Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Marius Steller Imingen
John Eric Woodcock

Automatic detection of sheep in Norwegian highland terrain using YOLOv3

Master's thesis in Informatics
Supervisor: Svein-Olaf Hvasshovd
June 2020

NTNU
Norwegian University of Science and Technology

Marius Steller Imingen
John Eric Woodcock

Automatic detection of sheep in Norwegian highland terrain using YOLOv3

Master's thesis in Informatics
Supervisor: Svein-Olaf Hvasshovd
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Summary

The usage of unmanned aerial vehicle's (UAV) has seen an increase as more people have been testing their potential. Usage areas such as wildlife monitoring, search and rescue, inspection and traffic monitoring are some of the fields that benefit from the development.

Combining UAV's with state of the art object detectors have shown great promise in automating the jobs stated above. This can help increase productivity, decrease human error and automate dangerous jobs. This thesis aims at exploring how a UAV equipped with a high resolution camera as well as an infrared camera, combined with a state of the art object detector can help locate sheep in Norwegian highland terrain. One of the state of the art object detectors is YOLO (You Only Look Once), which gives a good balance between precision and inference time. YOLOv3 is the implementation used in this thesis.

For the shepherd, finding the rough location of the sheep is the most important aspect. The exact location of every individual sheep in an image is not essential, as the sheep can move around from the time of detection to herding. Thus, filtering out images without any sheep is the most important. Another aspect is to compare the performance of RGB pictures and infrared, to see if the infrared camera is redundant. 4k RGB pictures requires more computing power, memory usage, as well as having three distinct wool colours. As opposed to infrared that only output greyscale.

It is unclear how much of a benefit infrared images yield at this point, as the results were very close between the networks. Infrared images increases performance in cold environments, but struggles with generating clear images at higher altitudes, while RGB images have a more consistent performance. Different network configurations were used on RGB and infrared images, making it hard to compare the results fairly.

Sammendrag

Bruken av ubemannede luftfartøy (UAV, eller drone på norsk) har sett en økning i bruk etter at flere folk har testet potensielle bruksområder. Bruksområder som overvåkning av dyreliv, søk og redning, inspeksjon og overvåkning av trafikk er noen av feltene som drar nytte av denne utviklingen.

Ved å kombinere droner med topp moderne objekt detektorer har det blitt vist stort potensiale i å automatisere ovennevnte jobber. Dette kan hjelpe med å øke produktivitet, minke menneskelige feil og automatisere farlige jobber. Denne oppgaven utforsker hvordan en drone utstyrt med høyoppløselig kamera i tillegg til et infrarødt kamera, kan kombineres med en topp moderne objekt detektor for å lokalisere sauer i norsk fjellområder. En av de topp moderne objektdektorene, er YOLO (You Only Look Once), som gir en veldig god balanse mellom presisjon og inferensstid. YOLOv3 er implementasjon som ble brukt i denne oppgaven.

For sauebonden er det viktigere å finne en tilnærmet posisjon for sauene. En eksakt lokasjon av hver eneste sau i et bilde er ikke det viktigste, siden sauen kan ha beveget seg i tiden mellom deteksjon og manuell gjenfinning. Det betyr at det å filtrere ut alle bilder uten sau er viktigere. Et annet aspekt av oppgaven er å sammenligne fargebilder og infrarøde bilder, for å se om et infrarødt kamera er overflødig. 4k fargebilder krever mer komputasjon, minne og fargebildene gir tre forskjellige farger på sauens ull. I motsetning har infrarøde bilder bare gråskala.

Det er uklart i hvor stor grad infrarøde bilder er bedre på dette stadiet, siden resultatene er veldig jevne. Infrarøde bilder gir bedre resultat i kalde omgivelser, men sliter med mer uklare bilder i større høyder, mens fargebilder har en mer konsistente resultater. Forskjellige nettverkskonfigurasjoner ble brukt på fargebilder og infrarøde, så det var vanskelig å sammenligne resultatene rettferdig.

Preface

This master thesis was written for the Master of Science in Informatics programme at Norwegian Univerity of Science and Technology, Trondheim 2019/2020.

We would like to thank our supervisor Svein-Olaf Hvasshovd for input and guidance during the thesis. The feedback has been valuable and very helpful. We would also like to thank Kari Meling Johannessen and Magnus Guttormsen for operating the drone to collect data. As well as Magnus Falkenberg Nordvik, Jens Tobias Kaarud and Håkon Rosseland Paulsen for cooperation on labeling the large number of images.

Table of Contents

List of Tables

List of Figures

Abbreviations

UAV	=	Unmanned Aerial Vehicle
ANN	=	Artificial neural network
CNN	=	Convolutional neural network
FOV	=	Field of view
RGB image	=	Red Green Blue image
YOLO	=	You Only Look Once
TP	=	True Positive
FP	=	False Positive
IOU	=	Intersection Over Union
SVM	=	Support Vector Machine
R-CNN	=	Region-Convolutional Neural Network
GPGPU	=	General-purpose computing on graphics processing units

Chapter 1

Introduction

Sheep is among the most popular farm animals in Norway (SSB, 2019) with about 2.1 million sheep out on pasture every year (Blix and Vangen, 2019). The sheep pasture in large areas, varying from closed in green fields to open areas like forests and rocky highlands[1]. One of the main challenges in sheep farming is the roundup at the end of the seasons, which is resource and time consuming. The sheep can pasture in flocks of varying size, from big flocks to smaller groups. Even if sheep tend to stick together, the individual groups can be spread out in a large area. Usually the farmer has to go over their area multiple times in order to gather all the sheep, where the last rounds of gathering are the most challenging.

As of now there are two approaches that help the farmer locate the sheep. One approach is to radio tag the sheep, which will decrease the time spent looking for the sheep, but it is an economic investment for the farmer. One of the problems with this approach is that some of the areas the sheep are grazing in do not have radio signal coverage, therefore making the economic investment a bigger risk for the farmer. The second approach is to use satellite in order to track the sheep. This comes at a very large economic investment and therefore might not be as valuable as simply looking for the sheep manually in areas where the radio transmitters doesn't work. Both approaches are also dependent on GPS to function.

This thesis proposes using unmanned aerial vehicles (UAV's) in order to solve the problem of gathering all the sheep. An off the shelf consumer UAV, equipped with a high resolution camera can cover a large area and take images or video of the area. The farmer can control the UAV himself and cover a specific area, or the UAV can fly in a predefined flight pattern and cover an area itself. As well as being a cheaper investment for the farmer. The main problem with this approach is the manual filtering of the images. The farmer has to manually filter through all the images taken and personally determine if an image

[1] A lot of sources on sheep and sheep farming in general is from the thesis supervisor Svein-Olaf Hvasshovd, who has a lot of experience with sheep farming.

contains a sheep, then figure out what area this image was taken in. This approach will generate a massive amount of images, taken from a large area. It is not unlikely that the sheep will have moved in the time it takes to filter the images to when the farmer is out looking for the sheep.

The developments in artificial intelligence and specifically computer vision and object detection shows great promise as a solution to the previously stated problem. State of the art object detectors like You Only Look Once (YOLO) (Redmon et al., 2015), faster R-CNN (Ren et al., 2015) and Single Shot Multibox Detector (SSD) (Liu et al., 2016) and others are all claiming state of the art results in the field of object detection and localisation. As of now, YOLO and specifically YOLOV3 is showing the better results with a more preferable balance between precision and inference time. It is able to compete with all the other detectors on precision, but has better inference time. This thesis will look at how well YOLOV3 performs in detecting sheep in Norwegian highland terrain from a UAV point of view, on both regular 4K RGB images and lower resolution infrared images.

Chapter 2

Literature Review

This chapter gives an overview of research related to our thesis problem. Related research includes automatic object detection on UAV imagery, as well as a look at state of the art object detectors.

2.1 Earlier master's thesis

Detecting sheep using UAV imagery has been a thesis topic in a few earlier years as well. In 2019, Jonas Hermansen Muribø achieved good results by using YOLOv3 (Muribø, 2019) on UAV images of sheep. With recall and precision at 0.99 and 0.94 respectively, the thesis shows that YOLOv3 is a good candidate for the thesis problem. The thesis does however point out a few threats to the validity of the experiment. One major point of concern is the data used. The images used in this thesis have only grassy fields as background, which in turn provides a very good contrast to the sheep and might be one of the reasons why the thesis achieves such good results. The grassy fields do not represent the most relevant environment for Norwegian sheep pasture, and it is unknown how well it would perform on sheep in a more diverse environment. The thesis states that it is possible that the model has learned to detect sheep as spots of white, brown and black surrounded by green. In a more realistic environment it will be harder to spot the sheep from the background, especially the brown and black that blend very well with Norwegian highland. This previous work does however indicate that using deep learning on high resolution RGB UAV images warrants further testing and research, and might be a very relevant solution.

2.2 Wildlife monitoring

Modern development in UAV technology has made great progress in the field of wildlife monitoring (Chabot and Bird, 2015), but a lot of untested potential still exists. According to Chabot and Bird, UAV's are particularly well suited for collecting data and the UAV's

prove both useful to humans as well as being unobtrusive to to the animals, especially for aggressive or sensitive creatures. UAV's also make gathering data easier in areas that are usually hard to navigate or places that are generally hard to reach, like birds nests. However, the paper argues that various barriers that hinder effective use remains, and that the only way to normalize UAV usage is further research.

Monitoring wildlife for preservation and research purposes is a job often done manually, and is in that case very slow, resource heavy and labour intensive (Gonzalez et al., 2016). The use of inexpensive, consumer friendly UAV's and automatic object detection might provide a solution to the manual labor (van Gemert et al., 2014). Van Gemert et al. tests out a fully automatic detection process, meaning that image capturing and object detection are done on board the drone. The paper then argues that deep learning models are too resource intensive for the on board electronics, even though these models are considered state of the art. The experiment does however show good results by using more light-weight machine learning models to run on board analysis. Three algorithms were tested, with exemplar SVM showing the best result with a precision of 0.66 and recall of 0.72.

Another approach from (Andrew et al., 2017) demonstrates the usage of a version of faster R-CNN (Ren et al., 2015) to detect cattle, and additionally testing individual cattle identification. The paper achieved an accuracy of 99.3% on object detection and 86.1% on identification. While the results are great, the paper does not provide precise precision and recall measures. Additionally the images in the dataset only contain 89 individual cattle, and the images are from the same two hour period. The results from the experiment might be biased toward the data and the model might not be able to generalize well enough.

2.3 Other usage areas

In addition to wildlife monitoring, the use of UAV's and automatic detections has many other usage areas. One area that may benefit is civil engineering, such as fire detection, search and rescue, bridge inspections, power line inspections and traffic monitoring. In one paper by Radovic et al (Radovic et al., 2017), they managed to achieve pretty good results on real time tracking and detection on video using the YOLO (Redmon et al., 2015) algorithm. In this paper they first trained the system on detecting airplanes and managed to achieve an accuracy of 97.5 %. Additionally they tested their system on a multiobject scenario, where they tried to detect more than one type of object in the same image using real-time video. They managed to achieve an accuracy of 84 %. The video feed was taken from a horizontal point of view though, and does not present the general point of view a UAV usually takes images in. Even so, the paper shows that YOLO is able to detect multiple objects in real time video with low inference time.

