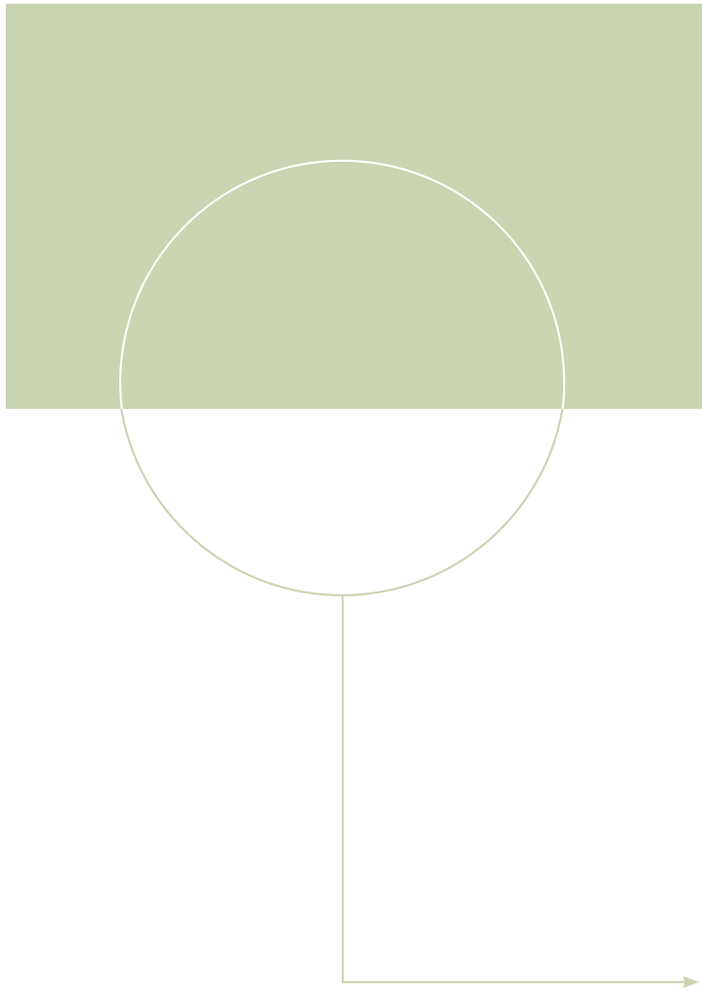


Master's thesis

2021

Håkon Sørensen Bøckman

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

Håkon Sørensen Bøckman

Locating sheep in the highlands with aerial footage and a lightweight algorithm system

September 2021

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Locating sheep in the highlands with aerial footage and a lightweight algorithm system

Håkon Sørensen Bøckman

Master of Science in Informatics
Submission date: September 2021
Supervisor: Svein-Olaf Hvasshovd
Co-supervisor: Svein-Olaf Hvasshovd

Norwegian University of Science and Technology
Department of Computer Science

Locating sheep in the highlands with aerial footage and a lightweight algorithm system.

Håkon Sørensen Bøckman

September 1, 2021

First I would like to thank my supervisor Svein-Olaf Hvasshovd, who have been contributed with input, guidance during this project even during his holidays. I would also like to thank Kari Meling Johannessen, Magnus Guttormsen for operating the drone, and creating the basis of the dataset which was the base of this project.

Abstract

The challenges the sheep farmer is facing in today's husbandry are physical heavy, slow work and it belongs to the time of the past.
One of the larger challenges the farmer is facing is when he or she are rounding up the animals from the pastures in the highlands in the fall.
Certain amount of animals get lost and the farmer needs to acquire them.
As of 2017 13% on average animals national wide are lost in a season, due to various reasons.
The farmer is bound by law to document and follow through on each animal, this can often be a lengthy and costly process.
Since the sheep travels across large pastures which are vast areas of land it can take several days and many man-days for a farmer to find the lost animals.

The usage of unmanned aerial vehicles (UAV) in commercial use have increased over the years, where its applications are ever so increasing as the UAVs components get more affordable and smaller.
UAVs are a capable development platform for new applications as they are fairly inexpensive, comes with low risk to personnel, well suited for image recognition systems as they are often electric powered, steady and increasingly mobile.
Today there is found several uses of this type in agriculture, power line maintenance and search and rescue in Norway.

In this thesis I aim to explore the potential possibilities of a using an light weight image recognition algorithm for identifying sheep on the pasture in the highlands, which a drone can carry and power by itself.
The drone is equipped with a high resolution camera taking 12Mp RGB pictures and a secondary lens taking 0.307MP Infra Red pictures.
The drone is operated with previously students, when collecting data.
I choose a state of the art classification CNN(Convolution neural network) called EfficientNet. Which gives a promising results of $\approx$ 95% accuracy across the different networks, with different preprocessing steps.

Sammendrag

Utfordringene som dagens sauebønner møter i sitt yrke er fysisk tungt og tregt arbeid og arbeidsoppgaver som burde tilhører fortiden.
En av de større utfordringene en sauebonde møter er under innsamling av sau om høsten fra fjellet. Enkelte dyr går seg vill i løpet av sesongen og bonden må finne disse.
I 2017 var tap av dyr på fritt beite 13% , av forskjellige grunner.
Bonden er pålagt av loven å måtte dokumentere utfallet av vært enkelt dyr, noe som kan ofte bli en lang og dyr prosess.
Siden sauene beveger seg over store beiteområder som er store landområder kan det ofte ta sauebonden flere dager og mange dagsverk før bonden finner det bortglemte dyret.

Bruken av unmanned aerial vehicles (UAV) i kommersiell sammenheng har økt over årene. Bruken øker siden UAV komponenter blir stadig billigere og mindre. UAV'er har en god utviklings plattform for nye kommersielle løsninger siden de er nokså billige, har en lav risiko for personell og egner seg godt for bildegjenkjenningssystem da de ofte er elektriske, stødige og alltid mobile.
I dag er det flere løsninger av denne typen i bruk i jordbruk, inspeksjon avhøyspentkabel og i redningsoperasjoner i Norge.

