP_{50:95}$ for small objects of area smaller than $32^2$
- $AP_M$: $AP_{50:95}$ for objects of area between $32^2$ and $96^2$
- $AP_L$: $AP_{50:95}$ for objects larger than $96^2$

## 3.2 Object Detection With Convolutional Neural Networks (CNN)

A CNN is a type of neural network that is especially effective at detecting features in images. Since the introduction of utilising CNNs for object detection in 2014, this has become the predominant method of object detection. Object detection using CNNs can be grouped into two categories: multi-stage detectors and single stage-detectors, [14]. The R-CNN family of algorithm is an example of a multi-stage detector and it involves two steps: Region proposal and classification. In contrast, Single-stage detectors skip the region proposal stage and run detections in just one stage. YOLO, SSD and RetinaNet are examples of single stage detectors. In general, multi-stage detectors often achieve slightly better detection performance, while single-stage detectors are more time efficient, [19].

There has been a great development in the object detection field since the first deep learning based detection algorithms where developed in 2014, [14]. Sections 3.2.1 and 3.2.2 explain what contributions to the field were made by some of the most influential detection algorithms.

### 3.2.1 Influential Multi-stage detectors

#### R-CNN

R-CNN, [16] was among the first algorithms to use deep learning for object detection and the idea behind it was simple. First, 2000 regions are proposed using a classical region proposal technique called selective search. Each of these regions are then warped to a fixed size and fed to a CNN. Finally, the output from the CNN is fed to a linear Support-vector machine classifier, which determines the object class.

At the time of release, R-CNN improved previous best mAP results on the VOC2012 data set, [20] by more than 30%. However, the algorithm had some clear disadvantages. The main disadvantage was that it was very slow because the 2000 regions were processed independently through the CNN despite having a great deal of overlapping features.

#### Fast R-CNN

In 2015, R. Girshik proposed Fast R-CNN, [21], which addressed the main issues of R-CNN by proccesing each image only once through the CNN and then projecting the regions proposed by selective search onto the already computed feature map. Finally, each region of the feature map was sent to two fully connected layers that output the class scores and bounding box regressions.

Fast R-CNN was 10x faster to train than the original R-CNN. At test time, it could process an image in 2.3 seconds. In Fast R-CNN, the processing time was bottlenecked by the selective search region proposal stage.

#### Faster R-CNN

In 2015 S. Ren *et al.* proposed Faster R-CNN, [22], which addressed the main issue of Fast R-CNN by introducing a region proposal network to replace the expensive selective search process.

This was very effective since the RPN was also able to profit from the convolutional features generated by CNN backbone. Faster R-CNN reduced proposal time from 2.3 seconds to only 0.2 seconds and also

improved mAP performance.

#### Feature Pyramid Networks

In 2016, T.-Y. Lin *et al.* proposed using Feature Pyramid Networks (FPN) for Object Detection, [23]. FPN exploits the natural pyramid structure of CNNs in order to improve detection over multiple scales. In general, this is done by combining low-resolution, semantically weak features with high-resolution, semantically strong features in order to generate semantically strong features with high resolution. By using FPN with a basic Faster R-CNN system, the authors were able to achieve better performance on the COCO detection challenge than all other existing single-model entries. Most of the SoTA systems today (both multi-stage and single-stage) incorporate FPN.

#### Cascade R-CNN

In 2017, Z. Cai *et al.* proposed cascade R-CNN, [24]. Cascade R-CNN is a multi-stage detector that involves training a sequence of detectors with increasing IoU thresholds. This has the effect of improving the quality of detection for all IoU values but especially for stricter IoU levels. The architecture achieved SoTA on the COCO detection challenge. The authors propose that this cascading architecture is applicable and advantageous across detector architectures and not just for R-CNN.

#### Libra R-CNN

Libra R-CNN, [25] was proposed in 2019 by J. Pang *et al.* Libra R-CNN is a framework that aims to balance the training process in three stages: sampling stage, feature stage and objective stage. By doing this, detection mAP performance is improved by significantly compared to FPN Faster R-CNN. The framework was also successfully applied on the single stage architecture RetinaNet but then without the balanced sampling procedure (as retinaNet does not have a sampling step).

### 3.2.2 Influential Single-stage detectors

#### You Only Look Once(YOLO v1, v2, v3)

YOLO, [26] was proposed by J. Redmon *et al.* in 2015 and was the first single-stage deep learning based detector. In YOLO, bounding boxes and class probabilities are predicted directly by a single CNN. YOLO works by splitting the image into an SxS grid and for each grid cell, B bounding boxes and corresponding class probabilities are predicted. The greatest advantage of YOLO is that it is very fast and it generalises well. Another general advantage of single-stage detectors over multistage detectors is that they 'see' the entire image, which allows them to reason globally and encode contextual information about the image. Because of this, YOLO demonstrates fewer false positives. The original YOLO had some issues. Each grid cell could only predict one object and as a result, the algorithm struggled to detect small objects that were very close to each other. Compared to existing region based methods, YOLO had a higher localisation error and a lower Recall.

In 2016, J. Redmon proposed YOLOv2, [27], which introduced some improvements to the original YOLO. The main improvements of YOLOv2 was adding batch Normalization layers, a slightly deeper network and the use of Anchor Boxes. YOLOv2 was better and faster than the original but still struggled with small objects. YOLOv3, [28], released in 2018 by the same authors dealt with the problem of detecting small objects by implementing multi-scale detection with a process similar to FPN. In addition, YOLOv3 used a deeper network than YOLOv2. Despite a deeper network and multi-scale predictions, YOLOv3 is still very fast. YOLOv3 is about 3.8x faster than its counterpart RetinaNet but still achieves very similar $mAP_{50}$ score. YOLOv3 certainly has a great speed-accuracy trade-off and is therefore a very popular method for

applications that require real time processing. On the other hand, YOLOv3 struggles to get boxes perfectly aligned, so it does achieve a lower $mAP_{0.5:0.95}$ score than other SoTA detectors.

#### RetinaNet

In 2017, T.-Y. Lin *et al.* proposed RetinaNet, [29]. The designers of RetinaNet identified that extreme foreground-background imbalance during training was a cause for why single stage detectors had previously achieved inferior accuracy compared to multi-stage detectors. RetinaNet deals with this problem by introducing Focal Loss, which weights down the loss for well classified examples. In addition, RetinaNet also utilises FPN in its design. In doing this, RetinaNet was able to match the speed of other single-stage detectors and surpass the accuracy of all other existing SoTA detectors.

#### Fully Convolutional One-Stage Object Detection (FCOS)

In 2019, Z. Tian *et al.* proposed FCOS [30]. FCOS is different from other SoTA detectors since it is anchor box free. Instead of anchor boxes, FCOS outputs a 4D vector for each pixel in the image, which describes the pixels position relative to the edge of a bounding box. The advantage of this is that complicated calculations such as IoU are not needed. Moreover, tuning of hyper parameters related to anchor-boxes are also not needed. The authors report that FCOS surpasses performance of previous single stage detectors with the advantage of being simpler.

### 3.2.3 State of the Art (SoTA) Object Detection Algorithms

So which object detection algorithm is best? It turns out that this is a very difficult question to answer. Firstly, it is difficult to do a direct comparison of the different methods as reported by their respective research papers because they are all trained under different conditions (eg. different backbone network or different resolutions). Secondly, the definition of best is dependent on the desired application of the detector. Choice of detection method must always take into account the speed accuracy trade-off between the various design choices. Moreover, consideration must be made concerning how well the bounding boxes need to align with the objects. Is a course detection sufficient (IoU $>$ 0.5) or do the boxes need to be perfect (IoU $>$ 0.95)?. Finally, object detection is evolving so rapidly that comparisons quickly become invalid.