In another paper from 2017, Wang and Zhang (Wang and Zhang, 2017) propose using UAV's and machine learning in order to automatically detect cracks in wind turbines. They argue that a system like this will enable more frequent inspections and eliminate the danger with human inspection. Additionally, it will help decrease human error and the subjective impression of the human inspector. This is a common theme in UAV

assisted automatic detection where the objective result of the computer system together with a human inspector might help lessen error. In this paper, Wang and Zhang compare two Logitboost cascading classifiers on Haar-like features. The first classifier is a novel Logitboost cascading classifier, while the other is an extended cascading classifier. The extended classifier is a Logitboost same as the first one, but substitutes the Logitboost algorithm with Decision Tree or SVM respectively, if the number of predefined features are reached. This is done in order to combat overfitting at later stages and in order to seperate negative and positive samples at later stages. The extended classifier shows the best result with an accuracy of 97%.

2.4 State of the Art Object Detection

As the previous sections show, a variety of algorithms and approaches for the object detector exists. Even though traditional computer vision techniques are not as prevalent, they still hold up with modern approaches like novel machine learning algorithms or deep learning (O'Mahony et al., 2019). A few reasons a novel computer vision approach might still hold up, is the vast data and computational resources deep learning models need in order to achieve great accuracy, as well as being more dependent on high resolution data and time in order to train a model.

In the field of deep learning, many models can make the claim to be "state of the art". Different implementations yield different result, where the comparable is usually on metrics measuring accuracy, and inference time. As of now, the biggest gain some of the models have on other models seems to be on inference time, as the measure in accuracy is very close between the models, which can be viewed in figure 2.1. As the figures show, YOLOv3 has significantly better inference time, while still keeping up with other models in accuracy.

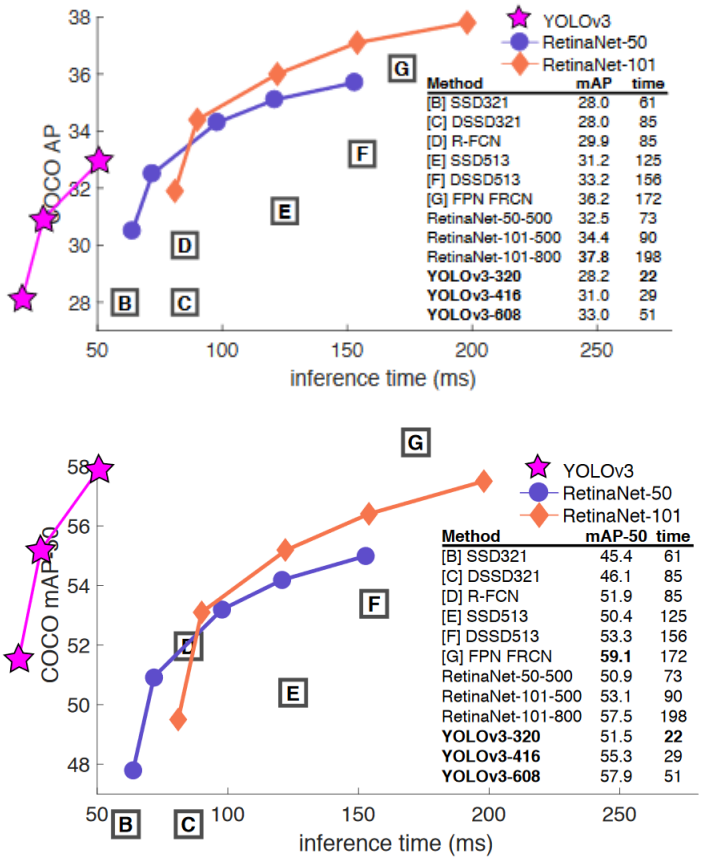


Figure 2.1: Two comparisons between state of the art models (Redmon and Farhadi, 2018), showing that the models generally achieve very similar accuracy (mAP), but there is a lot to gain in inference time.

Chapter 3

Basic Theory

This chapter will first explain the nature of neural networks, then explain convolutional neural networks as well as an explanation of YOLOv3, and lastly how the area covered by an image is calculated.

3.1 Neural Networks

3.1.1 Artificial neural networks

An artificial neural network (ANN) is a collection of nodes with some form of structured connection between each node. A node will ”fire” an output which value depends on its inputs the input connections weight and lastly the activation function (See Figure 3.1 for an overview of an artificial neuron). The structure of the nodes connections and the type of activation function used will describe the properties of the network. There are several neural network typologies of which the Feed-forward network is relevant (See Figure 3.3). In a Feed-forward network there is a direct ”flow” from the input of the network to the output, each layer will only send output to layers after it and will never loop back to previous neurons (Russell and Norvig, 2009, p.727-728).

3.1.2 Convolutional neural networks

A Convolutional neural network (CNN) is similar to the previously mentioned ANN, but the main differences are that CNN‘s assume the input will be an image in the form of a 3D matrix. In the CIFAR-10 dataset the images are 32 pixels wide, 32 pixels high and 3 pixels in depth (the matrix has a depth of 3 because its a RGB image). In a normal ANN each pixel would need to be connected to a neuron in the input layer making the input a size of 32*32*3 = 3072 which is manageable, however using larger images will quickly make a fully connected network significantly slower. Therefore CNN‘s use small 2d matrices as weights, which are commonly called filters (or kernels). The filters are than applied to a area on the input and perform a convolution with the filter and the applied pixels in the

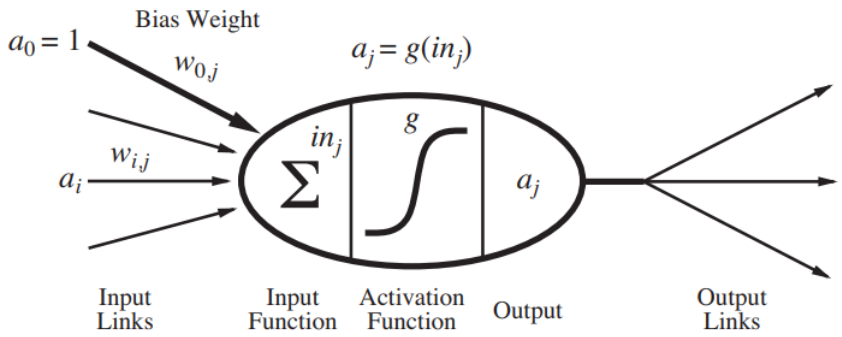


Figure 3.1: An artificial neuron where a_i defines the neurons input while w_{ij} is the neurons weights (Russell and Norvig, 2009).

input, extracting the more important details of the image. After the convolution, a pooling layer will be used to reduce the size of the input, which is how the network handles larger resolution images (See Figure 3.3 for how the width and height decrease). A CNN consists of three main types of layers a convolutional layer, a pooling layer and a fully connected layer (Stanford, 2020a).

Convolutional layer

The convolutional layer will perform a convolution between a filter and a part of the input resulting in a single value. The filter is then moved to the right by a given amount of pixels which results in a second value. The amount of pixels the filter moves is defined by the layers stride, which is chosen when creating the network. The filter will continue to slide until the filter has covered the entire image, after which the next filter is used. The resulting output will be a 3d matrix with the width and height reduced by $inputSize - (filterSize - 1)$, while the depth will depend on the input depth and the amount of filters (Stanford, 2020a).

Pooling layer

The pooling layer is used to reduce the size of the features width and height, thereby reducing the necessary computation needed to run the network. Larger networks are also more prone to overfitting which the pooling layer also helps control. The pooling layer selects a small 2 matrix of pixels similar to the convolutional layers. The selected pixels however are not used to convolute, but to reduce the selected area down to one value, the pooling layer can use different methods to chose which value is output, but typically the max value in the filter is selected. The layer typically only uses a 2x2 matrix with a stride of 2, resulting in about 75% of the width and height of the layers input being removed (See Figure 3.2).

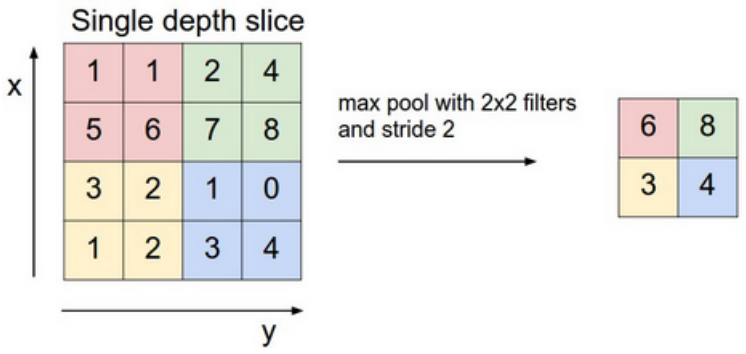


Figure 3.2: Figure showing how max pooling works (Stanford, 2020a)

Fully connected layer

Fully connected layers are used after the convolutional layers have extracted the features of the image. Eventually the spatial size of the features should be small enough to connect to the fully connected layers. Fully connected layers function just like a layer in an ANN as explained above where each neuron in a ANN connects to all activation's in the previous layer. The activated neuron in the last fully connected layer will then serve as the classifier for the original input image (Stanford, 2020a).

3.1.3 Transfer learning

Training a convolutional network from scratch requires a large amount of data and will take considerably longer to train, therefore several methods to reduce the time needed have been created. The general method is to download the weights for a network which has already been trained and use it either partially or completely, potentially also adjusting the downloaded weights. The first approach is to load the convolutional section into a new network and train the remaining fully connected layer. The second approach will similarly load a finished network into the convolutional layers, but the difference is that during training the network will also adjust the convolutional layers which were loaded and not just the fully connected layers. The last approach is to load the entire network and fine-tune the downloaded weights through training (Stanford, 2020b).

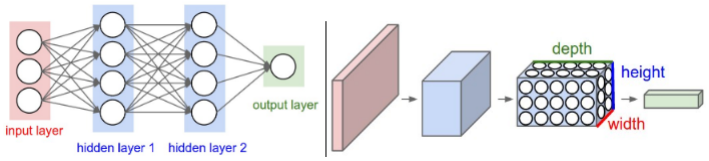


Figure 3.3: Left shows the architecture of a ANN, while right shows the architecture of a convolution neural network. Figure was taken from (Stanford, 2020a).

3.2 YOLOv3

You Only Look Once (YOLO) is a real time object detection algorithm which uses a convolutional neural network type architecture to predict both an objects class and bounding box. YOLOv3‘s architecture consists of a total of 106 layers the first 53 of which are already trained on the Imagenet dataset. The first half of the network is structured as seen in Figure 3.4 with several residual blocks allowing shortcuts through the network. The pre-trained network is simply used to extract features from the input image and pass them on to the other half of the network. The remaining 53 layers then need to be trained to the specific dataset they are going to be used on.

The second half of the network is where the prediction of bounding boxes, objectness and class happens. Bounding boxes are coordinates of where the object is located, objectness is a single value which predicts whether or not there is an object in the bounding box and class is a prediction of which class the object belongs to. The predictions are done three times in the default YOLOv3 network, each at a different scale. At layer 82 the first predictions are made, but the input to the prediction block is also passed to an upsampling block which upsamples the feature maps. After upsampling, convolutional layers are used to merge the newly upsampled feature maps with feature maps from earlier in the network. The merged maps are passed to the second prediction block to make a new prediction, but with the maps now being twice the size. The same process is repeated for all three YOLO layers in the network(Redmon and Farhadi, 2018).