I denne oppgaven sikter jeg meg inn på å utforske det mulige potensialet ved å bruke en lettvekts algoritme for gjenkjenning av sauer på beite i fjellet, hvor dronen kan drifte systemet av seg selv.
Dronen er utstyrt med et høyoppløsnings kamera som tar 12Mp RGB bilder og et sekundært linse som tar 0.307Mp Infrarøde bilder.
Dronen har blitt brukt av tidligere studenter ved innsanking av data.
Jeg har valgt et toppmoderne CNN (konvulsjon neuralt nettverk) som heter EfficientNet. Som gir lovende resultater på $\approx$ 95% nøyaktig på tvers av de forskjellige nettverkene med de forskjellige preprosessering steg.

Contents

Figures

Chapter 1

Introduction

In the summer season sheep farmers across Norway release their herd around 2 million of animals to the pastures for grazing over the season.
While the livestock is away on pasture the farmer sow grass on local fields at the farm to ensure food for the winter season.[1]
Depending on local varieties of exposure to the elements and the weather during the season the herd is out on the pasture for a period of 16 weeks.[2]
The sheep farmer is conducting weekly checkups and inspection of his herd to identify abnormal behaviour. This can differ from farmer to farmer and accessibility to the pasture and of the size of the pasture and the number of animals.
It is challenging to keep track on every sheep for the farmers as the sheep's are organized in flocks, a family-group of 8-10 animals which is lead by an older ewe with a bell around her neck.
The bell serves to help young lamb locate the ewe with sound when she moves on and in few cases the lamb have lost their way or ewe is not directly visible through eyesight.[3]

In the following fall in the period from September to October the farmers round up the animals from the pasture.
The date can vary from location to location within Norway as the northern parts are more prone to cold weather earlier in the fall and the local weather conditions is also an aspect in the decision of the farmer.
The general rule is that the farmers often tries to get their animals down from the pasture in the mountains before the first snow.[2]
The round up of animals are sometimes organized with several farmers, since some pastures are serving multiple sheep farmers as a pasture for the season.
In the round up process the farmers are herding their animals into a smaller fenced of area for processing. The farmer and its helper will eventually need to go out on the pasture to locate the few missing animals. As it happens the pasture is very large and the animals not very easy to spot.
This causes the farmer to spend a lot of resources and time into locating the animals.[1]

The farmer is bound to locate the animals by Norwegian Nature Surveillance to confirm their status which is under Norwegian Environment Agency. [4] [5]
According to NSG (Norsk Sau og Geit) the cost of losing a lamb is 1850 NOK and ewe is 3585 NOK.

The cost per animal differentiate between farmer to farmers by its nature the cost is bound to the scale of the farm i.e. total amount of sheep the farmer is owning.[6] [7]
The process of locating the animals are often not economically conventional for the farmer, as the cost of locating the animal is more than the animal itself.[8]

Unmanned aerial vehicle (UAV) typically consist of generalized aircraft design and a control system (CS) or sometimes a ground control system (GCS) whereas communication is relayed between the unmanned aircraft (UA) and the CS.
UAVs have been increasing in popularity and availability for commercial use and in private use due to advancements in electronics in regard to its ability to continuously smaller footprint, strong performance and lower cost.
The UAVs are often serving as a affordable and necessary tool to close range missions and other high risk areas where the risk of the pilot and cost of a full sized aircraft would not be feasible to take.
Despite modern UAVs continuously increases its performance there are limitations to range, battery and optics. [9] [10] [11] [12]
This is especially visible when heavy computational tasks are needed to be done on the UAV.
One solution to this is to use a form of a communication between the UAV and a server, where the server is doing the computational work of the task.
This is a viable approach but it meet challenges in delay, security and relaying on 4G cellular communication standard coverage. As of 30 of June 2020 the coverage of Norway was 83.6%, Where the coverage are prioritized for the high intensity populated areas. [13]

A big trend in machine learning (ML) the last 9 years have been deep learning.
After Alex Krizhevesky and his co authors won the annual LCVR2012(Large Scale Visual Recognition Challenge 2012) with their "alexNet" where they won with a extensive margin. [14]
The deep learning paradigm have had a increase in popularity as a result, and one of the better performing architectures are Convolution Neural Network (CNN) in field of unsupervised learning.

The motivation for this project is to use a popular light-weight algorithm to identify sheep from UAV footage to help the sheep farmer locate his lost animals on the pasture.
By having a low cost resource wise algorithm with good performance that can be

potentially deployed on a SBC (single board computer) and be carried and maintained by the UAV itself. I believe we are one step closer to a complete product that could come to use of a sheep farmer.
I also believe such a solution can also provide a use for the farmer during the farmers weekly checkups with the herd during the grazing season.
In the past there have been done several similar projects, but where the authors have forgotten the viability of their system to be able to run and perform well on the UAV itself under the constrain of low power and weight.

Chapter 2

Literature Review

In this chapter I present a few different master projects to show the reader what have been done in the past and to highlight the possibilities, I will also talk about existing solutions that are working and sold on the market today and to show why these products are of less popularity among farmers. I hope this will cast a light on my motivation to my light-weight project.

2.1 Earlier Master's thesis

The challenges the sheep farmer is presented with during a roundup in the fall besides monitoring his animals during the summer season is a topic that have been tackled by several prior master projects and reported on the last years: [15] [**Ytterland_2019**] [16] [17] and have been formulated by Svein-Olaf Hvasse-hovd.[8] The drone, footage and the problem space are the same for all projects.