For the purpose of this project, where the goal is to detect sheep in aerial images, real time processing speed is not required. Moreover, bounding boxes do not need to be perfect so $AP_{50}$ performance is more important than $AP_{50:95}$. MMDetection, [2] is an object detection and segmentation toolbox developed from code by the winners of the COCO Challenge 2018. It includes implementations of some popular and modern detection methods and is continually being updated. In their technical report, they present the results that each of their implemented models achieve on COCO val2017 data set. These results can be seen in table 2. Although the MMdetection framework does not represent all current SoTA models, it does have many relevant models that are trained under the same conditions and at the same scales. As a result, this allows for a more controlled comparison of methods. As shown in the table, Cascade R-CNN with Resnext 101 64x4d backbone has the highest reported $AP_{50:95}$. For $AP_{50}$, libra Faster R-CNN with Resnext 101 64x4d backbone has the best performance, closely followed by cascade R-CNN.

| **Method** | **Backbone** | **Lr Sched** | $AP_{50:95}$ | $AP_{50}$ |
|---|---|---|---|---|
| Faster R-CNN | R-50 | 1x | 36.4 | 58.4 |
| | R-101 | 1x | 38.5 | 60.3 |
| | X-101-32x4d | 1x | 40.1 | 62.0 |
| | X-101-64x4d | 1x | 41.3 | 63.3 |
| Cascade R-CNN | R-50 | 1x | 40.4 | 58.5 |
| | R-101 | 1x | 42.0 | 60.3 |
| | X-101-32x4d | 1x | 43.6 | 62.2 |
| | X-101-64x4d | 1x | **44.5** | 63.3 |
| SSD300 | VGG16 | 120e | 25.7 | 43.9 |
| SSD512 | VGG16 | 120e | 29.3 | 49.2 |
| RetinaNet | R-50 | 1x | 35.6 | 55.5 |
| | R-101 | 1x | 37.7 | 57.5 |
| | X-101-32x4d | 1x | 39.0 | 59.4 |
| | X-101-64x4d | 1x | 40.0 | 60.9 |
| RetinaNet-GHM | R-50 | 1x | 36.9 | 55.5 |
| | R-101 | 1x | 39.0 | 57.7 |
| | X-101-32x4d | 1x | 40.5 | 59.7 |
| | X-101-64x4d | 1x | 41.6 | 61.3 |
| FCOS | R-50(c) | 1x | 36.7 | 55.8 |
| | R-101(c) | 1x | 39.1 | 58.5 |
| Libra Faster R-CNN | R-50 | 1x | 38.5 | 59.5 |
| | R-101 | 1x | 40.3 | 61.2 |
| | X-101-32x4d | 1x | 41.6 | 62.7 |
| | X-101-64x4d | 1x | 42.7 | **63.8** |

**Table 2:** *Results reported by MMdetection, [2], of different detection methods on COCO val2017. R-50 and R-50 (c) denote pytorch-style and caffe-style ResNet-50 backbone. X-101-64x4d refers to the resnext 101 64x4d backbone.*

# 4 Requirements

In this project, object detection using deep learning is applied to drone images of sheep. This detection is envisioned to be used to detect sheep from aerial imagery of potential areas where sheep are grazing in order to help farmers locate their sheep for collection at the end of the season. Further description of the envisioned system can be found in section 2.3 of this report. With this use case in mind, some requirements for an object detector can be set. An overview of these requirements can be seen in table 3. In a fully operational system, all requirements must be met, however due to time and data restrictions, not all requirements are considered by this project.

**R1: Bounding Box Quality**

Bounding box quality refers to the required predicted bounding box overlap with a ground truth bounding box. This metric is called intersection over union (IoU) and is explained in more detail in section 3.1. For the use case of this project, perfectly fitting bounding boxes are not necessary. Course fitting boxes (IoU $>$ 0.5) are sufficient.

**R2: Precision and Recall**

The object detector should aim to maximise precision and recall. Precision and Recall are explained in more detail in section 3.1. In short, precision describes how many of the predicted sheep are actually sheep and recall describes what proportion of the total number of sheep in the data set were found by the detector. Average precision (AP) is the average precision over a range of recall values and is therefore a suitable metric that takes both precision and recall into account. Since course bounding boxes are sufficient for the desired use case, $AP_{50}$ is the metric that the detector should aim to maximise. In the big picture, a lower recall can be acceptable because sheep often travel in groups and for the desired use case it is enough for the system to identify just one sheep in the herd in order for the farmer to find the entire group.

**R3: Time**

The desired use case does not require real-time processing so time is not a major limiting factor. Nonetheless processing time should be 'reasonable'. Further investigation to determine what processing time is considered reasonable is required. This could for instance be quantified as square meters covered per second ($m^2/s$). However, for simplicity no time requirement is set in this project.

**R4: Image Quality**

Since image quality can vary based on factors such weather conditions and drone movement, the object detector should be robust against variances in image sharpens, brightness and contrast.

**R5: Sheep variation**

In Norway, sheep come in a range of shapes, sizes and colour. Figure 6 shows some examples of how the different races of sheep can appear. As a result, the object detector should be robust to variation in sheep shape, size, race and colour.

**Figure 6:** *Some examples of sheep races that exist in Norway, [1].*

#### R6: Terrain/background Variation

Sheep can be found in various terrain. The object detector should be able to find the sheep regardless of where they are. This includes rocky areas, area with many trees, on the road, in the snow or anywhere else that the sheep may be, as long as they can be seen from the drone.

#### R7: Other animals

The object detector should be able to distinguish sheep from other animals that could be found in the area of interest. As with the time requirement (R3), this is an important feature of a finished product but will not be considered in this project due to lack of time and relevant data.

| ID | | Requirement |
|---|---|---|
| R1 | ✓ | Predicted bounding boxes should match groundtruth bounding boxes by an IoU > 0.5 |
| R2 | ✓ | The detector should aim to maximize $AP_{50}$ |
| R3 | | The detector should process an image within a reasonable time |
| R4 | ✓ | The detector should be robust with regards to sharpness brightness and contrast |
| R5 | ✓ | The detector should be robust to variation in sheep shape, size and colour |
| R6 | ✓ | The detector should be able to detect sheep regardless of terrain, background or weather. |
| R7 | | The detector should be able to distinguish sheep from other animals in the areas of interest |

**Table 3:** *Object Detection Requirements. Requirements marked with a checkmark are considered in this project.*

# 5 Method

This chapter explains the method used for gathering data, training the deep learning object detector and applying the object detector on images.

## 5.1 Data Collection

### 5.1.1 Equipment

In order to capture images of sheep and perform object detection, a camera drone is needed. The drone used is the DJI Mavic 2 Enterprise Dual (M2ED).

#### DJI Mavic 2 Enterprise Dual (M2ED)

The M2ED, [12] (shown in figure 7) is the drone used for photographing the sheep. This drone is a compact and foldable drone with dual sensors that allow it to capture side-by-side visible and IR images. More detailed specifications are listed in table 4. This drone is chosen for several reasons:

- It is believed that IR technology can be a great aid in identifying warm, live animals in their relatively colder surroundings.
- It is an affordable drone. The market price of the M2ED is approximately NOK $30000$, which is a price that an average Norwegian farmer would be able to afford.
- It is light and compact, which allows it to easily be transported in challenging terrain.
- It is easy to pilot.

A disadvantage of this drone is the low resolution of the IR camera, which may limit how high the drone can fly while still being able to capture sheep in the IR images.

| | |
|---|---|
| **Take-off weight:** | 899g (without accessories) |
| **Dimensions:** | Folded: 214mm x 91mm x 84mm. |
| | Unfolded: 322mm x 242mm x 84mm |
| **Max flight time:** | 31 mins |
| **Max speed:** | 72kph |
| **Thermal sensor:** | FLIR Lepton |
| **Thermal camera resolution** | 160 x 120 |
| **Visual camera resolution** | 3040 x 4056 |
| **Visual camera Field of view (FOV)** | Approx. 85° |
| **Visual camera 35 mm format equivalent** | 24mm |

**Table 4:** *DJI Mavic 2 Enterprise Dual (M2ED) specifications.*

**Figure 7:** *DJI Mavic 2 Enterprise Dual (M2ED).*

### 5.1.2 Method

Drone photographs of sheep are collected during three separate field trips to Storlidalen valley in Oppdal. It is beneficial to collect images under different conditions because this gives more variation in the data and therefore helps to achieve requirement R6 (chapter 4). During the data collection, images of sheep with different appearances are to be collected in order to achieve high variation in compliance with requirement R5.