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3.4: YOLOv3 feature extractor network (Redmon and Farhadi, 2018)

3.3 Field of View

As mentioned in earlier chapters the goal of this thesis is to make it easier for a sheep herder to locate his sheep, therefore finding the rough location of a flock is more important than finding every instance of a sheep. In addition there is not a defined distance required for a group of sheep to constitute a flock, thus an assumption of approximately 200 meters was made. To find out how much of an image can be considered a flock, the length and height of the area in the image needs to be calculated. This is done by using trigonometry on the height of the UAV and the Field of View (FOV) of the camera. According to the UAV's manual (DJI, 2019) the RGB camera has a diagonal FOV of 85 degrees. The images were taken in heights ranging from 14m to 120m.

Given an RGB image taken at 120 meters, the length of its diagonal will be:

$$d = 2 * 120 * Tan(85/2) = 219.919m \tag{3.1}$$

Since the aspect ratio of the RGB images are 4:3 the horizontal and vertical lengths can be calculated as follows:

$$h = 219.919 * Cos(Arctan(3/4)) = 175.9356m \tag{3.2}$$

$$v = 219.919 * Sin(Arctan(3/4)) = 131.9517m \tag{3.3}$$

Where d is the diagonal length of the area covered by the image, h is the horizontal and v is the vertical. The UAV's RGB images therefore have a maximum FoV of 175.937m by 131.952m

For the infrared camera the UAV documentation only mentions the horizontal FoV, however the infrared sensor documentation shows the diagonal FoV (FLIR Systems Inc). With the diagonal FoV the same equation can be used to find the length of the images field of view.

$$\begin{aligned} d &= 2 * 120 * Tan(71/2) = 171.190m \\ h &= 171.1903 * Cos(Arctan(3/4)) = 136.952m \\ v &= 171.1903 * Sin(Arctan(3/4)) = 102.714m \end{aligned} \tag{3.4}$$

The FoV for the infrared camera is therefore 136.952m by 102.714m. While the total area a RGB or infrared image can cover is significant, it is close enough so that any sheep found in the same image will be defined as a part of the same sheep flock [1].

3.4 Measuring performance

Measuring the performance of the networks are done by looking at different metrics the networks produce when they are done training, and on different test sets.

[1] This area was sufficient enough for the farmer to locate the sheep manually after, as Svein-Olaf Hvasshovd concluded.

Precision

Precision is defined as the number of true positives over the number of true positives plus the number of false positives (scikit-learn developers , 2020).

$$\frac{TruePositives}{TruePositives + FalsePositives} \tag{3.5}$$

Essentially, this means the number of detections that were correct detections. If a dataset contains 50 sheep, and a network detects 25 sheep, but 20 of them were actually sheep, then the precision is 0.8

Recall

Recall is defined as the number of true positives over the number of true positives plus the number of false negatives (scikit-learn developers , 2020).

$$\frac{TruePositives}{TruePositives + FalseNegatives} \tag{3.6}$$

This means the fraction of the total number of objects in the dataset that were actually detected. False negatives are the number of objects that were **not** detected. Using the same example from the previous section, the recall rate is 0.4. Having high precision, but low recall means that the network detect few sheep, but the detections have a higher chance of being correct. The opposite means that the networks detect a lot of sheep, but also have a lot of false positives reducing the precision.

mAP@50

mAP@50 is a metric combining mean average precision and a intersection over union (IOU) threshold, where the threshold is .50. The threshold means that the IOU has to be 0.5 or higher in order to be considered a true positive.

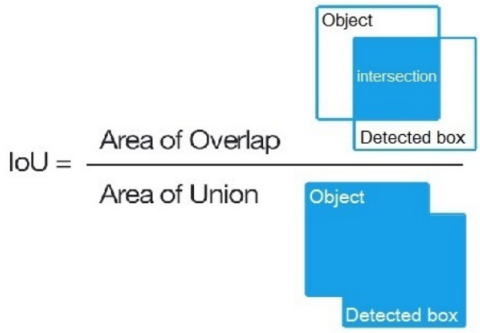


Figure 3.5: Shows how IoU is calculated (AlexeyAB, a).

Chapter 4

Planning & Structure

4.1 Planning

4.1.1 Data Acquisition and Analysis

A good dataset for training should ideally be large and contain a diverse set of images. The better the dataset is, the better the model is able to generalize. Defining diverseness in this experiment means having images taken from different heights, from different areas, with different backgrounds and weather conditions. A diverse set of wool color is also ideal. One frustration in dataset generation is that white sheep are more prevalent than black and brown, but black and brown wool color is harder to differentiate from the background. This information combined with the results from previous master thesis (Muribø, 2019), indicates that detecting sheep independent of their wool color should yield better results. Thus we trained our models at detecting sheep in general, and not white, brown and black sheep distinctly. In addition, distinguishing between sheep based on their wool color is close to impossible when using the infrared images.

All of the images in the dataset were taken between 21-22 August, 20-22 September and 25th of October 2019 at Storlidalen in Trøndelag. Data from a test flight nearby Dragvoll without images of sheep were added as negative samples as well. A rough location can be seen in figure 4.1. The data was collected by the supervisor Svein-Olaf Hvasshovd and two other master students at NTNU, Kari Meling Johannessen and Magnus Guttormesen.

The dataset contains images from different heights ranging from 14 to 120 metres. As explained in section 3.3 the maximum FoV any image can have is 176m by 132m on the RGB images. This means that the sheep is at maximum $176m/2$ (considering the UAV is always in the middle of an image) from the GPS location of the UAV. This in turn means that the maximum area the shepherd has to manually look for the sheep is an area

Figure 4.1: Approximate locations of the datasets images. Left is an image of Storlidalen in Oppdal (Google-Maps, 2020a) and Right is an image nearby Dragvoll in Trondheim (Google-Maps, 2020b)

of 176m by 132m. Assuming the sheep has not relocated. Different backgrounds were also prevalent in the dataset, with some grassy fields, forest environment, as well as rocky highland. Weather conditions also vary somewhat with mostly cloudy or sunny conditions, as well as some colder weather with snowy background (See Figure 4.2).

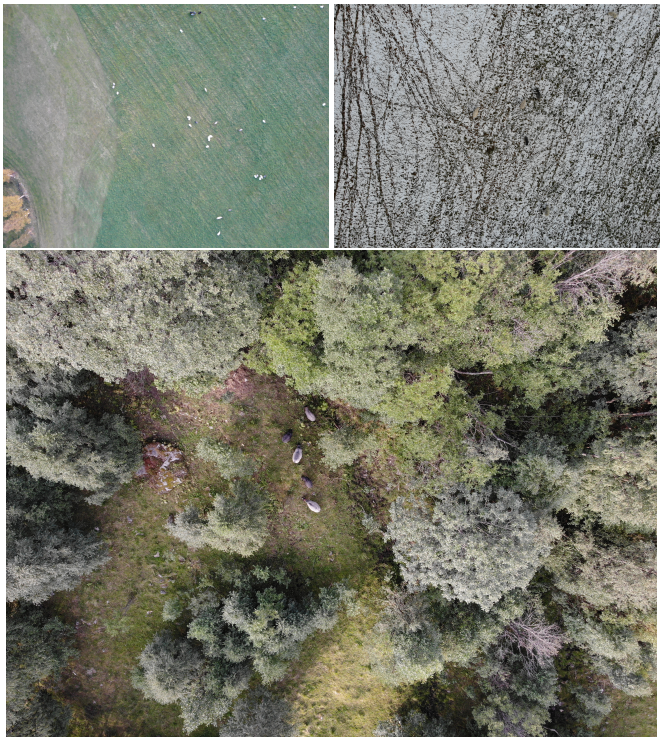


Figure 4.2: Some of the different backgrounds and altitudes of the RGB images.

Datasets for the experiment were generated manually in order to create the most optimal

training data for the model. Most images were suitable for usage, with only a few being unusable. Unusable images meaning blurry images, unclear infrared information or images that do not follow the same settings as the other images in the dataset. A lot of images were very similar to each other as well, almost looking like duplicates. The reason being that images were taken in bursts in order to gather a lot of data. This also required manual filtering of the raw dataset, in order to create a diverse dataset with as many different backgrounds, environments and altitudes as possible. The dataset should not contain too many images that look very similar as this could cause the networks to overfit. Images taken of the same herd at the same place, but at different altitudes were still defined as different images. It was only in the instance of similar images at the same altitudes that some of the images were filtered out.

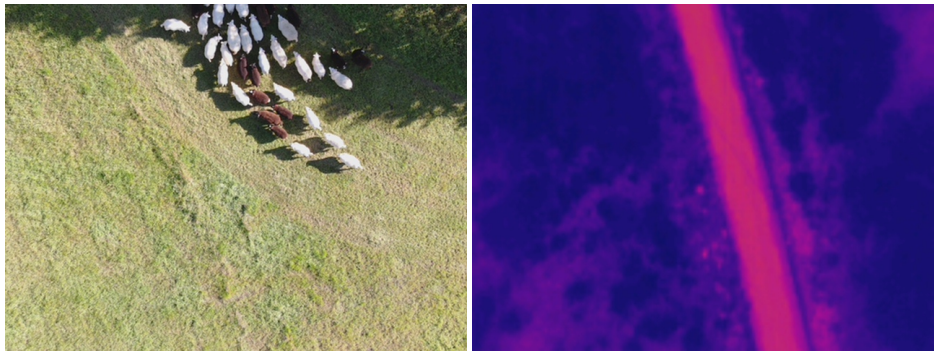


Figure 4.3: Examples of unusable infrared images, The left image was in the location of an infrared image and has a resolution of 640x480 making it unusable for both the infrared and RGB network. Image on the right used a different color palette than gray scale.

Labeling

After cropping the 4k images the number of images had effectively multiplied by 20, causing the amount of work needed to label them to multiply by 20 as well. Therefore this projects supervisor recommended cooperation on sharing data. Together with another group (Magnus Falkenberg Nordvik, 2020), the RGB images were uploaded to Labelbox (Labelbox, 2020) and used the websites tools to label the images. The infrared images on the other hand were not labelled cooperatively and were instead labelled using labelimg (Tzutalin). This was because of the significantly lower number of images causing labeling to be less work.

4.1.2 Experiment Structure

The experiment used the darknet implementation by Alexey (AlexeyAB, a), which is a popular fork of the original darknet repository by the creator of YOLO (pjreddie). The repository by AlexeyAB contains many improvements over the original code. Some of the improvements include general performance, more optimal GPU usage, windows support, runtime warnings and improved metric calculations. In addition to code improvements,

AlexeyAB provides a detailed plan for improving detection on custom datasets. AlexeyAB's implementation is more optimal for training on GPU's, which was a concern for this experiment as the training was done on the NTNU IDUN computing cluster (Själander et al., 2019). The cluster has more than 70 nodes and 90 GPGPU nodes. Half of the nodes are equipped with two or more Nvidia Tesla P100 or V100 GPGPU's, which this experiment will take advantage of.

In order to test YOLO's performance on both infrared and RGB images, two different models were needed. Training and testing on both models were carried out separately. The difference in size between RGB and infrared images, as well as the difference in the images, warrants different settings and thus separate testing was preferred. For instance, the RGB images contain sheep in three different colors, but for the infrared images all sheep are shown as white dots indicating heat.

With this in mind, these research questions were formed:

RQ1: How well does YOLOv3 perform in detecting sheep in highland terrain?