2.1.1 Magnus Guttormsen

Guttormsen, the author of [17] developed a software system, to be used as a tool for analysis of IR and visual images. The tool convert IR picture to a matrix, and uses a clustering algorithm DBSCAN [18].

The tool allows temperature measurements on specific part of a IR picture, average temperature and height of the drone.
The tool is also support for the user to choose a specific temperature they want to see in the picture and it will display only those areas that are compliant with the chosen temperature.
The tool intended to combine infrared(IR) pictures and visual pictures together. This was intended to identify the warm area on the IR picture, then find the same area in the high resolution visual picture.
In this way the use of the IR image is to map the important spots/areas on the visual image, and save time with avoiding processing parts of the visual picture that are not of importance for the algorithm. He encounter a problem with the

drone used in the project.
The drone have two separated optics for IR and visual photography.
The visual optics have a so called "fish-eye" lens that bend the light, and therefore the objects in the visual picture is erroneously placed. This was problematic when he tried to overlap the visual and IR picture. He was unsuccessful in finding a solution in solving this challenge.
He mentions that this will be an analyser tool for a developer or a farmer, and the system is designed towards running on a screen of 24 inches and 1920x1080 resolution and was never intended to run on a drone because of limitations in resources.
The author also found that by calculating the average temperature in the IR picture they could quickly remove every temperature below the average temperature, and as a result they will get back a picture displaying only spots of higher temperature, i.e. a sheep in the picture. He believed this tool could serve as a great asset in collecting metadata for further development in solving the problem of the sheep farmer. [17]

2.1.2 Jonas Hermansen Muribø

Muribø, the author of [16] used a CNN architecture called YOLO (You Only Look Once).[**Y**]
In the last years the YOLO algorithm have been very popular as a object detection algorithm.
The performance of the algorithm is state of the art when doing object detection in real time on the COCO-image dataset. The performance is maybe not the best, but its speed, resulting in fps(frames per second) are of the best quality.[19] [20]
The algorithm first divides the input image into various grids. It then calculate the center of the cell and start working outwards from this position, the anchors are adjusted with weights, this makes the algorithm target the point of interest in the picture first. And then by performing object detection per cell of the picture. This can often be on tens or on hundred of cells.
If an object that the algorithm is looking for is within a cell it will be assigned responsibility for detecting it.
Sometimes the object consist of several cells, then it merges the cells together and when the algorithm is sure it has the whole object it will draw a bounding box around the targeted object.
As a result while training the algorithm can miss the object partially or believe two objects that are adjacent to each other of being one, and the bounding box is drawn wrongly. This is something Muribø is pointing at in his thesis in chapter 6.4.3.
He also have a hypothesis about making a sub-class of a sheep, a sheep with colour and trying to use the algorithm to find these.
As the results he receives are of lower quality then sheep in general or for that matter a white sheep compared to brown and black sheep.

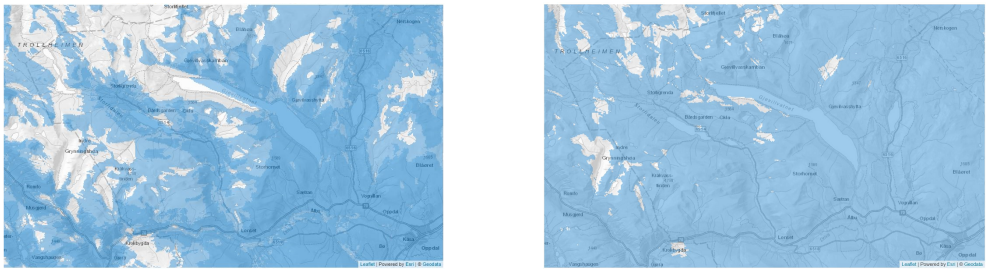
He concludes with this is probably related to that the amount of data in the dataset containing black or brown sheep are minimal and he believes this to not be of such a big issue as most Norwegian sheep are of the color white, but this can be addressed more seriously when having a better dataset in regard to brown and black sheep.
He also found that by tweaking parameter post training was changing the outcome, but the best results he received was by keeping the input pictures resized down to 832x832 pixels of the visual pictures which has the original size of 3000x4000.
Finally Muribø mentions that for future work the algorithm could be running on a SBC device made by Nvidia as its training of the model is done he believes the requirements should not be to great to run the model.

2.2 Wildlife monitoring

There are many different approaches to monitoring wildlife. Several wildlife monitoring solutions has been developed concurrently while the arise of communication coverage and standards defined for IoT(internet of things) got more practical and available for companies and "the common man".
This is only a approach that could be combined with husbandry and farming which are wildlife monitoring cases regularly occurring close to human populations and therefore available to take advantage of the cellular communication networks.

2.2.1 State of the art communication networks in Norway

Norway has a good, reliable and modern cellular communication network which is state of the art. The coverage of this network is distinguished with its population cover of 99.9% and 83.6% surface cover of the fourth generation cellular communication 4G.[13]
Even with this high coverage rate, access to this network are often prohibited for sheep farmers as the pastures are often located far away from population and in remote places.
Since the telecommunications companies are building and focusing on coverage for their customers their traditional cellular network will most likely never be a sufficient or reliable source of communication for wildlife monitoring solutions.
A solution to this is by using the NB-IoT (Narrow Band Internet of Things) or LTE MTC Cat M1(Long Term Evolution Machine Type Communications Category M1) also referred to as LTE-M.
Both are a Low Power Wide Area Networks (LPWAN), and the communication standard is standardized by the 3rd Generation Partner Ship Project (3GPP) which is a umbrella for standardization companies that are standardizing telecommunications internationally.[21]
The network is operated by telecommunication companies and focuses on low cost, power consumption and high connection intensity. By utilizing the older,