Whilst collecting the data, visibility of the sheep in IR and visual images should be recorded at different flight heights ranging from 10m to 100m. Temperature and weather conditions should also be noted.

## 5.2 Training the Object Detector

### 5.2.1 Environment

#### Hardware

Training was performed on a computer with the following Graphical Processing Units (GPUs) and Central Processing Unit (CPU):

- 2 GeForce GTX 1080 GPUs
- Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz

#### Software Environment

Based on the requirements for the object detector outlined in the chapter 4, some design choices are made regarding choice of software frameworks and architectures.

Firstly, **MMdetection** [2] was chosen to use as the starter code. MMdetection is an open source object detection toolbox that includes a range of common object detection algorithms. The toolbox stems from the code base developed by the winners of the COCO detection challenge in 2018 and is continuously being maintained by researchers from the Multimedia Laboratory at the Chinese University of Hong Kong, [31]. Some of the results produced from this toolbox are shown in table 2. The reasons why this toolbox is chosen

are:

- It supports a range of popular object detection methods including SoTA methods
- The modular design makes it easy to modify.
- It is easy to get started using this toolbox.

MMdetection is built with **PyTorch**, [32], which is a popular open source machine learning framework that enables powerful tensor computations on Graphical Processing Units (GPU). PyTorch is mainly developed by the Facebook AI research lab and has a python interface.

**Libra R-CNN** is an object detection model. This model is described in more detail in section 3.2.1. It is chosen to use for this project because it has the highest reported $AP_{50}$ results of all the models presented in the MMdetection paper (see table 2). On a lower level, the following environment is used:

- Python 3.7.4
- PyTorch 1.3.0
- CUDA 10.1

## 5.3 Data Preprocessing

#### Image Sampling

It is necessary to filter out images that are too similar to each other in order to avoid over training on certain images. Images should vary by sheep being in different positions or by varying environmental conditions such as the lighting conditions. In addition, only images that contain sheep are to be kept. This can be justified by the fact that the sheep in the images are small compared to their background so the network will still have enough exposure to background samples.

#### Labelling

Labelbox, [33] is used for annotating the images with bounding boxes. Figure 8 shows a screen shot from the tool in use. Labelbox is a useful labelling tool that allows for easy labelling collaboration. In addition to just bounding boxes, it also allows the user to save features about the object. For each labelled sheep, it's colour is to be recorded. Labelbox also has a GraphQL API that can be used to access label information and for uploading new labels. This is very useful as it makes it possible to efficiently use the current best object detection model to predict bounding boxes for new images and then upload these to labelbox via their API. These labels of course need to be manually checked but a lot of time can be saved from not having to manually label many of the sheep.

#### Split into Training and Validation sets

When training a neural network, it is common to split the data into a training, validation and test set. The training set is used to train the network, the validation set is used for monitoring the training process to avoid overfitting and finally, the test set is used to evaluate the performance of the model. These three data sets should be completely independent from one another. It is also important that the data sets are varied enough to be representative of the scope of this project. Therefore it should be considered if there is enough varied data to have a separate test data set.

#### Image Classification by Sheep Size

The images should also be classified by the median sheep size in the image. The median sheep size is a proxy measure for the drone flight height since sheep size will decrease linearly with increased flight height


**Figure 8:** *Screenshot from labelling toolbox Labelbox.*

by the pinhole camera model, [3]. Sheep size is defined by the length of the diagonal of the bounding box as shown in figure 9.

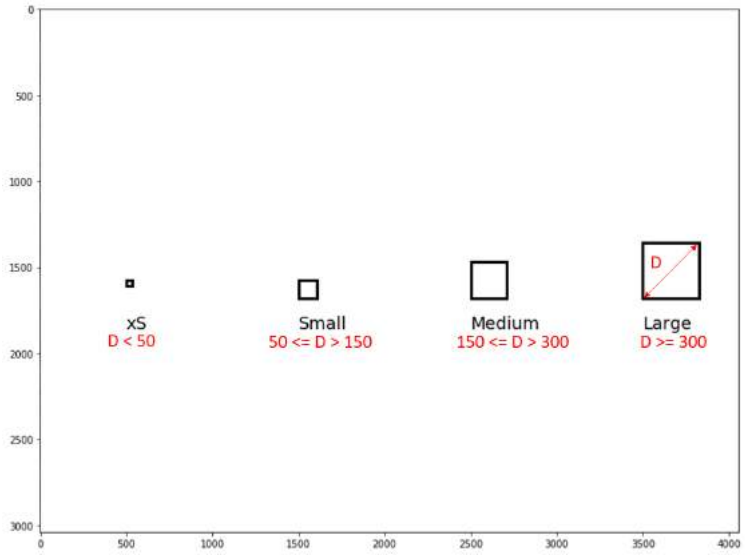


**Figure 9:** *Definition of extra small (xS), small, medium and large sheep. Boxes are shown in relation to the full image dimension of 3040 × 4056. Sheep labels are classified as one of these by the diagonal length (D) of their bounding box.*

### Cropping

Since the aim of this project is to maximise $AP_{50}$ performance without considering test and training time, the deepest neural network base model (ResNeXt-101-64x4d) and the original image resolution is used. The GPU memory can be a limiting factor, which means that the input images need to be sent to the network in parts. The maximum image size that will fit in the GPU memory at one time with this backbone network is 1024x1024. Therefore, crops of this size should be prepared. Figure 10 shows how crops are made from the images. For variation purposes, crops with 50% overlap as well as rotated crops with 50% overlaps are made for the training set. Only non-rotated crops are used for the validation set. More differently scaled crops are not necessary because the FPN and the anchor boxes built in to the network architecture assures scale invariance.

The result of cropping may leave significantly more background samples than samples containing sheep. It is experimented with having different background to sheep ratios in the training set.

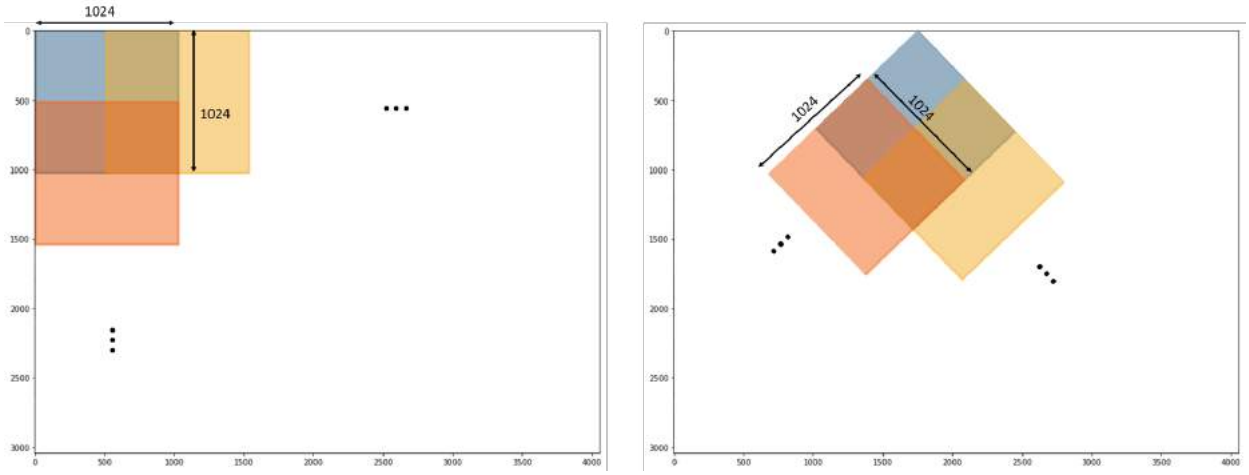


**Figure 10:** *Crops used to train the network. Left: 1024 × 1024 crops with 50% overlap. Right: 45° rotated 1024×1024 crops with 50% overlap.*

### 5.3.1 Training

MMdetection comes with some easily customisable configuration files where most of the relevant parameters can be set. Examples of these parameters are network architecture details such as network dimensions, anchor box scales and ratios, loss functions, number of classes and much more. Most of these parameters are kept as recommended by the framework, however some are changed in order to better suit the custom data set. The most notable parameters that must be set are listed in table 5.