RQ2: Do infrared images improve the detection of sheep as opposed to RGB images?

The performance of a network was determined by comparing the performance data the different network configurations generate. Many different metrics were generated when the networks were tested, but the most important metrics was:

- **Precision:** The accuracy of predictions, the percentage of predictions that were correct predictions.
- **Recall:** The percentage of predictions to the number of objects in the dataset.
- **mAP@50:** Mean average precision with a threshold of 0.50 intersection over union.

4.2 Preprocessing

4.2.1 Preprocessing for YOLO

The network required some setup before the experiment and training could start. The YOLO implementation used, relied on different configuration files to be set up to fit the experiment and data. In order to start training the YOLOv3 network we first needed to ensure the data was setup correctly for the algorithm. The YOLO algorithm requires several files to be able to start training. It needed a .data file to point to the data location, a configuration file which defined the layer structure of the neural network and a weights file which loads previously trained weights into the network. In addition to these files the .data file contained the path to two additional require files namely the train.txt and val.txt files. These two files need to contain a list of paths to the actual image data the network

is supposed to use. The configuration files used in this experiment are available in the following git repository, `https://github.com/Imingen/master-sheep`.

4.2.2 RGB preprocessing

As mentioned earlier all of the RGB images were in 4k resolution. 4K resolution increase the training time for YOLO dramatically, as it has to downscale each input image to fit within the resolution of the network. One approach that could be taken to solve this problem was to downscale the images to a lower resolution before feeding them to the network. While this is a simple solution, the images risk loosing some of its detail and information. This is especially true for images taken at higher altitudes and this will in turn affect the performance of the network. Another approach was to split the images. This is a more cumbersome approach, but there is no risk of loosing any detail. As well as speeding up the training process, this approach also has a positive side effect in that it will generate a lot of images with no sheep in it. When the images were taken, the drone operators only took images when they found sheep. It can also be beneficial to the network performance to have images that does not contain any of the target objects it is being trained to recognize. The RGB images were cropped by using a python script developed by another master student group, with a similar thesis problem (Magnus Falkenberg Nordvik, 2020). The script calculated how many smaller images would be necessary to cover the entire area of the 4k image while ensuring a minimum of 200 pixels overlap. An image with a resolution of 4056x3040 required 5 images to cover the width and 4 images to cover the height while ensuring a minimum of 200 pixels overlap. Therefore each 4k image was split into a set of $4 * 5 = 20$ images, the exact areas with overlap can be seen in Figure 4.4.

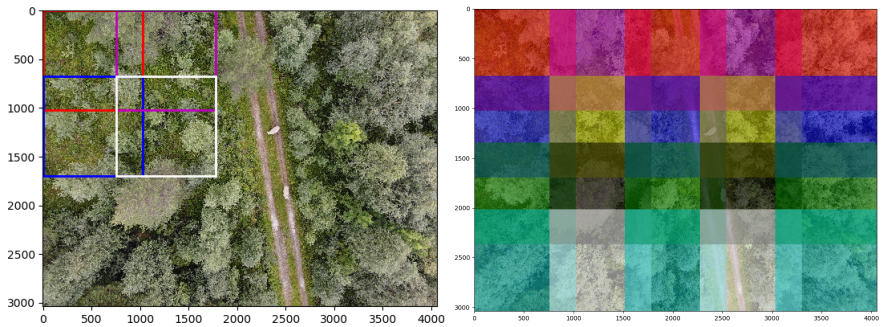


Figure 4.4: Left image shows partially how and where the 4k image was cropped, and the right shows how the whole image was cropped, blending colors shows overlap.

4.2.3 Infrared preprocessing

Since the infrared images were already in 480p the images could simply be input to the network as is, the network would then rescale them to the resolution stated in the config file. The operator of the drone also decided to turn on and off FLIR MSX during the collection of data, causing the images to look significantly different and have varying

amounts of detail. FLIR MSX is a program which adds details from the optical image to the infrared image (FLIR Systems, 2020).

A subset of the dataset consisted of images taken in october 2019 when there was snow on the ground, the subsequent temperature difference could impact the networks performance, therefore a seperate subset of the october 2019 images was created to test the networks performance under colder conditions.

To better understand the results the algorithm had produced, it was necessary to know the altitude of the drone when taking each individual image. In particular the infrared camera was unlikely to perform well at higher altitudes, therefore testing the infrared at lower altitudes was deemed necessary to properly evaluate its performance.

4.3 Structure

Finding the perfect network settings can be difficult and cumbersome work, and it can take a lot of time just finding the perfect configuration. Thorough research was therefore important, and using knowledge obtained from the literature, previous master thesis and the code repository mentioned in 4.1.2, three different configurations for RGB images and three for infrared images were chosen to be tested.

4.3.1 RGB Images

Two of the networks trained on RGB images did not differ much in their configuration, where the biggest difference were the network size. One network (network 1) used 832 x 832 and the other (network 2) used 1024 x 1024. The network resolutions were recommended from the code repository, were it recommended a higher resolution in order to increase precision. The max number of objects an image can contain was also increased to 300.

The third network (network 3) was tested on a full model, with 5 yolo layers from the code repository. This modified model was recommended by AlexeyAB if the purpose was to train on both small and large objects, which was the case for this experiment. The network tries to predict objects at five different scales, as opposed to standard YOLO layout which detects at three different scales.

8193 images were used in total for training the RGB models. The dataset were then shuffled thoroughly and 6412 were used for training and 1781 were used for validation. An equal amount of images with sheep and images without sheep were used, in accordance with ”How to improve detection” in (AlexeyAB, b) repository. 760 images were used for testing, which was handpicked before creating the training and validation sets. The networks where then trained on the IDUN cluster and each network ran for 9000 epochs.

Table 4.1: End result after training is complete on RGB images.

	precision	recall	mAP@50
network 1	0.93	**0.77**	**86.20%**
network 2	**0.94**	0.72	84.47%
network 3	0.93	0.69	81.64%

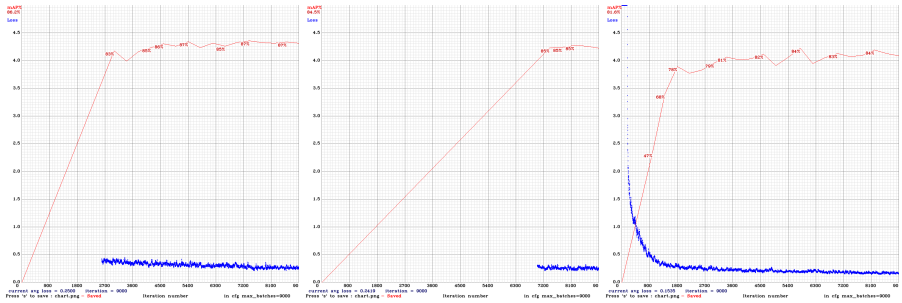


Figure 4.5: Training graphs for the RGB images.

Due to the training process being interrupted on network 1 and network 2, the graphs shown in figure 4.5 are not complete. The network was started again, from the previous stopping point and the graphs were then redrawn by the program. Total runtime was also not calculated, as the network did not run until it was manually started again, which could take up to 10 hours depending on the time it stopped. It is however safe to say that total runtime of network 1 and 2 exceeded multiple days, as network 3 ran continuously for 2 days and 20 hours.

4.3.2 Infrared images

Since the dataset had a large variety in the altitude the images are taken at it was important to chose the network configurations best suited to handle the variation. One of the relevant network configurations was the 5l network, which will increase accuracy on images of varying size (AlexeyAB, a). This is because it has five detection layers which attempt to predict objects at different scales, as opposed to the default configuration which has three. Similarly the tiny 3l configuration is designed to work better for varying sizes of objects and will also have faster prediction speed. However the networks smaller size will likely impact performance negatively. The yolov3-spp network configuration had the highest performance on the COCO dataset (AlexeyAB, a) meaning it could perform better on our dataset as well, while also being useful to compare the 5l and 3l-tiny network to.

The training set consisted of 1467 images where 80% was was used as training data, the remaining 20% was split equally into validation and test sets. As with the RGB images, half of the dataset consisted of negative samples. The training and validation set were used during training while the test set was saved in order to check the finished networks performance. Each network configuration was then trained on the IDUN cluster(Själander et al., 2019).

Each network reduced the loss score to under two after approximately 1000 epochs (See Figure 4.6), All three of the networks were run for 6000 epochs, where the 5l and 3l network settled at a loss value of approximately one while the spp network was about 0.6. The 3l-tiny network spent the least amount of time training completing after 11 hours and 30 minutes, followed by spp which ran for 26 hours and lastly the 5l network ran for 47 hours.

Table 4.2: End result after training is complete on infrared images.

	precision	recall	mAP@50
5l network	**0.86**	0.62	**79.62%**
spp network	**0.86**	**0.78**	78.56%
3l-tiny network	0.75	0.64	63.74%

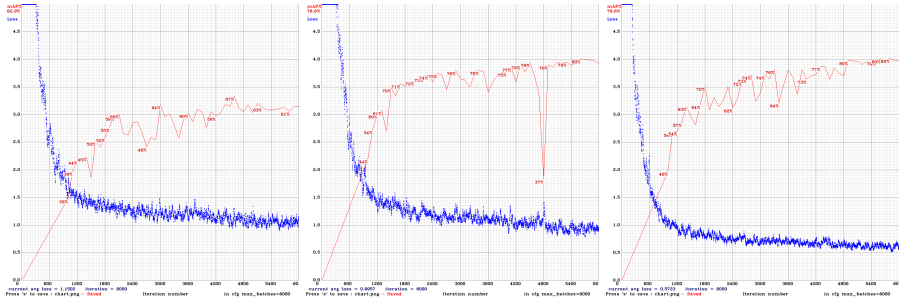


Figure 4.6: loss and mAP from 3l network during training. mAP score was run on validation data every 100 epoch from epoch 1000 to 6000. Higher resolution graphs are proviede in the appendix.

Chapter 5

Results & Analysis

This chapter presents the results of all the networks, as well as analyse the results and compare RGB and infrared results to each other. The networks are also tested on more specific test data. This is done to check if there are some environments where either RGB or infrared images yield different results as opposed to the test data that provides the baseline.

5.1 Results

5.1.1 RGB results

The test set used for RGB images were manually made. Images from different environments and heights were handpicked in order to create a diverse test scenario. Some images without sheep, but with confusing content like white snow patches and dark shadow was also picked in order to try and confuse/stress test the networks.The test set contained 760 images in total.

Table 5.1: Network performance on the test dataset, bold indictating the best results

	precision	recall	mAP@50	TP	FP
832x832	0.89	0.59	76.61%	200	25
1024x1024	**0.91**	**0.63**	**80.33%**	**214**	**22**
5l	0.85	0.60	75.16%	202	35

Looking at the table above, the 1024x1024 network shows best results. It achieved a precision of 0.91 and mAP@50 of 80%. All networks achieve fairly high precision, meaning all sheep that were detected have a high chance of actually being sheep. They do however perform worse on recall rate, meaning a lot of sheep were not actually detected. Positively, the high precision means that the networks will rarely notify the shepherd with a false detection. After images with detected bounding boxes were generated, there were

no images where at least one sheep was not detected, but there were many sheep which was not detected. The network did however struggle with precise bounding boxes and it was easy to see how the network got confused. In figure 5.1 its obvious how some of the boxes cover more than one sheep, and the boxes are not as precise, but it does find multiple sheep. In figure 5.2 its clear how the contrast of a grassy field help the network. It also shows how a white rock confuses the network, with 65% confidence in detecting a sheep. The last figure (figure 5.3, shows another wood terrain, but also shows how a shadow confuses the network. With a confidence of 51% it is labeled as a sheep. Bounding boxes were generated from the 5l network.