(a) 4G, 4G+ and 5G **(b)** NB-IoT and LTE-M

Figure 2.1: Figure a and b shows the differences in coverage between the two communication standard groups provided by Telenor. [22]

phased out legacy cellular communication networks(2G, 3G, 4G and soon 5G) for the IoT networks, where they are slightly modified in frequencies.
By keeping the communication standard in the sub-1GHz area the signal is able to travel much farther than traditional cellular communication standard and as a result the coverage is phenomenal. [22]
e.g. The category 1 NB-IoT network is operated by telecommunications companies like Telia and Telenor, where the focuses on low cost, power consumption and high connection intensity is a priority. [23] [24]
By reducing the frequency of the radio waves, they are able to send it further but at the cost of bandwidth, and latency. The delay can be between 1.6-10 seconds and have a 26-127 Kbit/s downlink speed and 16.9-159 Kbit/s uplink peak speed.[25] This can extend the signal from few to tens of kilometers, dependent on what kind of environment the device is placed in. [26]
According to IHS Global Insight they expect that by 2025 there will be more then 75 billion smart devices in the world, and 130 by 2030.[27] Only 10% of all IoT are connected through a cellular network. The reason is often because of the nature of application of the IoT device where the device needs to have a long battery life. By having a device connected to a cellular communication standard of the type e.g. 4G LTE, the communication standard demands that the device stays on the whole time while connected and processing different transmission request. This causes a quite large drain on the device's battery. The IoT devices are often located in industrial applications where the likelihood or the availability of charging the devices is not possible. e.g. a device measuring the temperature in a remote place. By having the device connected to the NB-IoT instead of the 4G LTE network it is demanded way less of processing transmission requests and the device can go off the network for hours or even days as seen fit. This enables IoT devices to have a extended battery life of up to 10 years depending on the application they are suited for.
The devices are additionally using the familiar Global Navigation Satellites Systems (GNSS) communication standards where they receives line of sight timed radio signals that enables the device to calculate its longitude and latitude pos-

itioning and local time with high precision. The tracking device is then communicating this data to the NB-IoT network and finally reaching the customer, the farmer or institution who are utilizing the data for their needs. Telia has their own platform, or a dashboard where the user can observe and view data regarding their IoT devices. [28]
The NB-IoT and LTE-M got a lot of its core functionality of the communication standard completed and standardized back in 2016. It is expected that the IoT over the years will increase more and more as it becomes cheaper and more accessible annually. [29]
I will now present a few solutions using this technology as of today by husbandry farmers in Norway.

2.2.2 Radiobjella

Telespor is a company who invented, produces or oversee the production(not clear) and are managing a wildlife monitoring device called Radiobjella. This device is designed to track sheep in particular but could be served for other husbandry animals.
Their device is 6.8 cm long, 5.4 cm tall and 5.2 cm wide, the device measures a weight of 104 grams with battery. It is fastened on the collar of the animal.
It uses a lithium battery of the type ER18505 on 3.6v, that hold 4 Ah (Ampere hours), this can vary from producer to producer of the batteries and the quality and charge-ability. They recommend to charge, or change batteries between every season.
The device spots a motion detection sensor, GPS sensor and a Bluetooth sensor, and is watertight. It uses the NB-IoT and LTE-M network for communication according to their website.
The device is programmed with 3 different alarms: one, the animal have not moved the last 3 hours. Two, the animal have occupied the same location for a longer period of time. Third and last, the device have not been able to report its position the last two reports, i.e indicating that the GPS is not working.
Telespor also designed the possibility of sending messages to the clients phone when these mentioned alarms are triggered. [30]

2.2.3 E-Bjeller

E-Bjeller made by FindMy is another wildlife monitoring device very similar to Telespor. The E-Bjeller differences is its battery capacity is supposedly able to last 2-3 seasons as this is dependent on the devices reporting frequency.
The user can choose from reporting every 5 minutes to every 24 hours. The device is able to detect stress within the family group, and will notify you as a user immediately.
It is not informed if the device is using a collected information from multiple devices from same family-group to determine if there is a element of stress by a exterior element.

The devices can use geo-fencing which it monitor itself where it will notify the user if it self left the fenced area. This is very useful as often the farmer can plot out the area he wants his animals to graze on over the summer.
Everything is done through a phone application and the device has recently been upgraded to version 2 which includes changeable batteries instead of being reliant on buying a so called a charging board.[31]

2.2.4 Smartbjella

Smartbjella by the company Smartbjella is again very similar to Telespor's Radiobjella and FindMy's E-Bjeller.
Smarbjella comes with temperature sensors which is unique compared to the other solutions, and provide a unique feature that they call death alarm. When the devices believes the animal is dead it will notify the client.
They are also stating that the device is going to last 1.5 years if reporting every hour or if reporting every 24 hours it will last 17 years. This is also as previously related on the batteries that comes with the device and the reporting frequency of the device.

2.2.5 Financially availability

The products, Radiobjella, E-Bjeller and Smartbjella are of similar nature, some of them are farmers that have gone together to develop a solution while others are people who have experience from shipping with tracking containers, and realizing this could also be utilized in husbandry industry.
The price of a devices are typically around a 1000 NOK a piece and an additional fee of 100-200 NOK per year per device, for subscription plan that includes a fee for the telecommunication companies who owns and maintains the NB-IoT and LTE-M network.
In Norway both Telenor and Telia are providing this service and all mentioned solutions can utilize either companies and their respective networks.
RadioBjella is almost twice as expensive as the other alternatives and there are not any obvious reasons for this to my understanding. The device is typically fastened around the sheep as a collar, and is often carried by the ewe along the traditional bell she carries.
There is not necessary to have all the animals carry the tracking device, as the hierarchy is often consistent and the sheep are of flock animal which stays in a family group of 8-10 animals. This means that by tracking the 3-4 of the older ewes in the group you would most likely have accurate data representation of the flock.
The developers behind E-Bjeller by FindMy recommend to minimum track 25% of the flock but also says that a for the best results tracking all the older sheep in the herd will result in the best results.
This is because within certain family groups when the lambs are growing up and the end of the pasturing season the ram lamb (young male sheep) tend to deviate