During the training process, training performance logs should be kept. These logs record loss values per $50^{th}$ iteration and validation $AP_{50}$ values per epoch.

### 5.3.2 Inference

When the model has finished training, the model parameters from the epoch with the highest $AP_{50}$ result on the validation set are used to make predictions on the test set. Since the model is trained on 1024×1024 crops, inference should also be done on 1024×1024 crops. 1024×1024 crops with 50% overlap are evaluated by the model and then all the predictions for one image are combined and duplicate predictions are removed by non maximum suppression (NMS). NMS is done by finding all bounding boxes that have and IoU greater than 0.5 with each other and discarding all but the one with the highest confidence value.

## 5.4 Performance Metrics

After Inference, the performance of the predicted bounding boxes are evaluated against the ground truth bounding boxes. $AP_{50}$ is the main performance metric since coarse predictions are good enough for the desired use case. As explained in section 3.1, $AP_{50}$ is calculated by taking the average precision from a an evenly spaced range of recall values from the precision recall curve.

In order to get the final set of predicted bounding boxes, it is necessary to set a confidence threshold that will determine which bounding boxes should be included and which should be excluded. This threshold should be chosen in such a way that the resulting bounding boxes have a desirable balance between precision and recall.This confidence threshold and the corresponding precision and recall values are reported in the results.

Besides $AP_{50}$, some additional AP performance metrics will be calculated. These are explained in table 6.

| | | |
|---|---|---|
| **Number of Classes** | 2 | The regions are either sheep or not. |
| **Image Scale** | $1024 \times 1024$ | This is the maximum scale that will fit in the GPU. |
| **Augmentation** | • Vertical Flip<br>• Rotation<br>• Random Gamma<br>• Random Contrast<br>• Random Brightness | Additional augmentation was added because there are some differences between the COCO data set that the template was designed for and this project's custom sheep data set. Vertical flip and rotations make sense for images taken from drones because these images have no defined up or down. The probability of vertical flip or rotation being applied is set to 0.5. Random gamma, contrast and brightness are applied to images with a probability of 0.2 in order to account for the varying image quality that occurs under the different weather conditions. (Extra augmentation was only applied to the training data set) |
| **Learning Rate** | 0.00075 | A higher learning rate was recommended, however this gave too noisy progression |
| **Images per GPU** | 1 | This was limited by GPU memory capacity |
| **Pretrained** | COCO | A pretrained model was used. This model was trained on the COCO data set. Starting with a pretrained model is advantageous because this enables the model to detect general image features from the start of training. |

**Table 5:** *Model Parameters.*

| | |
|---|---|
| $AP_{50}$ | The Average Precision of the detector with IoU threshold 0.5. |
| $AP_{75}$ | The Average Precision of the detector with IoU threshold 0.75. |
| $AP_{50:95}$ | The Average Precision averaged over ten IoU threshold from 0.5 to 0.95. |
| $AP_{50,xS}$ | The Average Precision of the detector with IoU threshold 0.5 for images with extra small sheep (diagonal length < 50 pixels). |
| $AP_{50,S}$ | The Average Precision of the detector with IoU threshold 0.5 for images with small sheep (50 <= diagonal length < 150 pixels) |
| $AP_{50,M}$ | The Average Precision of the detector with IoU threshold 0.5 for images with medium sheep (150 <= diagonal length < 300 pixels) |
| $AP_{50,L}$ | The Average Precision of the detector with IoU threshold 0.5 for images with large sheep (300pixels <= diagonal length ) |

**Table 6:** *Average Precision metrics reported.*

# 6 Results

This chapter presents results of data collection, results that were logged during training as well as results after inference for the best performing model. Examples of predictions made by the object detector on images in the validation data set are presented in appendix A.

## 6.1 Data Collection

Data gathering was performed in August, September and October of 2019. Conditions were very different for the three weekends and the most notable difference between the weekends was the presence of snow in the October sessions. In general, sheep were much easier to see in the IR images in October than in August. The opposite was true for the visual images. More details for each field work are given bellow.

#### August

Figure 11 shows a sample of the images collected by Magnus Guttormsen and Svein-Olaf Hvasshovd on the $21^{st}$ and $22^{nd}$ August 2019.

- **Number of Sessions:** 8
- **Free Ranging / Fenced:** All sheep ranging free.
- **Environment Conditions:** Green leaves and grass. Trees are dense with leaves.
- **Weather Conditions:** $\approx 10°C$, Cloudy (except 1 session with sun)
- **Sheep Visibility:** Visibility of sheep in IR images is low. At 30-40m flight height, the sheep are no longer observable in the IR images. Visibility of sheep in the visual images is better. White sheep are visible up to 70-100m, whilst the black sheep are visible up to about 40-50m

**Figure 11:** *Sample of images captured in August 2019.*

### September

Figure 12 shows a sample of the images collected by Magnus Guttormsen, Kari Meling Johannessen and Svein-Olaf Hvasshovd on the $20^{th}$, $21^{st}$ and $22^{nd}$ September.

- **Number of Sessions:** 5
- **Free Ranging / Fenced:** Most sheep grazing in confined, fenced area. Free ranging sheep were found at two locations.
- **Environment Conditions:** Same as August.
- **Weather Conditions:** Between $7°C$ and $15°C$, Mostly sunny
- **Sheep Visibility:** Visibility of sheep in IR images is better than in August. Sheep are visible in IR in images taken from heights up to 70-80m. Similar visibility of sheep in visual images as in August, however the reflection from the sun made it more difficult to distinguish sheep from rocks. The sun also caused large shadows to appear in the images.

**Figure 12:** *Sample of images captured in September 2019.*

### October

Figure 13 shows a sample of the images collected by Kari Meling Johannessen and Svein-Olaf Hvasshovd on the $25^{th}$ and $26^{th}$ October.

- **Number of Sessions:** 7
- **Free Ranging / Fenced:** All sheep grazing in confined, fenced areas. Much fewer sheep than in the previous month due to sheep having been sent to slaughter.
- **Environment Conditions:** Small layer of fresh snow, leaves have fallen off trees. However, 1 session on the way back from Storlidalen to Trondheim, had environmental conditions more similar to the previous month)
- **Weather Conditions:** Between $0°C$ and $3°C$, Mostly cloudy. One session with sun.
- **Sheep Visibility:** Visibility of sheep in IR images was significantly better under these conditions. Sheep could be seen in IR images taken from a height of up to 100m. On the other hand, observing sheep in the visible images was more difficult than the previous times due to the sheep blending with their environment.

**Figure 13:** *Sample of images captured in October 2019.*

## 6.2 Data Prepossessing

The data collection phase resulted in a total of 2381 images from 20 different sessions. Image variation between sessions was relatively large, however images within sessions were similar to each other since they depicted the same group of sheep in the same environment. After sampling the images, the image count was reduced to 631.

Ideally, results should be reported on an independent test set in order to present reliable metrics. However, a test set would have to sample its images from the training and validation sets, which would make these data sets less representative and thus also lower the validity of the results. Since the data set collected only consisted of 20 independent image groups, it was decided that there was not enough independent data to have a separate test set so the validation set was used for both validation and testing. Of the 20 independent image groups, it was decided to use 3 of these for validation and the remaining 17 for training. For the validation set, one session from each of the 3 different months was used and sheep varied in colour and grazing environment.