Figure 5.1: The left image showing a big flock and the right showing three sheep in the woods.

Figure 5.2: The left image showing the good contrast the green grass gives, while the right show a false positive with a white stone.

Figure 5.3: The left image show another woodland terrain, while the left show how the networ struggles with shadow.

The networks were also tested on a set of images with snowy and cold background. This set consisted of 54 images, in order to compare with the infrared set.

Table 5.2: Network performance on images with only cold/snowy ground, bold indicating the best results.

	precision	recall	mAP@50	TP	FP
832x832	0.94	**0.57**	81.72%	**82**	5
1024x1024	0.94	0.47	**82.93%**	67	4
5l	**0.97**	0.47	69.10%	68	**2**

On snowy backgrounds, the results are not as clear. The 5l network got the highest precision, but the 832x832 network achieved the best recall with 0.57, beating the two other networks with 10%. However, the mAP@50 metric was best for the 1024x1024 network with a 82%, and the 5l network was the worst with 69%. The 832x832 network got overall the best results, with a higher recall and true positives than the other networks, while still being close in mAP@50 and precision. The networks gain a few points in accuracy on the previous test, showing that there are fewer false positives in snowy backgrounds. However, all networks loose some points in recall, showing that it is harder to find all the sheep in a cold/snowy environment. One reason for this could be that it struggled with finding one particular wool color. As both figure 5.4 and figure 5.5 show, the network really struggles with darker wool color. This is surprising considering it detected a shadow as a sheep in figure 5.3. This is clearly the reason the network got a poor recall rate, as it almost never detected any sheep with darker wool color. The higher precision in this environment seems to be from the fact that there are less false positives.

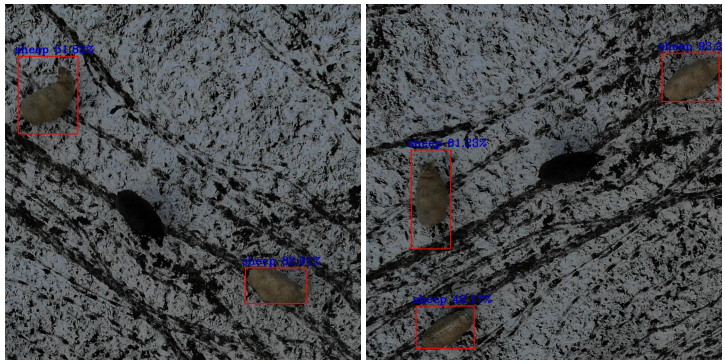


Figure 5.4: Images show how well it detects white sheep, but no black sheep.

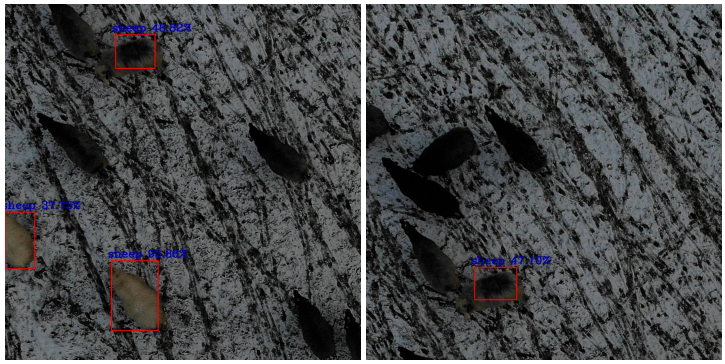


Figure 5.5: White sheep seem to give good result, but the network struggle with darker wool.

5.1.2 Infrared results

The spp network had the best performance on the test data taken from the original dataset, although the 5l network had fewer false positives and a slightly higher precision. The 3l-tiny network had significantly lower results than both the 5l and spp network (See Table 5.3). It was very clear that it would be easier to spot sheep in the infrared images in comparison to the cold/snowy environment, which is likely due to the increased temperature difference between the sheep and the background. Therefore superior results were expected on the cold/snowy dataset. The infrared networks performed significantly better than the previous test set on this dataset with the 5l network reaching a recall of 0.84 and mAP@50 of 93.58%. The 5l had 230 false positives beating both the 3l and spp networks. Unexpectedly the 3l tiny network managed a recall of 0.85, higher than both the spp and 5l network while predicting faster than both. However it had worse mAP than the 5l network and made a total of 514 false positive predictions. Lastly the spp network performed worse than both the 5l and 3l network with a recall of 0.81 and mAP of 88.39%, false positives were at 389, lower than the 3l but still higher than the 5l (See Table 5.4). Some of the networks better predictions in practice can be seen in Figure 5.6. The network was

successful in locating most of the sheep in these images while managing to avoid similar objects such as rocks. On the other hand Figure 5.7 shows some of images where the network clearly failed to detect sheep a human would have been able to. Lastly Figure 5.8 shows how well the network performs on the colder images, while also failing at detecting sheep which seem simple to detect. All the bounding boxes were generated with the 5l network.

Table 5.3: Network performance on the test dataset, bold indicating the best results.

Network	precision	recall	mAP@50	TP	FP
5l network	**0.84**	0.55	75.65 %	222	**42**
3l tiny network	0.72	0.59	61.34 %	236	92
spp network	0.82	**0.71**	**77.44** %	**285**	62

Table 5.4: Network performance on images with only cold/snowy ground, bold indicating the best results.

Network	precision	recall	mAP@50	TP	FP
5l network	**0.89**	0.84	**93.58** %	1819	**230**
3l tiny network	0.78	**0.85**	88.74 %	**1854**	514
spp network	0.82	0.81	88.39 %	1769	389

5.2 Infrared performance in different altitudes

After inspecting the results of the infrared images, a pattern of high altitude images performing noticeably worse than the lower images was discovered. Therefore further testing was needed to ensure these observations were not anecdotal. However according to one of the data collectors the altitude section of the datasets EXIF altitude data was incorrect, meaning the only place to acquire altitude data was in the drones flight logs. Unfortunately the images taken in the cold/snowy environment seemed to have lost the EXIF timestamp making these images unusable for this test. Considering the networks improved performance on these images, lacking these will likely impact any comparison to the previous datasets. Additionally for some of the images the logs did not have a height measurement at the respective images timestamp meaning these image had to be discarded leaving only 88 images in the test set. The test set was split at 50 meters, where 26 images was above and 62 was below. The 5l and tiny 3l both performed better on the image sets below 50 meters, for these two networks it is clear that lower altitudes improve performance. However the spp network achieved higher mAP and precision on the higher images (See Table 5.5).

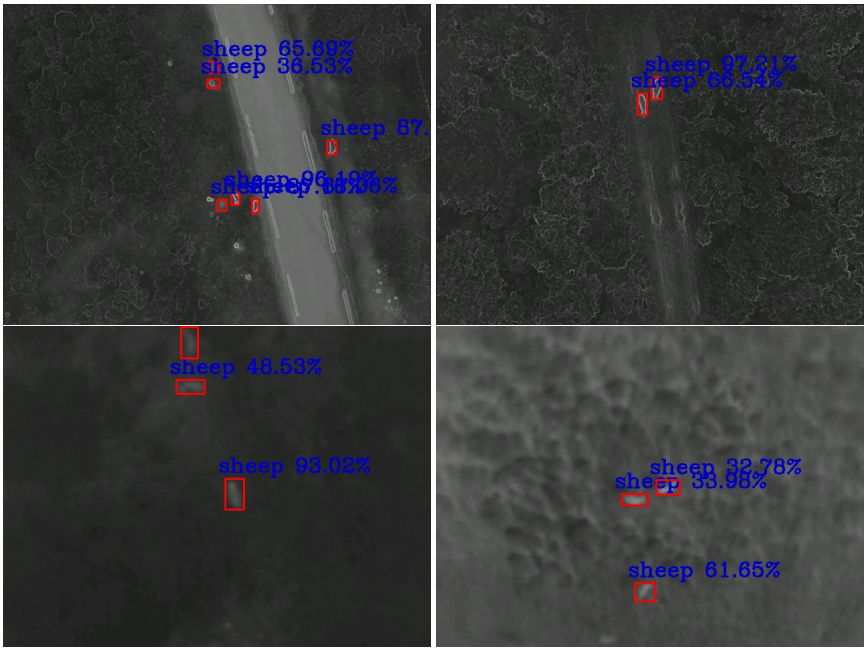


Figure 5.6: Good detection results from the normal test set using the 5l network. Upper left detects most of the sheep while not detecting the nearby rocks.

Table 5.5: Network performance on the test set randomly separated from the training set and split based on height.

Network	precision	recall	mAP@50	TP	FP
5l below	0.87	0.60	85.58 %	34	5
5l above	0.80	0.42	63.21 %	20	5
3l tiny below	0.80	0.65	69.21%	37	9
3l tiny above	0.67	0.50	54.57%	24	12
spp below	0.74	0.70	74.79%	40	14
spp above	0.86	0.67	77.20%	32	5

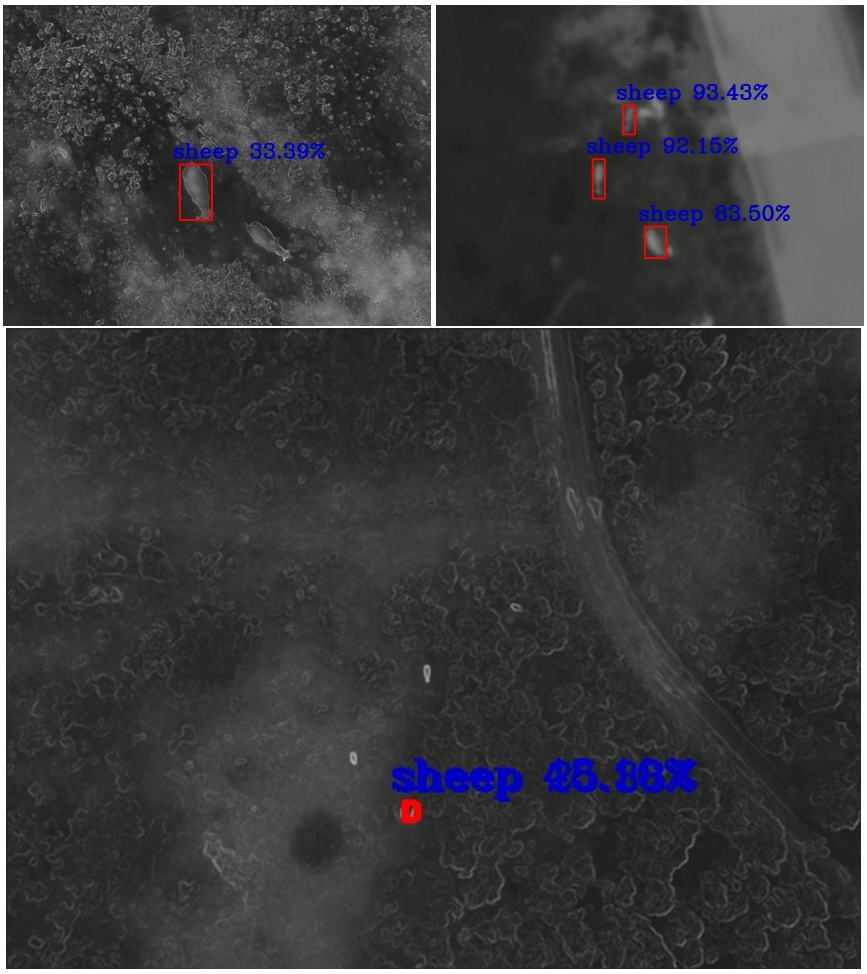


Figure 5.7: Bad detection results from the normal test set using the 5l network. Upper left fails to detect any sheep at all, while the right misses two. Lower image misses several.