Table 2.1: Cost of using monitoring solutions for 156 sheep

solution	seasonal costs	buy 25%	50%	75%	100%
E-Bjeller	35 724	66 261	132 522	198 783	265 044
SmartBjella	15 840	46 264	92 528	138 791	185 055
RadioBjella	15 444	36 235	72 470	108 704	144 939

from the original family group (can occur, not a normal behaviour), to form their own family group. This is often with the assistance of one or more older sheep as they are inexperienced and need guidance.
A family group often consists half of older sheep, and the other half young lambs. [3] According to "Driftgranskningar i Jord og Skogbruk 2019" by Norsk institutt for bioøkonomi, in its example of calculation of a sheep farmers expenses and earnings of the year 2019, the farmer have a 156 winter fed sheep.[6]
As mentioned earlier the number of sheep during the grazing period of the summer, varies depending on how many successions there is in early spring but roughly the amount of animals doubles during the summer season.
This is per say not important for the tracking devices as they are carried by the older sheep, but if the amount of lambs are as many as half the herd then it is important to track more of the older sheep then if the lambs would consist of only 1/3 of the herd or 1/4 of the herd.
If there is almost the same amount of lambs as older sheep the older more experienced sheep will be spread thinner and the farmer could risk that certain family-groups are only carrying one device or even none if the farmer have a less coverage of device then 100% of the animals.
This is very sensitive as some animals are lost to natural causes and predatory attacks. Annually it is reported a loss of 10-12% animal loss due to predators attacking the sheep. [5]
If the sheep in the family-group with a tracking device would fall to pray of a predator it would result in the family group going undetected on the pasture over the season. With 156 winter fed animals, we could do an assumption that there are 312 animals on the pasture, 35 family-groups and within a family group there are 50/50 sheep and young lambs.
With 25% device coverage that will result in 1 device per family-group, With 50% 2.23 devices, with 75% 3.34 devices and with 100% 4.5 devices.
A few ewes are getting twins which in return will affect the consistency of lambs and older sheep, the family-groups will differ from family to family. This means that there are a little uncertainty in how the family groups are divided, and this advocate even more for the farmer to opt for a higher coverage of his sheep herd.

As it becomes more and more common for sheep farmers to have larger and larger sheep herds, it is difficult to see them financially be able to support themselves such a large investment. [32] As well the sheep farming industry does not have a huge margin of profit that could support such a investment. [6]

In the FindMy web-page they are informing that the farmer can apply for at the local government for economical support of buying the devices. As a result, a lot of farmers are not using these devices.
In Norway alone there are approximately 2.2 million sheep on the pasture during the summer, but the company FindMy alone have only 40 000 devices which is quite underwhelming considering the potential devices needed to track 2.2 million animals is higher.
The other companies are having different but similar size of devices sold in regard that there is clearly not very popular solution among sheep farmers across the country.

2.3 UAV usages with image recognition

A lot of species are existing in more remote locations and are monitored in the way of counting the species annually or with certain intervals to determine how the specie is thriving.
This have been done for several hundreds years, and are still done today. This involves a trained observer that travels to the destination to monitor, count and note down any abnormal behaviour of a specific species.
If location of the species are located in a challenging location to observe, usually equipment's like boats, helicopters, cars and so on are used to carry the observer to an ideal position where he or she can monitor/observe the species for a short period of time. [33] [34]
In the last 10-15 years different commercial applications including precision agriculture, surveillance, tracking, mapping and monitoring power lines, oil rigs and construction have become the domain of UAVs.
These domains have generally been done by UAVS because of the nature of higher risk in the applications.
Now these days we are seeing more and more usage of UAVS in applications where the risk aspect is not the deciding factor to choosing the UAVs but the features the UAVS can provide as they are ever so increasing in their mobility, low cost, payload and airborne time.
We are also seeing a new emerging domain for UAVS the last few years. Recent developments are done towards enabling UAVS working together to provide communication network and flying base stations for mobile operators to meet their always increasing communication demand. [11] [12] [10]

2.3.1 Jarrod C. Hodgson and co authors

The authors; Hodgson, Jarrod C and Baylis, Shane M and Mott, Rowan and Herrod, Ashley and Clarke, Rohan H of "Precision wildlife monitoring using unmanned aerial vehicles" are talking about how the data acquired by human observer and a UAV would yield different data that cant be compared directly without understanding a little more of the nature of the data collected on a UAV compared to a

human observer.[35]
In this article they are performing a count of different bird species in tropical and polar environment. Thereafter they are comparing the data between the two methods.
What they found was that UAV observation are consistently having smaller variance between the observations compared to ground based observations.
The UAVS precision does not imply estimate accuracy, as you would never be able to count every single member of a species. They are estimating based on multiple observation and density of that observation. The accuracy of UAV observations would most likely increase chances of finding trends in the the population data, that earlier was not so likely to see.
It was also discovered that UAV observation are significantly larger then ground observations, (i.e. counting more animals) they argue this is because of the topology of the observation done as the drone is looking top down, while the ground observer needs to deal with the topology of the land. Which results in animals can overlap each other and prohibit the observer from counting correct.
The UAV observations tended to have few less duplicate counts compared to ground observation.
The authors are finally concluding that UAVS as observer can, if utilized correctly, improve the wildlife monitoring process. The UAVS footage was merged together to create a larger picture witch represented the whole school of birds observed in this case.
The picture then was presented on a computer screen to a counter, who sometimes was not a bird specialist, but received a additional high resolution close up picture of the species in question.