Table 7 shows an overview of the number of images in the training and validation data sets. In addition, the table shows the number of images in each of the median sheep size groups. The median sheep size is used as an indication of the drone flight height and also expresses the sheep resolution, which can be linked to the difficulty level. Table 8 shows the number of sheep and their colour distribution in the training and validation data sets.

| | | Median Sheep Size | | | |
|---|---|---|---|---|---|
| | **Total** | Extra Small | Small | Medium | Large |
| Training | 555 | 67 | 207 | 266 | 15 |
| Validation | 76 | - | 38 | 38 | - |
| All labelled | 631 | 67 | 245 | 304 | 15 |

**Table 7:** *Number of images by median sheep size in the training and validation data sets. (See figure 9 for definition of sheep size). Median sheep size is used as a proxy measurement for drone flight height since sheep size decreases linearly with increased flight height, [3].*

| | | Sheep Colour | | | |
|---|---|---|---|---|---|
| | **Total** | White | Grey | Black | Brown |
| Training | 4317 | 2348 | 1439 | 419 | 112 |
| Validation | 575 | 386 | 166 | 17 | 6 |
| All labelled | 4893 | 2734 | 1607 | 436 | 118 |

**Table 8:** *Number of sheep of various colours in the training and validation data sets.*

## 6.3 Training Progression Log

Training was run on the training data set for 12 epochs. Total training time was approximately 12 hours. The background to sheep ratio was 1:2 for the training set and 1:1 for the validation set. Learning rate was set to 0.00075 for epochs 1 to 7, then reduced to 0.0005 for epochs 8 to 12. Figures 14 and 15 shows the training loss and validation $AP_{50}$ that was recorded during training. Loss was recorded every 50 iterations, whilst validation $AP_{50}$ was recorded every epoch.

#### Loss

Training a neural network is about minimising the loss function on the training data set. This means that a model that is learning should have a gradually decreasing loss during training. The loss progression shown in figure 14 does show a general decrease in loss, however training progression is very noisy. This is especially evident when observing the loss recorded per $50^{th}$ epoch in figure 15. Too high learning rate, random augmentation or difference in difficulty within the data set are all possible reasons why the progression is noisy. These reasons are drafted in more detail in chapter 7.3.

#### Validation AP$_{50}$

During training, $AP_{50}$ on the validation data set was measured every epoch. Validation $AP_{50}$ can be seen together with the training loss in figure 14. Keeping track of model performance on a validation data set is useful for preventing overfitting to the training data. Commonly, validation performance will increase at the start of training then begin to stagnate or even decrease. If validation performance begins to decrease while loss continues to improve, this is a sign that the network has begun to overfit to the training data. In figure 14, validation AP reaches it's peak at epoch 9, then it begins to drop slightly. This indicates that the model is finished generalised training at epoch 9 and that these model parameters are the best for a general object detector.

In the Validation $AP_{50}$ curve in figure 14, there is a distinct drop in validation $AP_{50}$ at epoch 6. Similarly to the noisy nature of the loss curve, this is an indication of lacking stability during training and could also be caused by a learning rate that is too high.

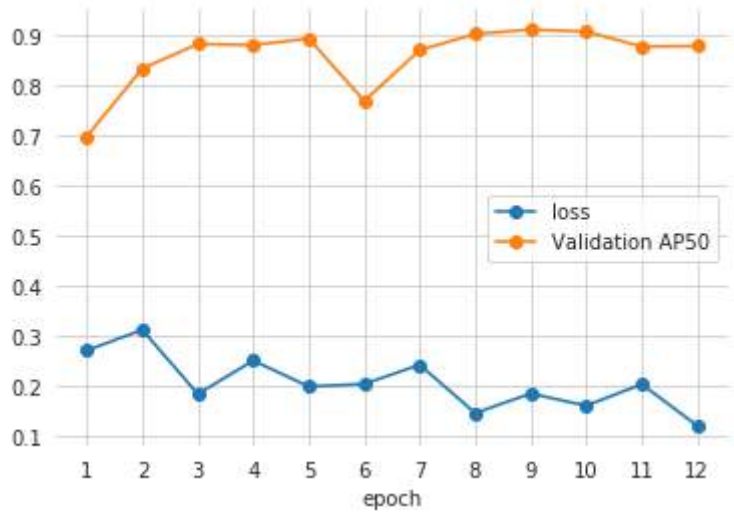


**Figure 14:** *Training loss and Validation $AP_{50}$ per epoch during training.*

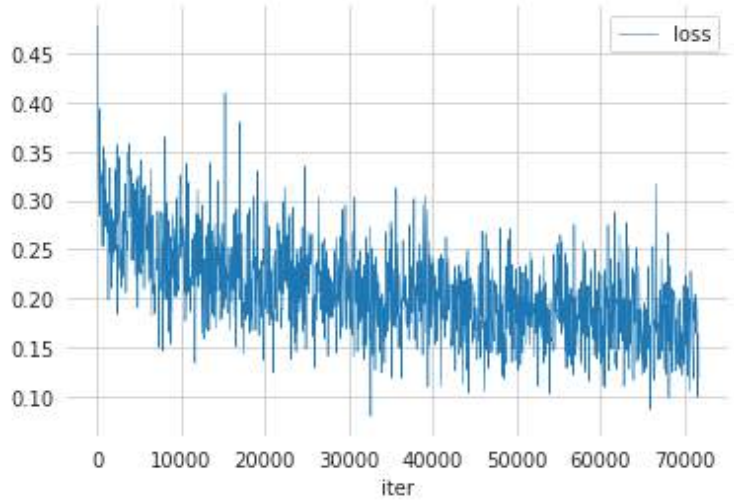


**Figure 15:** *Loss per $50^{th}$ training iteration.*

## 6.4 Detection Results

After training, inference was done on the validation and training images as described in section 5.3.2. Inference for one image took approximately 36 seconds. The model parameters from epoch 9 were used since this had the highest validation $AP_{50}$ recorded during training.

### 6.4.1 Precision $\times$ Recall Curve

The precision $\times$ recall curve in figure 16 shows the precision, recall trade off of the predictions on the validation data set. The curve is created using an IoU threshold of 0.5. $AP_{50}$ is then calculated by taking the average of precision values from 11 evenly spaced recall values between 0 and 1 on the curve. For the validation set, the final $AP_{50}$ was 93.21%.

From the precision $\times$ recall curve, a suitable confidence threshold is chosen that gives an acceptable precision, recall trade off. A confidence threshold of 86% was chosen because this gave a precision result greater than 90% while still producing an acceptable recall. The red line in figure 16 shows where this threshold falls on the precision $\times$ recall curve.

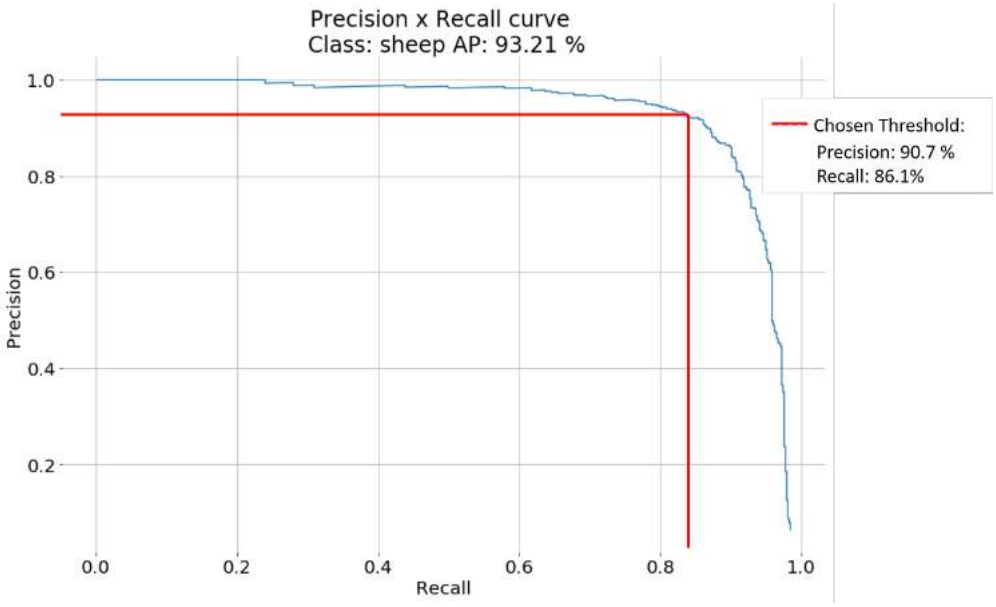


**Figure 16:** *Precision × Recall curve for validation data set. $AP_{50}$: 0.932. Red line shows the precision and recall values obtained at the chosen confidence threshold of 0.86.*

### 6.4.2 Average Precision

Table 9 shows AP results for the training and validation data sets. In addition to the main AP result ($AP_{50}$), results are also reported for stricter IoU thresholds and for image groups with various sheep sizes.