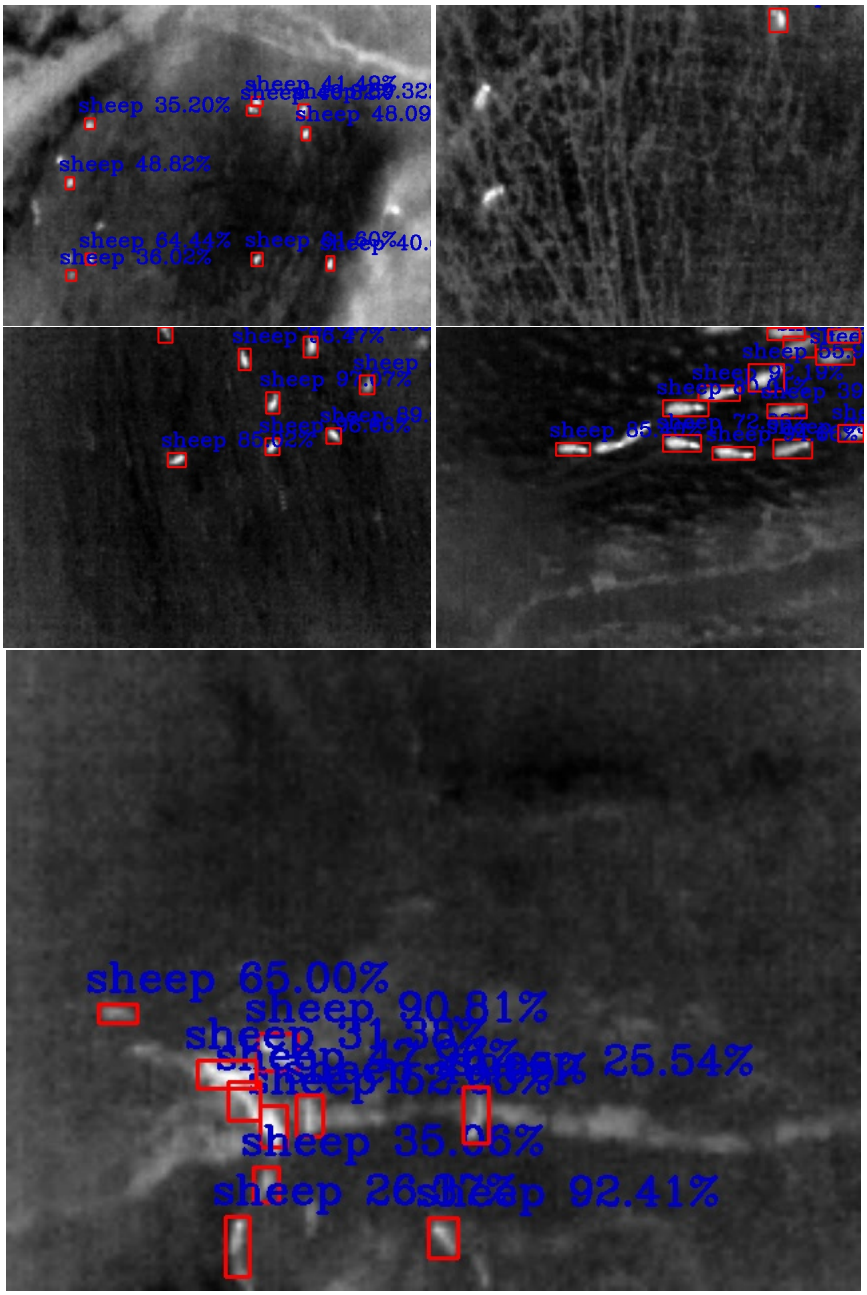


Figure 5.8: Good and bad examples of the 5l networks test results on the cold/snowy images.

5.3 Discussion

The results clearly show that the networks perform very similar on the normal test dataset. The infrared networks seems to yield better recall, but the RGB networks give better precision and mAP@50. This might be a result of the sheep being more distinct in the infrared images, meaning that it is easier for the network to distinctly detect the sheep in the images. Comparing the 5l network in both instances, it is clear that it performs slightly better on RGB images on the normal test set. It does however perform way better on infrared images with a cold/snowy background, with almost double recall value. Overall, the infrared images gives a huge performance boost when the background is cold. This might show that the infrared is better in general, if some precautions are taken.

One clear issue that the infrared images had, was that the images performed worse on images taken at higher altitudes than 50 metres. Although the altitude difference seems to impact the spp network less than the 5l and 3l tiny networks, the spp still found a higher percentage of sheep in the lower altitude images. It is unclear how big this issue is and if the trade off is impactfull at all. On one note, the pilot would have to fly lower in order to get the best performance that infrared images can give, but would at the same time cover less area in the images taken. The RGB images do not have the same issue because of their high resolution, meaning as they are cropped and thus cover a reduced field of view they will still retain enough detail for the sheep to be clearly visible, assuming the sheep was clearly visible from that location.

The loss of performance on higher altitudes is not as severe as to completely discard the infrared images. The difference on the basic dataset is not as severe as well, and the infrared keeps a higher recall rate while the RGB has a higher mAP@50. This changes in a cold/snowy environment where the infrared images yield a much higher mAP@50 and recall rate. Usually, the mAP@50 metric is the most important, but in this scenario we are more concerned about finding sheep and not the exact location. Overall, it seems that the infrared images yield slightly better results. The difference in a general environment is not as huge, but the gain in a specific, infrared friendly environment is large.

5.3.1 Research questions.

RQ1: How well does YOLOv3 perform in detecting sheep in highland terrain? The best network configuration of YOLOv3 acheives a recall of 0.71 locating 71% of any sheep visible in an image. Out of 347 sheep detections 285 were actual sheep and 62 were false positives. YOLOv3 ability to detect sheep is significant, however with room for improvement.

RQ2: Do infrared images improve the detection of sheep as opposed to RGB images? Infrared images improved detection on cold background, indicating that the right environment might make infrared images superior in detecting sheep. Additionally it does not lose to much on a dataset containing a diverse set of environments, and it still keeps a higher recall

rate than RGB images. In general, infrared images seem to be a good candidate in sheep detection.

5.3.2 Validity of the experiment

While the detection networks performed well on the test datasets used in this experiment there are several questions on how valid these results are, for example how would it function in different environments or simply on a different dataset.

The use of FLIR MSX

Since the infrared dataset consisted of many images where FLIR MSX was used it is unknown to what extent the networks performance would be affected on a dataset with only FLIR MSX or normal infrared images. The effect of FLIR MSX can be seen in Figure 5.9 MSX makes the location of some sheep even more clear, however it does not accomplish this for all sheep and will highlight not only sheep but other miscellaneous objects as well, such as stones or buildings possibly increasing false positive detections.

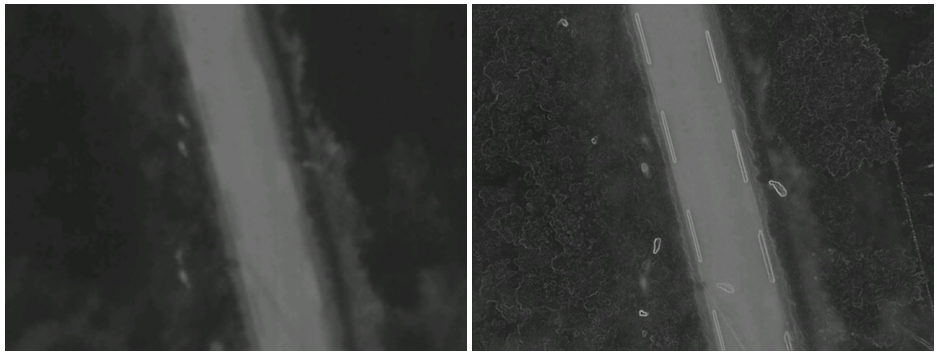


Figure 5.9: Two images taken at the same spot within a minute of each other, the image on the left did not have MSX while the one on the right used FLIR MSX.

Ability to generalize

While the test sets were separated from the training and validation sets before the training of the networks, they were still from the same UAV flights meaning their terrain and environments were the same. Therefore these test results do not necessarily represent the networks performance under different conditions.

Different network configuration

One main issue with this experiment is that different network configurations were used, and might make the results incomparable. One network, the 5l configuration, was used on both RGB and infrared images and is therefore the most interesting to compare. On

the baseline test set, the 5l network performs slightly better on RGB images. It does however get a huge performance boost on infrared images in cold environments. It would be interesting to test similar configurations on both types of images, to see if the result would be different. In hindsight, the test should have either been 3 similar network configuration or use all 5 different configurations on both infrared and RGB images. This could potentially show a bigger difference between images, making one type of camera redundant.

5.4 Further Work

5.4.1 YOLOv4

On the 23rd of April 2020, an improved version of YOLO was released, YOLOv4 (Bochkovskiy et al., 2020). This paper was written by the creator of the repository used in this experiment. This version claims to run twice as fast as other detectors, while still keeping comparable results. It also improves 10% on YOLOv3's average precision and 12% on FPS. Using YOLOv4 might give better results than the version used in this thesis and might show great results on live video feed. It would also be interesting to see a fully integrated system, where the system not only detects sheep in images, but also notifies the user with the location of the detected sheep on a map.

5.4.2 Data collection time

Considering the infrared camera's increased performance on the colder environment, future research on the usage of infrared camera's should focus on collecting data during the coldest time of day to ensure a significant difference in temperature between the sheep and the background.

5.4.3 Improving infrared quality

In order to improve infrared performance, the infrared images needs to be of a higher quality. Currently, the infrared images need a specific environment in order to beat the performance of RGB images. The infrared images have a higher recall rate, but lower precision than RGB images. If it could keep its recall rate but increase its precision it would be a strong contender.

Chapter 6

Conclusion

The main goal of this thesis was to test YOLOv3‘s ability to detect sheep on UAV imagery in order to help shepherds gather sheep in Norwegian sheep farming terrain. Additionally, RGB and infrared images were compared in order to see which one yields the best results. Overall, the infrared images gave the best results, with the YOLOv3 spp network achieving a recall of 0.71 and a mAP@50 of 77.44% on the infrared images. In comparison the RGB images achieved a recall of 0.63 and a mAP@50 of 80.33% using the default YOLOv3 configuration on the normal test set. When testing on cold backgrounds the infrared performance was boosted significantly, with the 5l network achieving a recall of 0.84 and a mAP@50 of 93.58%. On the RGB images the networks performance was slightly reduced achieving a recall of 0.57 and a mAP@50 of 81.72%. The quality of the infrared images do get worse at higher altitudes, and thus the results are worse on infrared images at higher altitudes. Further testing on infrared environments, and a better infrared camera would be favorable in order to see if infrared cameras are superior to RGB.