2.3.2 Sean Ward and co authors

The authors; Ward, Sean and Hensler, Jordon and Alsalam, Bilal and Gonzalez, Luis Felipe are using UAV type 3DR IRIS, autopilot (Pixhawk) with thermal camera (FLIR Lepton) a SBC (Raspberry Pi 2) and a GPS(3DR brand) module to track and predict a path of a dog (test subject).
The on-board computer Raspberry Pi 2 receives a image captured from the FLIR Lepton camera, the camera takes several pictures every second. The images are processed for wildlife detection by using a detection algorithm from the computer vision library OpenCV on the Raspberry Pi 2 and at the same time coordinates from the GPS 3DR brand sensor.
The algorithm convert the picture to a greyscale 60x80 matrix, where each cell represent a color in the original thermal picture, ranging from 1-255.
If the algorithm believes it have detected a animal, the original picture is stamped with GPS coordinates and saved on the device.
The algorithm will then identify the correct pixel coordinate of the detected animal in the picture and send it to the Ground Control Station (GCS). At GCS the algorithm knows the angle of the camera, 22°, and will calculate the actual GPS

position of the animal in the picture with references to the UAVS GPS position, trigonometry and the pixel position the animal have in the picture.
This will then be displayed in a map. The UAV will uses it built in autopilot that will follow a arbitrary path further and upon detecting another animal it would repeat the same process and will lead to an "animal detected" stamp on the map.[36]
The limitations of the this project is the UAV is only flying 10 meter above the target animal, a dog, as well as the example pictures in the article are showing a dog on a grassy open area.
It does not state if this was actually the conditions of the experiment location, but if so the scenario is very far from representative for a wildlife detection scenario where species would live in most likely in a way more remote, occupied and noisy landscape and not a human maintained grassy lawn which is very flat, and consistent.

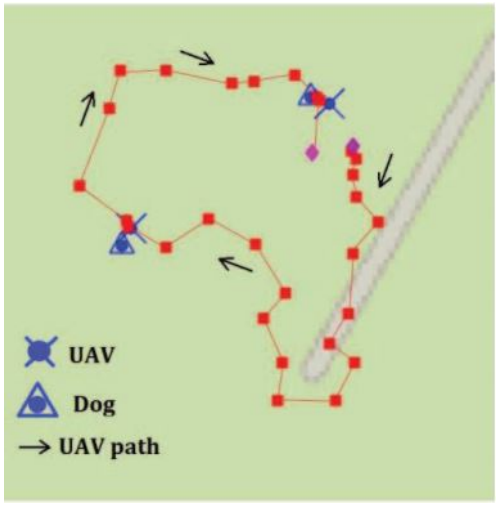


Figure 2.2: Wildlife detection, UAV's and GPS positions on a map

Chapter 3

Theory

In this chapter we will be talking about different reinforcement learning architectures with emphasises of a the artificial neural network (ANN) and its extension into a Convolution Neural Network (CNN).
CNN is a architecture that is currently the state of the art in image detection and most existing solutions or research done in this field have strongly taken premises from this architecture with modifications according to their believes or nature of task.
The main source of this chapter comes from the book: "Artificial Intelligence: A modern Approach" written by Russel, Stuart and Norvig, Peter with co authors. [37]

3.1 Perceptron

A perceptron is a neuron design inspired by the biological neurons which we can find in many animals including oneself, and original introduced and invented by Frank Rosenblatt. [38]
The perceptron receives a several numeric inputs ranging from 0 and 1, sum up every input multiplied with a corresponding weight. Then the value is combined with a unique value for the perceptron called bias. The bias serves as noise to avoid overfitting. You could say noise will prepare the algorithm more to a real world problem as the real world are full of "random" things occurring and the algorithm need to be able to differentiate between important information and noise in its assessments. Next the value is past over to the activation function.
Commonly a step function is used in a perceptron. This means that the value calculated needs to meet a certain threshold before "the signal" is passed on. This means that if the value after calculations is .e.g. higher then 1 the value is given as a output, otherwise output zero.

The activation function can either have a soft or a hard threshold. This means the threshold can be either a range of a value (soft) or a exact value (hard).
By having a soft threshold you are allowing more information pass through the

perceptron, and you have more data to calculate on at the expenses of computation and accuracy. Ultimately the soft threshold is enabling a lot of noise. Which again might result in difficult or impossible for the perceptron to infer in a good validation. (e.g. a soft threshold might end up with a not linear separable problem).
Hard threshold is opposite, it can remove a lot of noise, but also if used eagerly can remove a lot of valuable information/data. [37] chapter 18.6.3 figure 18.15 illustrates this phenomena. When training a perceptron the weights are adjusted according to the error margin of output compared to target value. If a greater error, the greater the weights are adjusted. This results that the perceptron is trying to minimize the differences between the output and and the target value. Compared to other learning algorithms that try to minimize a loss function. e.g. error squared.

Binary classification functions that can be separated with a hyperplane that are solve-able with a single perceptron. On the other hand the problem space that is not linear separable is not possible to solve on a single perceptron. [39]

3.2 Artificial Neural Network

If a perceptron is part of a network and if the network is sufficiently wide and deep enough, the network can learn a not separable problem space, multi dimension problem space, then it is called an artificial neural network (ANN).

An ANN consist of a collection of neurons connected together by the directed links. A typical network consist of several layers of neurons both in width and depth. A neural or a node consist of the same components as a perceptron: one or several input links, bias, weights, input function, activation function, output and output-links.
The input links are either coming from previous existing neurons or input values e.g. number representing a pixel value of a picture currently analysed. Input function summarize the weights and the input values, this are then passed to the activation function. Where the function will determine if the node will "activate" and pass the signal along its output-links to the neighbouring node. This can differ as there exist as mentioned earlier hard and soft threshold, it also exist step functions and so on. It depends on the chosen activation function that have been chosen.
Certain functions are more fitting to the nature of the task. A common activation function is Sigmoid. The network have multiple outputs and therefore are returning a vector, the target value will also be a vector, this is different to perceptron which returns a scalar. Figure 3.1 is illustrating a single neural node.