#### $\mathbf{AP}_{50}$

Table 9 show that the object detector achieved $AP_{50}$ results of 91.12 and 93.21 on the training and validation data sets respectively. This is surprising because it is generally expected that a neural network will perform better on data it has been trained on than on unseen data. As a result, there is likely a disparity in difficulty between the two data sets. This is discussed further in chapter 7.4.1.

#### AP by IoU Threshold

As expected, AP is worse at stricter IoU thresholds both for the training and validation data sets. However, the validation data set suffers more than the training data set from the increased strictness of the $AP_{75}$ and $AP_{50:95}$ metrics. This could be due to the detector having learned more specifics of the bounding boxes in the training data sets by being exposed to them during training.

#### AP by Sheep Size

$AP_{50}$ for sheep sizes are also reported in table 9. These are calculated from subsets of the training and validation data sets that are grouped by median sheep diameter. Definition of sheep sizes are given in figure 9. The number of images in each sheep size group is given in table 7. It should be noted that the validation data set only had images containing small and medium sheep but not extra small and large sheep. For the training data set, the detector achieved the much worse $AP_{50}$ result on images with extra small sheep than other sizes. For both the training and validation sets, images with a medium sheep size had the best $AP_{50}$. In general it would seem that the detector performs better the larger the sheep size and by proxy at lower the drone flight height. $AP_{50,L}$ is the exception to this. The slightly lower performance on $AP_{50,L}$ compared to $AP_{50,M}$ can be explained by the much lower prevalence of images with large sheep (see table 7).

| Data set | $AP_{50}$ | $AP_{75}$ | $AP_{50:95}$ | $AP_{50,XS}$ | $AP_{50,S}$ | $AP_{50,M}$ | $AP_{50,L}$ |
|---|---|---|---|---|---|---|---|
| Training | **91.12** | 72.18 | 59.99 | 63.55 | 85.09 | 96.18 | 93.89 |
| Validation | **93.21** | 71.81 | 55.94 | - | 90.53 | 94.43 | - |

**Table 9:** *Average precision (AP) for a range of IoU thresholds and Sheep sizes.*

### 6.4.3 Precision and Recall

As mentioned, a confidence threshold of 86% was chosen to decide which bounding boxes to keep and which to discard. When this confidence threshold has been applied to the predicted bounding boxes, a precision and recall can be calculated. Precision and recall are presented in table 10. The table shows that 86.1% of the sheep in the validation data set were found (recall) and 90.7% of the predictions made were correct predictions (precision). For the training data set, precision and recall was 83.2% and 85.2% respectively. As mentioned in section 6.4.2, the inferior performance on the training data compared to the validation data is unexpected and indicates a disparity in difficulty between the two data sets.

| | | | **Recall(%)** | | | |
|---|---|---|---|---|---|---|
| Data set | **Precision (%)** | **Recall(%)** | White | Grey | Black | Brown |
| Training | 83.2 | 85.2 | 88.8 | 84.6 | 70.6 | 70.5 |
| Validation | 90.7 | 86.1 | 92.5 | 71.1 | 82.0 | 100.0 |

**Table 10:** *Precision and Recall for detections at confidence threshold 0.86 and IoU threshold 0.5.*

#### Recall by Sheep Colour

Table 10 also shows recall values for the different sheep colours. These values should be seen in perspective of the prevalence of each sheep colour in the data set. The number of sheep of the various colours in each data set is reported in table 8. For instance, it should be noted that the validation data set only contains 6 brown sheep so 100% recall for brown sheep in the validation set is not representative of the data set as a whole. Due to a greater sample size, the training set is more representative of the relative recall for the different sheep colours. Results in table 10 indicate that white sheep are the easiest to find and black and brown sheep are the most challenging to find.

# 7 Discussion

This chapter discusses some of the findings presented in the results chapter. In addition, some aspects of the method are discussed.

## 7.1 Data Collection

The results of data collection had some strengths and some weaknesses. It was good that the three field works had such varied weather and environment because this added some useful variation to the data set. Moreover, it was probably useful to have a mix of free ranging and fenced sheep in the data set. The fenced sheep are probably easier to detect because they are not occluded by trees and there are much fewer objects present in the environment that can be confused with being sheep.An advantage of images collected from fenced sheep is that large groups of various different sheep appearances can be photographed at one time, which is positive because the network will be exposed to a range of different sheep. In comparison, free ranging sheep are more difficult to find but they have the advantage of having a more realistic, varied and challenging environment.

In the future, more data of free ranging sheep should be collected in order to expose the network to more realistic situations. With the addition of more varied data, it would also be positive to have a separate test data set without reducing the representativeness of the training and validation data sets. Moreover, the data set should be supplemented with images of other animals that can be found in the same areas as sheep so that the network can learn to distinguish sheep from other animals.

## 7.2 Preprocessing

One of the major decisions in the preprocessing stage was to not have an independent test data set. The reason for this was lack of sufficient independent data. The fact that detection results were better on the validation data set than the training data set could indicate that the difficulty of validation data set was different from the training data set. This could also be interpreted as the validation data set not being representative enough of the data set as a whole, which further justifies the decision of not sparing valuable data for a test data set.

A preprocessing decision that in hindsight should be reconsidered is the inclusion of rotated crops of images in the training data set. The reason for including this to begin with was for augmentation purposes and to have rotated images in the data set without the inevitable black border that appears when images are rotated programatically (as shown in figure 17). However since random rotations up to 45° are applied programmatically anyways then these rotated images become redundant and only aid in increasing the training time.

It should also be considered to limit the data set to images containing small and medium sized sheep. This is because the images with extra small sheep were very challenging for the object detector and when manually observing these images, the sheep do not look like sheep but rather small white specks that are indistinguishable from certain rocks in the data set. Therefore, by excluding these images from the data set, it is possible that this will aid in reducing the number of false positive detections made by the object detector. The images with large sheep should be excluded because they are taken from a very low flight height, which is not realistic for the desired use case.

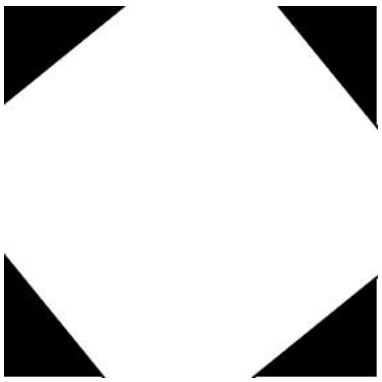

**Figure 17:** *Black border that appears when images are rotated programatically.*

## 7.3 Noisy Training Progression

As shown in the plot of loss during training (figure 15), training progression is quite noisy. Reasons for this could be too high learning rate, random augmentation or difference in difficulty within the data set .

Too high learning rate is a common cause of noisy training progression because it causes updates to the network weights that are too drastic. As a result, the network will always be 'over compensating' for it's bad predictions and have difficulty finding the optimum solution.

As explained in table 5, random gamma, contrast and brightness augmentations were applied to the training images with a probability of 0.2. The purpose of this augmentation is to add variation to the data set and thus making the network invariant to variations in gamma, contrast and brightness that may occur due to varying weather conditions and camera settings. However, random application of these augmentations also has the effect of making difficulty of the data set more variable and can therefore make the training progression noisier.

Another reason for the noisy nature of the training loss progression (figure 15) could be the nature of the data set itself. If there is a great difference in difficulty between images, then loss will vary greatly depending on which images are evaluated.

## 7.4 Detection Difficulty

From looking at the predictions that were made by the object detector, some observations about difficulty can be made. Figure 18 shows some examples of where the object detector has done a good job detecting some difficult sheep in the validation set. This includes some black sheep that are almost impossible to see with the human eye as well as some white sheep that are partially occluded by trees and reflecting a lot of light. The middle image in figure 18 shows a partially occluded sheep, where the boundary of the sheep is not completely clear. This fuzzy nature of some of the bounding boxes in the data set is another reason why $AP_{50}$ is a better metric for the use case in this project than the stricter $AP_{50:90}$.