The results do give a skewed image, as the network configurations are not similar in RGB and infrared scenarios. Only one similar network was tested, and it would be interesting to see how similar networks would perform.

Further work would benefit from testing improved versions of YOLO such as YOLOv4, a more diverse dataset and better infrared images. On board testing would also be interesting or testing on a live video feed, as YOLO claims quick inference time and high FPS.

Bibliography

AlexeyAB, a. Alexeyab / darknet. https://github.com/AlexeyAB/darknet. Accessed on 02/10/2019.

AlexeyAB, b. Alexeyab / darknet. https://github.com/AlexeyAB/darknet#how-to-improve-object-detection. Accessed on 02/10/2019.

Andrew, W., Greatwood, C., Burghardt, T., 2017. Visual localisation and individual identification of holstein friesian cattle via deep learning, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 2850–2859.

Blix, A., Vangen, O., 2019. Sau. https://snl.no/sau. Accessed on 11/11/2019.

Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 .

Chabot, D., Bird, D.M., 2015. Wildlife research and management methods in the 21st century: Where do unmanned aircraft fit in? Journal of Unmanned Vehicle Systems 3, 137–155.

DJI, 2019. Mavic 2 Enterprise Series User Manual-EN. DJI.

FLIR Systems, 2020. What is msx®? https://www.flir.com/discover/professional-tools/what-is-msx/. Accessed on 23/05/2020.

FLIR Systems Inc, . Lov om dyrevelferd. https://www.flir.com/products/lepton/. Accessed on 09/03/2020.

van Gemert, J.C., Verschoor, C.R., Mettes, P., Epema, K., Koh, L.P., Wich, S., 2014. Nature conservation drones for automatic localization and counting of animals, in: European Conference on Computer Vision, Springer. pp. 255–270.

Gonzalez, L.F., Montes, G.A., Puig, E., Johnson, S., Mengersen, K., Gaston, K.J., 2016. Unmanned aerial vehicles (uavs) and artificial intelligence revolutionizing wildlife monitoring and conservation. Sensors 16, 97.

Google-Maps, 2020a. 62.693465, 9.094728 - google maps. https://www.google.com/maps/place/62%C2%B041'36.5%22N+9%C2%B005'41.0%22E/@62.693465,9.0925393,508m/data=!3m2!1e3!4b1!4m11!1m4!3m3!1s0x0:0x0!2zNjLCsDQwJzU3LjYiTiA5wrAwOCcxOS45IkU!3b1!3m5!1s0x0:0x0!7e2!8m2!3d62.6934647!4d9.0947283. Accessed on 28/05/2020.

Google-Maps, 2020b. 63.403771, 10.464441 - google maps. https://www.google.com/maps/place/63%C2%B024'13.9%22N+10%C2%B027'56.6%22E/@63.403848,10.4635183,496m/data=!3m2!1e3!4b1!4m13!1m6!3m5!1s0x466d3038626e6b93:0x894ea08c53ba31b7!2sDragvoll!8m2!3d63.4089983!4d10.4708427!3m5!1s0x0:0x0!7e2!8m2!3d63.4038482!4d10.4657066. Accessed on 28/05/2020.

Labelbox, 2020. Labelbox. https://labelbox.com. Accessed on 06/01/2020.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: European conference on computer vision, Springer. pp. 21–37.

Magnus Falkenberg Nordvik, Jens Tobias Kaarud, H.R.P., 2020. Mastergradsarbeid 2020. Master's thesis. Norwegian University of Science and Technology.

Muribø, J.H., 2019. Locating Sheep with YOLOv3. Master's thesis. Norwegian University of Science and Technology.

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J., 2019. Deep learning vs. traditional computer vision, in: Science and Information Conference, Springer. pp. 128–144.

pjreddie, . pjreddie / darknet. https://github.com/pjreddie/darknet. Accessed on 02/10/2019.

Radovic, M., Adarkwa, O., Wang, Q., 2017. Object recognition in aerial images using convolutional neural networks. Journal of Imaging 3, 21.

Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A., 2015. You only look once: Unified, real-time object detection. CoRR abs/1506.02640. URL: http://arxiv.org/abs/1506.02640, arXiv:1506.02640.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 .

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, pp. 91–99.

Russell, S., Norvig, P., 2009. Artificial Intelligence: A Modern Approach. 3rd ed., Prentice Hall Press, USA.

scikit-learn developers , 2020. Precision-recall. https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html. Accessed on 05/05/2020.

Själander, M., Jahre, M., Tufte, G., Reissmann, N., 2019. EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure. arXiv:1912.05848.

SSB, 2019. Fakta om jordbruk. https://www.ssb.no/jord-skog-jakt-og-fiskeri/faktaside/jordbruks. Accessed on 11/11/2019.

Stanford, 2020a. Cs231n convolutional neural networks for visual recognition. https://cs231n.github.io/convolutional-networks/. Accessed on 02/02/2020.

Stanford, 2020b. Cs231n convolutional neural networks for visual recognition. https://cs231n.github.io/transfer-learning/. Accessed on 02/02/2020.

Tzutalin, . Labelimg. https://github.com/tzutalin/labelImg. Accessed on 15/10/2019.

Wang, L., Zhang, Z., 2017. Automatic detection of wind turbine blade surface cracks based on uav-taken images. IEEE Transactions on Industrial Electronics 64, 7293–7303.

Appendix

6.1 The training graphs of Figure 4.5

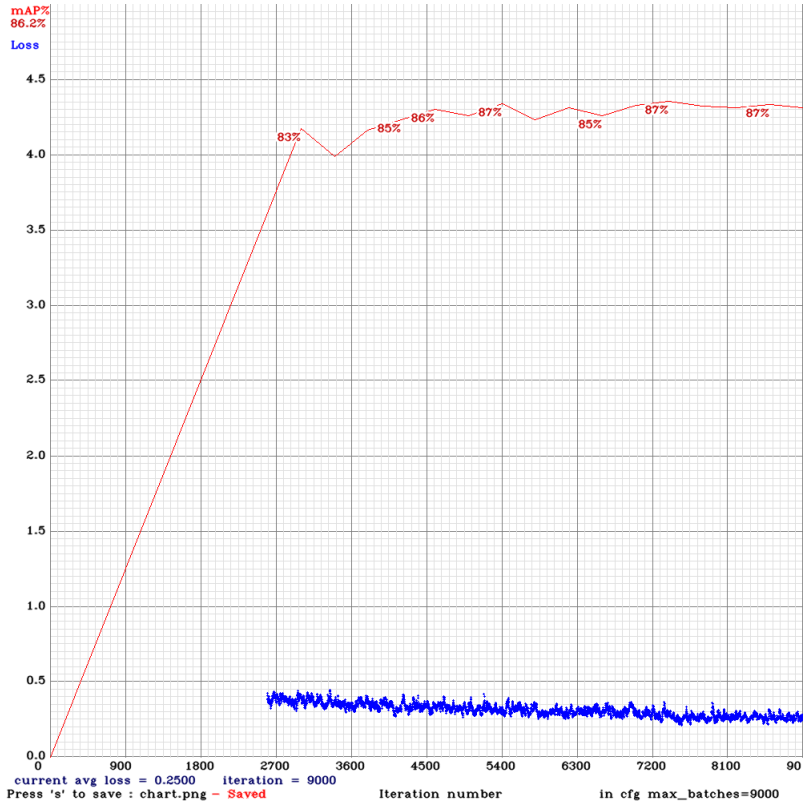


Figure 6.1: Loss and mAP from the 832x832 network during training. mAP score was run on validation data every 100 epoch from epoch 1000 to 9000. Training was interrupted multiple times so graph is incomplete.

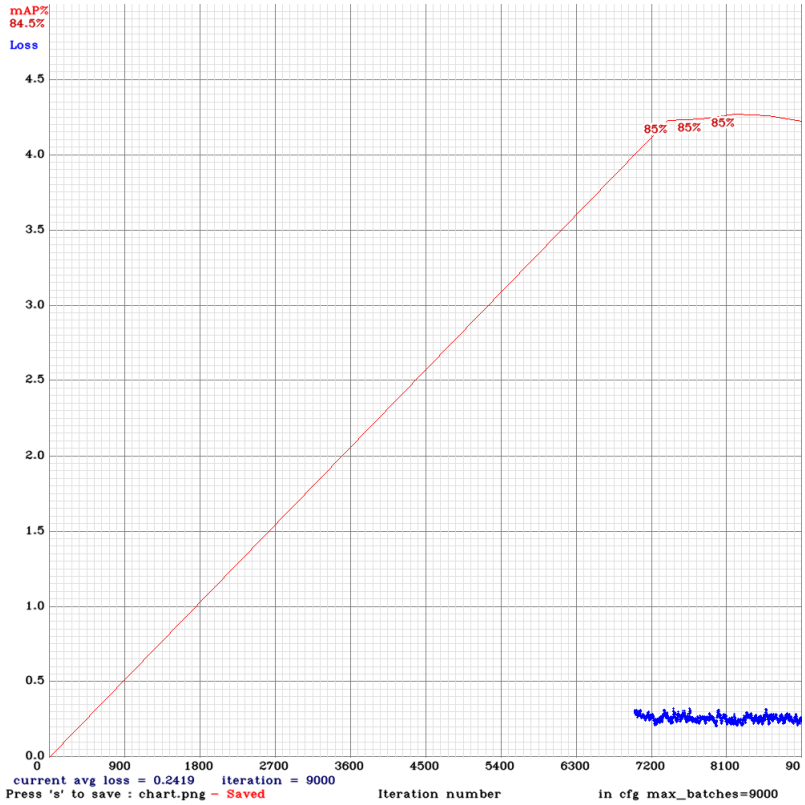


Figure 6.2: Loss and mAP from the 1024x1024 network during training. mAP score was run on validation data every 100 epoch from epoch 1000 to 9000. Training was interrupted multiple times so graph is incomplete.

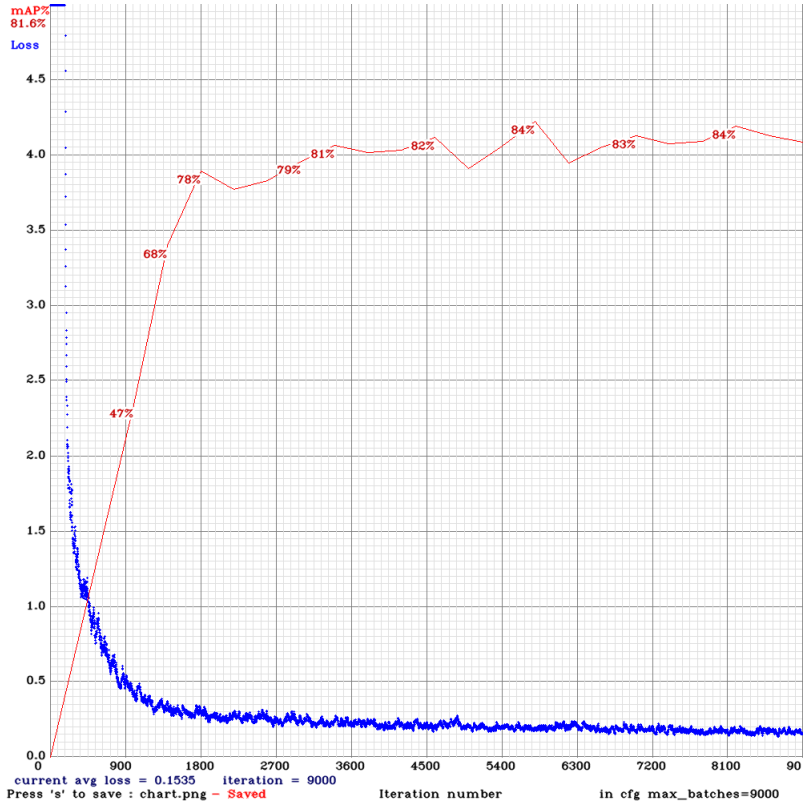


Figure 6.3: Loss and mAP from the 5l network during training. mAP score was run on validation data every 100 epoch from epoch 1000 to 9000.