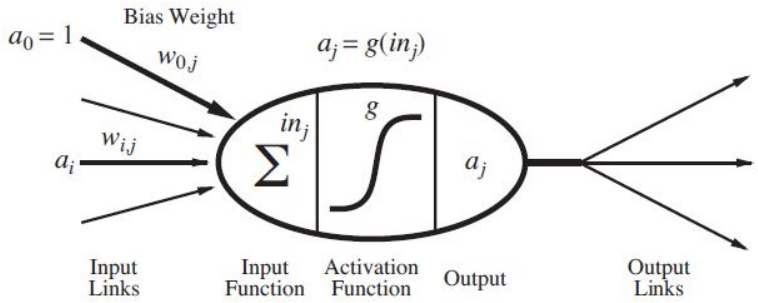


Figure 3.1: Illustration of how a Artificial Neuron works

3.2.1 Hidden units

Hidden units are referring to the ANN's deepness. I.e. how many layer there are in the network after the initial layer. Since these layers are after the initial layer, we can't directly observe what these layers are receiving of information without calculate step by step to the particular node we are interested in. Therefore the nodes, or layers are perceived as hidden because there is no way of observing their calculations, while the algorithm is running. A network is considered deep learning the moment it have more then one hidden row of nodes.

3.2.2 Backpropagation

Weights are being adjusted to infer with the importance of the current link they are associated with. If the output of the node is incorrect to the target value, the weights are adjusted. This is the same as a perceptron. The differences is that in a artificial neural network the nodes are most of the time sending the output as an input to the next node. The only nodes that does not have a parent node are the first layer, and the only nodes that do not have children are the output layer. Compared to the perceptron where input are actually raw inputs and outputs are compared to target value. In order to calculate the weights across the network the network needs to calculate backwards from the output, were output and target value are compared. This is called backpropegation and can be done with stochastic gradient descent.
It is important to understand that nodes on the same layer, are affecting its neighbouring node on the same layer. This happens especially in fully connected networks where each nodes are connected to all its children. This means that the whole previous layer is the parents of current node, but also the neighbouring node. Therefore the output of the node will affect its neighbouring nodes. One epoch on a ANN is a full iteration through the whole network and a fully backpropegation with adjusting all weights in the network.

3.2.3 Network shape

There exists several different neural network designs. The designs varies in depth, length and connections between nodes.
A typically network is a feed-forward network. It is a network where all connections are only going in the same direction. Often illustrated from left to right. The network is always receiving new inputs from parent nodes and pass it on to children nodes. Because of this the network will eventually be representative function of its current inputs. Where it will not have an internal state more then its weight. On the opposite of the scale you have recurrent network which is feeding its outputs in as inputs in its own network. Then instead of looking for the errors in the output, the activation of nodes in the network will at some point reach a stable state system wide. Or oscillation and even chaotic behaviour. It is important to look for a pattern and stop the network from continue learning.
Often the data is perhaps not so easy to decipher for us humans of what is the best for the neural network to receive and train on. Therefore there are a lot of experimenting with different activation functions, and different network designs which allows the information to be processed differently. A red line will eventually appear or a pattern will evolve. The most important parts are to have data that are diverse, vast and not to noisy.

3.3 Convolution Neural Network

Convolution Neural Network are ideal for imagery task. Images hold very large number of data when processed when each pixel represent RGB and there exist billions of them in a picture. It often ends up with the networks becoming very taxed and slow when processing such a large input.
What convolution neural network do is to retain the information at the same time shrink it down. This is done primarily through two different type of layers. There exist several more advanced techniques that are more sensitive to the unique situation the network are presented in. I will mentioned a more general applications, while keep in mind the more advanced specialized techniques are for more specific problem spaces. The general and specialized applications have the same goal of reducing the input size to increase efficiency and performance of the network.

3.3.1 Convolution Layer

The convolution layer is inspired by the neurons found in the biological visual cortex in humans. The neurons are not able to absorb more then a small fraction of information of the combined visual field. Thus each neuron is covering a small area of the visual field. Each area is overlapping the neighbouring area. This causes several areas to register some of the same information. By doing so it ensures that not only one area is picking up/analyzing a specific spot but ensuring that minimum two evaluations have been done on the same spot.

By applying a filter to each area in turn, the machine is able to extract information based on the filter. E.g. some filters are looking for a specific shape of pixels like a straight line, round shaped, s-shaped and so on. When a filter have been applied to the whole picture, the a representative feature map is created where the results from the filtering have been mapped to a same size picture. This is then repeated for the remaining filters. It is not uncommon that there are hundreds of filters.

3.3.2 Pooling Layer

The next stage is pooling layer. Here the feature maps are processed further. In the same fashion as convolution layer the machine works with a smaller section of the image one at the time.
Typically the size of the section is predetermined but are often of either 2x2 or 4x4 pixels. In this stage there are two ways of processing the section image, max pooling and average pooling. In max pooling the machine takes the highest value within the 2x2 area of pixels. This value will then be representative of this region, and be mapped to a picture of 1/4(dependent on the size of the section) of the original input picture. Thus the pictures size have shrunk, but retained the information without losing to much information. The average pooling will do the same except instead of taking the maximum value it will find the local average of the section. [40]