In contrast, figure 19 shows some examples of mistakes made by the object detector. Sheep that were occluded by trees, reflecting a lot of light, blurred from movement and/or small were some typical examples of sheep that were not detected (FNs). Wrongly classified sheep (FPs) were often tree logs or rocks that reflect a lot of light or melted snow imprints on the ground where a sheep once had laid.

### 7.4.1 Difficulty Disparity Between Training and Validation Set

When comparing $AP_{50}$ performance between the training and validation sets, validation results are better. This unexpected result is likely caused by a disparity in difficulty between the two data sets. When comparing the two data sets, some differences can be seen that may indicate that the training data set is more difficult.


**Figure 18:** *Example of well classified difficult sheep in validation data set. Green boxes show the ground truth, red boxes show the predictions and the red numbers above the boxes shows the predicted box confidence score.*

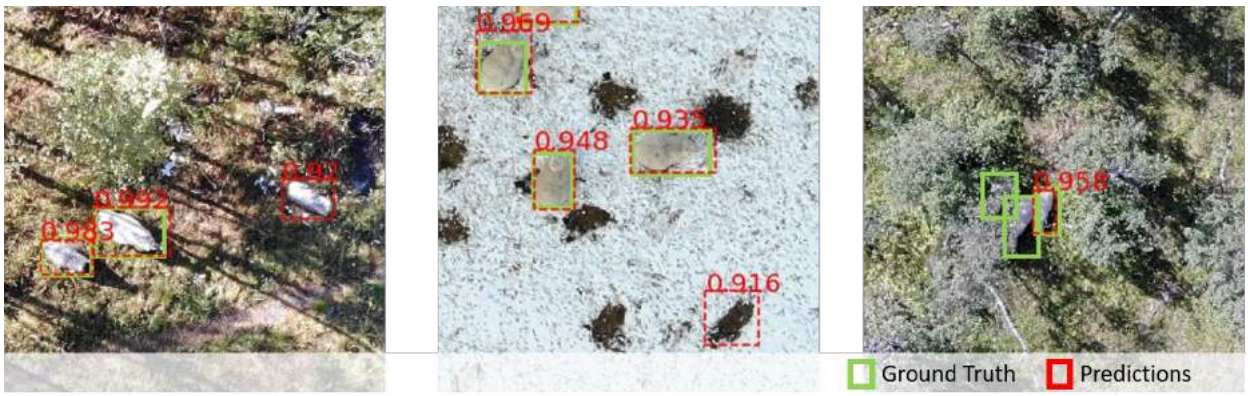


**Figure 19:** *Example of mistakes made on the validation data set by object detector. Green boxes show the ground truth, red boxes show the predictions and the red numbers above the boxes shows the predicted box confidence score.*

Looking at table 7, it can be seen that the training data set has 12% of it's images classified as having extra small sheep, while the validation data set doesn't have any of these images. Images with a small sheep size are taken from a greater flight height, which means the resolution of the sheep is low and as a result the sheep are more difficult to find. The low $AP_{50,xS}$ score obtained substantiates the claim that extra small sheep are extra difficult to detect.

In addition, table 8 reports that the training data set contains 12% black or brown sheep, while the validation data set only contains 4% black or brown sheep. Black and brown sheep appear to be more difficult to detect since they have a lower recall than white or grey sheep in the training data set. The training recall results gives a better indication of relative difference in difficulty between sheep colour than the validation results since it is a larger and more representative data set.

Moreover, from observing the predicted boxes and ground truth bounding boxes on the images in the training data sets, it appears that there are also many environmental factors that make the training data set difficult. Some examples are shown in figure 20. These environmental factors include several rocks that appear similar to the small sheep in the images, grey sheep partially occluded by grey trees or shadows of sheep that themselves look like sheep.

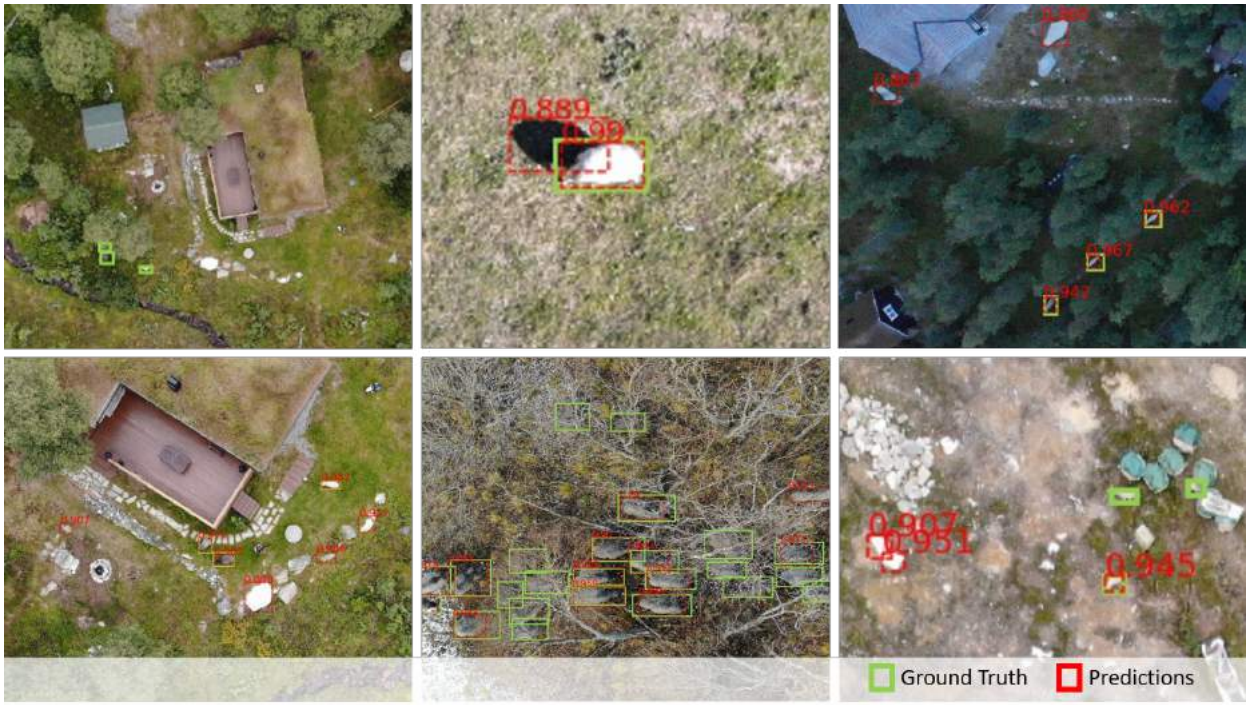


**Figure 20:** *Some examples of difficult cases in the training data set. Green boxes show the ground truth, red boxes show the predictions and the red numbers above the boxes shows the predicted box confidence score.*

## 7.5 Time Considerations

An inference time of 36 seconds per full image is not practical when considering the use case of mapping a large area, which could potentially include several thousand images. As an example, classifying 1000 images at this pace would take 10 hours. Actually, inference on one cropped image takes just 1 second, however since the inference process of a full image involves splitting the image into overlapping crops, making predictions on these then combining all the results together, the total inference time for a full image is quite high. Factors that affect the inference time are hardware capabilities, image resolution and model complexity.

One simple way of reducing inference time is to utilise the hardware better. Currently, the MMdetection

toolbox does not support batch inference so only one crop is processed at a time even though the hardware memory has capacity to process more. Distributing inference to more GPUs would also significantly decrease inference time. Another way to increase inference time without sacrificing resolution or model complexity is by training and testing with better hardware. If the GPU memory capacity is higher, then training and inference can be done on higher resolution images, which means that fewer crops with a lower relative overlap need to be processed.

It is also worth researching the effect of using lower resolution images and less deep models on the overall performance. It is certainly plausible that similar results could be achieved without maximising these factors. Another option is to test out some faster architectures such as YOLOv3.

Finally, the use of rotated crops in the training data set is probably redundant since rotation augmentation can easily be added programmatically. By not including these rotated crops in the training data set, the number of training images is greatly reduced, which would reduce the training time and free up some storage space.