6.2 The training graphs of Figure 4.6.

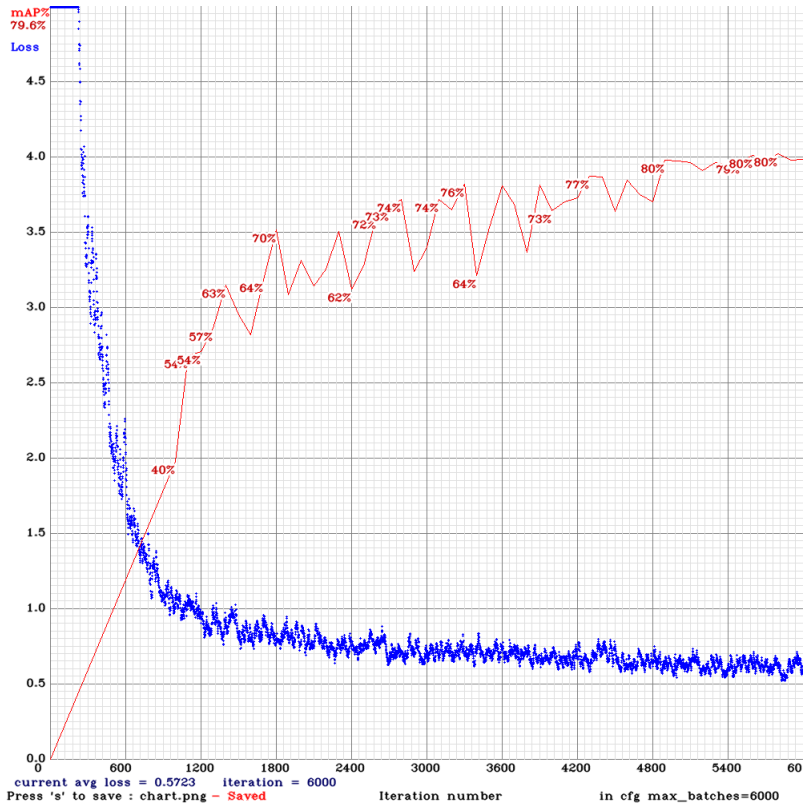


Figure 6.4: Loss and mAP from the 5l network during training. mAP score was run on validation data every 100 epoch from epoch 1000 to 6000.

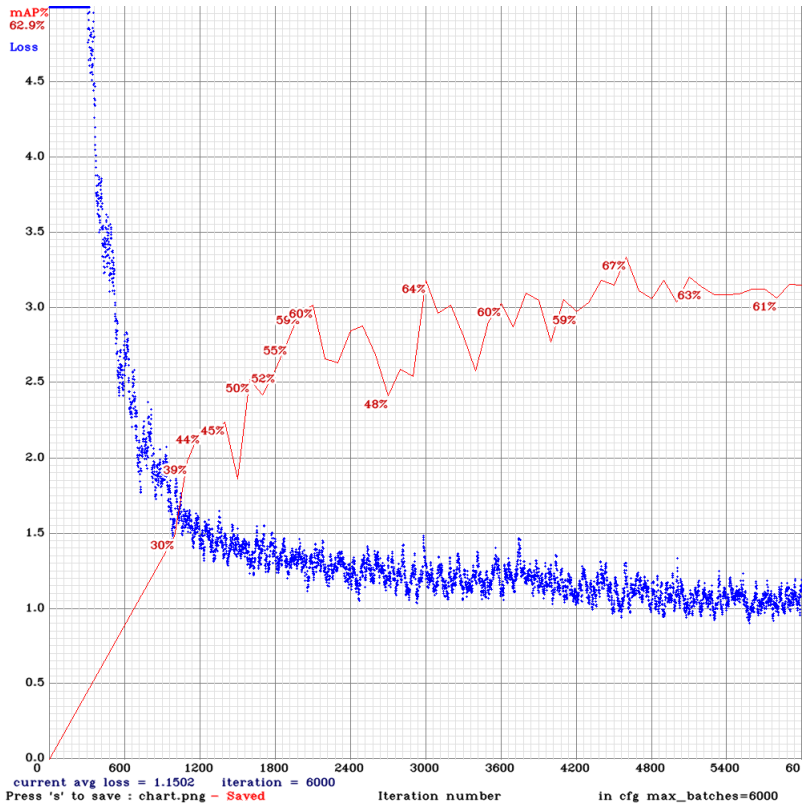


Figure 6.5: Loss and mAP from 3l network during training. mAP score was run on validation data every 100 epoch from epoch 1000 to 6000.

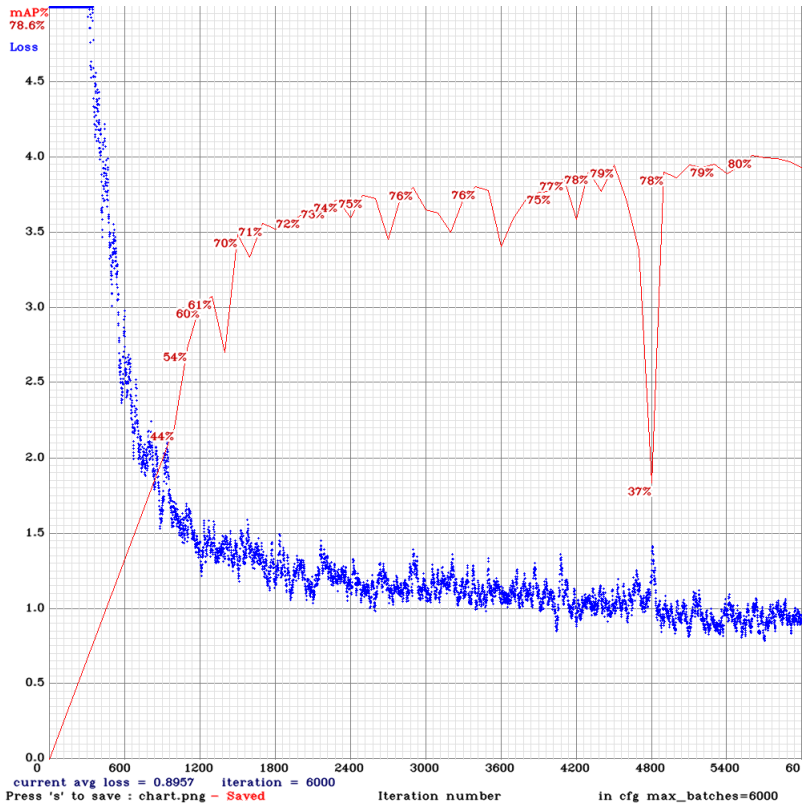
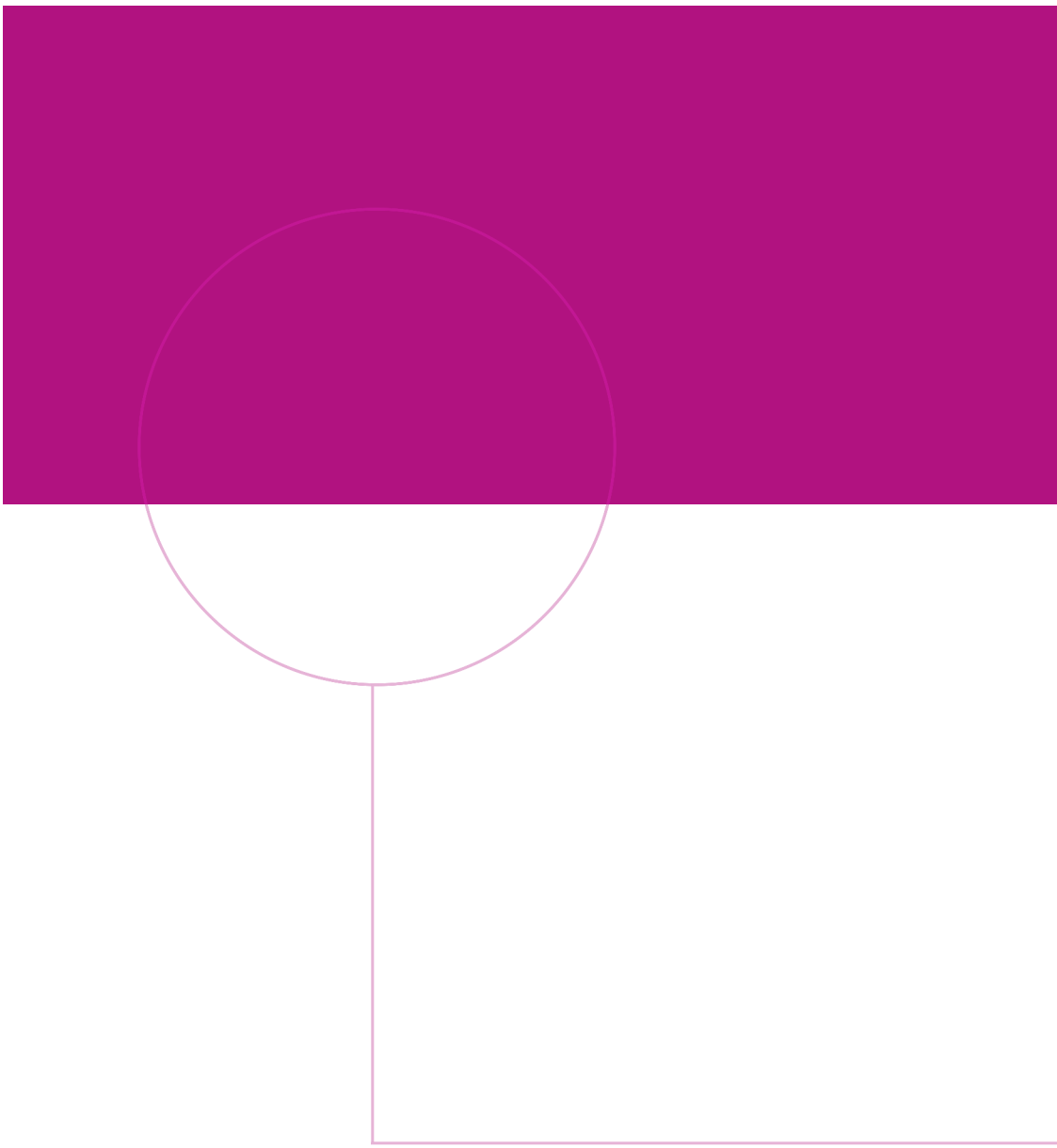


Figure 6.6: Loss and mAP from the spp network during training. mAP score was run on validation data every 100 epoch from epoch 1000 to 6000.



NTNU

Norwegian University of
Science and Technology