## 7.6 Result in Perspective

The object detector achieved an $AP_{50}$ of 93.21% on the validation data set. When filtering the predictions by the chosen confidence threshold of 86%, the detector achieved a precision and recall of 90.7% and 86.1% respectively. An 86.1% recall may be sufficient for the desired use case since sheep often stay in groups, which means that only one detection per group is sufficient for the farmer to locate the entire group.

However, it should be noted that the precision result is not representative of precision that would be achieved if the detector was applied to an unseen data set covering a large area as depicted in the envisaged solution (figure 2.3). This is because most of the images in a data set such as this would not contain any sheep and as a result there would likely be a greater proportion of false positive detections. In addition, the data set used also includes sheep grazing in confined, fenced areas, which would not be the case when the detector is applied in a real life scenario. Sheep grazing on open fields are likely easier for the object detector to find due to the simple background and lack of other objects that block the view of sheep or can be confused with being sheep.

# 8 Conclusion and Further Research

## 8.1 Conclusion

Automatic detection of sheep using deep learning can be a great complementary aid to existing solutions that help farmers keep track of their sheep. The objective of this project is to examine if deep learning based object detection algorithms can be used to automatically detect sheep in drone images.

This is tested by using the Libra R-CNN object detection architecture on images of free ranging and fenced sheep. The model achieves a 91% precision and and 86% recall on the validation data set, which shows that automatic detection of sheep using deep learning can be a great aid to help farmers find their sheep. Other findings of this project are that the size and colour of the sheep in the images affect how difficult they are to detect. Sheep that have a diagonal bounding box length of less than 50 pixels are much more difficult to find than larger sheep. With regards to colour, white sheep are most successfully detected, whilst a smaller relative portion of black and brown sheep is found by the detector.

## 8.2 Further Research

Although results are promising, the scope of this projects has some limitations. The main limitation is not having enough varied data for a separate test data set. Collecting more varied data of free ranging sheep for training and testing would improve the overall applicability of the results. In addition, the model was only tested on images that contain sheep. A more realistic use case will have an over representation of images that do not contain sheep so a higher rate of false positives would be expected.

There are several matters that can be attempted in order to make further improvements to the object detector. For one, more time could be spent optimising the model parameters such as learning rate, background to sheep ratio of training data, momentum, the number of anchor boxes and anchor box scales.

Inference time is another aspect the should be researched further. Further investigation to determine what processing time is considered reasonable is required. This could for instance be quantified as square meters covered per second ($m^2/s$). With this time considerations in mind, changes to the hardware utilisation, image resolution and network architecture should be explored.

Furthermore, it should be investigated how other object detection architectures perform in comparison to Libra R-CNN. Cascade R-CNN is a model that would be interesting to test since it achieved similar $AP_{50}$ results and better $AP_{50:95}$ results on the COCO data set compared to Libra R-CNN. Single-stage models such as FCOS or YOLOv3 also have potential to perform well on this data set. An argument for why a single-stage model could be useful for this use case is that they are able to reason more globally about the image, which means that they could potentially learn facts such as relative size and that sheep generally stick together in groups. Different object detection architectures have different strength. If multiple different models are trained then results from these could be combined by ensemble learning to obtain better overall prediction results.

Finally, the possible contribution of the IR images should be investigated since the IR images contain a new dimension of features that have potential to improve detection results. An issue with this is that the IR and visual images do not align pixel for pixel. Alignment of visual and IR images is required in order to relate the coordinates in the IR camera coordinate system to their respective coordinates in the visual camera coordinate system. Solving this alignment problem and obtaining a pixel to pixel overlap would makes it possible to use the information from both the visual and IR channels in order to train a neural network for object detection

# Bibliography


[1] og Gjeit (NSG), N. S. 2019. Sauerasene i norge. URL: http://www.nsg.no/saueraser-i-norge/category719.html.

[2] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., & Lin, D. 2019. Mmdetection: Open mmlab detection toolbox and benchmark. arXiv:1906.07155.

[3] natural Resources Canada. 2019. Concepts of aerial photography. URL: https://www.nrcan.gc.ca/earth-sciences/geomatics/satellite-imagery-and-air-photos/national-air-photo-library/about-aerial-photography/concepts-aerial-photography/9687.

[4] Nortura. 2019. Nortura hjemmeside. URL: http://www.nortura.no/.

[5] norske leksikon, S. 2019. Sauehold i norge. URL: https://snl.no/sau.

[6] Nortura. 2019. Temahefte - utmarksbeite til sau. URL: https://www.geno.no/globalassets/geno-sa/02_dokumenter/02_aktiviteter/06_husdyrtreff/oppgave-sau/2020/utmarksbeite_sau_web.pdf.

[7] Hvasshovd, S.-O. 2019. Droner og sau og litt til !! anvendelser og muligheter. URL: https://www.fylkesmannen.no/contentassets/cbf122460efa4e37a051c17c07fade0d/droner-buskerud-2017.pdf.

[8] Smartbjella. 2019. Smartbjella. URL: https://smartbjella.no/.

[9] findmy. 2019. findmy. URL: https://www.findmy.no/.

[10] Telespor. 2019. Elektronisk overvÅking av husdyr. URL: https://telespor.no/.

[11] Anne-Cath. Grimstad, N. S. o. G. 2017. Tilsyn med drone: Rimelig og effektivt. *Sau og Geit*, Sau og Geit Nr.4/2017. doi:http://dx.doi.org/10.1002/andp.19053221004.

[12] DJI. 2019. Dji mavic 2. URL: https://www.dji.com/no/mavic-2.

[13] MathWorks. 2019. What is object detection? URL: https://se.mathworks.com/discovery/object-detection.html.

[14] Zou, Z., Shi, Z., Guo, Y., & Ye, J. 2019. Object detection in 20 years: A survey. arXiv:1905.05055.

[15] Pancast. 2019. Object detection: What is it and how is it useful? URL: https://www.panacast.com/object-detection-what-is-it-and-how-is-it-useful/.

[16] Girshick, R., Donahue, J., Darrell, T., & Malik, J. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524.

[17] Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J. M., & Zisserman, A. 2009. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 303–338.

[18] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. 2014. Microsoft coco: Common objects in context. arXiv:1405.0312.

[19] Wu, X., Sahoo, D., & Hoi, S. C. H. 2019. Recent advances in deep learning for object detection. arXiv:1908.03673.

[20] pascal VOC. 2012. Visual object classes challenge 2012 (voc2012). URL: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/.

[21] Girshick, R. 2015. Fast r-cnn. arXiv:1504.08083.

[22] Ren, S., He, K., Girshick, R., & Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv:1506.01497.

[23] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. 2016. Feature pyramid networks for object detection. arXiv:1612.03144.

[24] Cai, Z. & Vasconcelos, N. 2017. Cascade r-cnn: Delving into high quality object detection. arXiv:1712.00726.

[25] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., & Lin, D. 2019. Libra r-cnn: Towards balanced learning for object detection. arXiv:1904.02701.

[26] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2015. You only look once: Unified, real-time object detection. arXiv:1506.02640.

[27] Redmon, J. & Farhadi, A. 2016. Yolo9000: Better, faster, stronger. arXiv:1612.08242.

[28] Redmon, J. & Farhadi, A. 2018. Yolov3: An incremental improvement. arXiv:1804.02767.

[29] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. 2017. Focal loss for dense object detection. arXiv:1708.02002.

[30] Tian, Z., Shen, C., Chen, H., & He, T. 2019. Fcos: Fully convolutional one-stage object detection. arXiv:1904.01355.

[31] Laboratory, M. 2019. Multimedia laboratory. URL: http://mmlab.ie.cuhk.edu.hk/.

[32] PyTorch. 2019. From research to production. URL: https://pytorch.org/.

[33] Labelbox. 2019. Labelbox homepage. URL: https://labelbox.com/.

# A Appendix

## A.1 Validation Image Prediction Examples

Some examples of results from predictions on validation images. Green boxes are ground truth values, red boxes are predictions and red numbers are confidence values.