Both the convolution and pooling layers serves as a way of decreasing the size of input but retaining the information as much as possible. As an picture of fairly small size 100x100 pixels will in a straight feed forward ANN have over 10 000 weights for each neuron in the second layer. This illustrates the importance of having techniques of minimizing the data being fed into a ANN. By applying a "tiling layer" or pooling layer of size 5x5 to the picture will results in only 400 parameters.
Often the values in the matrix representing the input are stored as 32 bits float value. Often the number is not spread evenly over the bits, i.e they are not filling up 32 bits, but if the number is of the length 0.2157634 it only covers 1/4 of its assigned bit space in memory. Thus many algorithms are doing a Quantization. This means that they are choosing a datatype like INT8 or INT16 to represent the value, and effectively cut the memory in half or more when between layers when the data values in the memory needs to be loaded into the processing unit, which is one of the heavier steps in ANN processing steps.**??**

3.4 Transfer Learning

Since deep learning and Machine Learning are heavily reliant on massive amount of data and computational power a lot of models are trained prior to being released, this is referred to as transfer learning.
What is done is that the whole model which are sometimes several hundreds lay-

ers are trained on a huge data set like ImageNet over a longer period of time. After sufficient training and testing that the algorithm is performing well, the network is saved and published. This is done because the complexity of the algorithm and the amount of data to train it is so vast that anyone except large institutions and companies will have the possibility to train the algorithm.
It is also a problem with vanishing gradients, when the algorithm is doing back-propegation. As the last nodes are in the network are propagated fist, they then send their values backwards to the second to last layer in the network and so on until reaching input nodes.
The problem is that for each node the value to update the weights are reduced a little bit for each jump backwards.
This results that the value being diminishing small when reaching the earlier layers's weights. The weights are basically not adjusted because of the little value change.
This affect the activation function little to none and the node will activate regardless of that the value was indicating that this connection is of a less important one for the network.
As the first layers of nodes are the ones relaying the information to the rest of the network, they are of the highest importance to optimize. This is where random noise is added in as bias, dropout layer where portion of the output is disregarded and shuffling of dataset prior to running it again are done and much more.
It also exist exploding gradients where the value returned are to large, and in the similar way are sending the weights through the roof.
The transfer learning is done to enable the algorithm to normal people who perhaps don't have an access to multi million krones server.
As the model i pre-trained before being published, it is common to not touch the pre-trained stage of the algorithm, meaning the layers. Since they have been optimized on a state of the art data set and been running on massive computational power they are considered impeccable. Therefore when choosing such a model that is prep-trained it is common to add a few layers to the end of it, typically a standardized feed forward network that reduce nodes into a few nodes where each nodes represent the class in question in the problem space.

Chapter 4

The Experiment

When I started with this project I decided to have these three research questions, with emphasise the possibility to run the algorithm on a drone and relay the information regarding its findings to a database and then further on to a application with the end user, the sheep farmer. As I am only one person I agreed with the supervisor to narrow down my project to be consisting of a light weight algorithm that could potentially run on a drone. So with that in mind i made these three research questions.

- **RQ1:** How well do a lightweight classification algorithm to identify lost sheep on UAV footage perform?

- **RQ2:** Will the performance change by filtering footage on quality and diversity prior to training at the loss of quantity of the dataset?

- **RQ3:** Can the lightweight classification algorithm potentially be operating on the drone with a low power consumption hardware? e.g., System-on-Module (SoM) like Coral's Dev Boards, Nvidia's Jetsons or a newer mobile phones?

4.1 Data Acquisition and Analysis

In this project the choice of drone fell to a smaller electronic drone of an operation time of approximately 20 minutes. This was done because of simplicity and accessibility.
The ideal drone to intended for the this project are fix wing, with a considerable larger size and a longer run time of around 5-10 hours. But this a much more costly and more challenging drone to operate as it would demand a runway to operate or a ramp that can shoot it into aerial velocity.
So for data gathering and relevance to the project a simpler drone was chosen. The a drone used to gather data in this project is a DJ Mavic 2 Enterprise Dual. The drone runs with two optics, one for RGB images of 4000x3000 pixels, there-

fore its product name dual. The other is a FLIR(Forward Looking Infrared) lens which takes photos of the size of 640x480 pixels, both images are taken in a 4:3 ratio.
For each visual picture taken it exists a IR equal part, taken at the same time. [41]
Further on in this thesis I will be referring sometimes to the RGB pictures as visual images, as the RGB pictures are capturing the visible light spectrum humans are able to see while the infrared(IR) pictures are capturing wavelengths that are longer and outside of this spectrum, but representing it with shorter visible light so we can see output of the camera.[42] [43]

The data used was images from several different excursion in the span of 2018-2020. The drone have been operated by quite few different personnel, where the goal have been to collect representative data regarding sheep herding and sheep on the pasture.
Because of lack of guidelines when gathering the data, the pictures have quite few noticeable differences and quality differences. This causes the data to have quite a lot of noise and irregularities.
One of them is the height from the surface. The pictures are taken at several locations with varying height of 20 meters to 120 meters. This is something Ytterland and Winsnes are explaining that this occurred because of the varying landscape, and that they choose to stand on a hill when operating the drone which caused it to be difficult to
There is a correlation between height of the drone and the relevance of the picture. When the drone is to far away from the target(sheep) it will eventually only occupy a few pixels. This causes it to be a very challenging for the system to identify as their is only a few relevant pixels among millions.
Also the pictures provide very little data on how a target(sheep) looks like as they only occupy a few pixels and differences between a white rock, a patch of snow left from the winter or a sheep are minimal.
Muribø found in his research that the IR camera had a upper limit of 86.78 meter before IR camera was not able to detect animals, and 97.2 meter for the visual camera.
If it should become necessary to fly higher then 120 meters over ground, then there is a whole process of qualifications that need to be met by the operating pilot of the UAV or drone. This involves a examination of a higher level and notifying the CAA(Civil Aviation Authority) about the details surrounding that specific flight, which will need to occur for every specific flight.
Also the CAA have yet to allow autonomous driving of drones, or flying drones without the pilot line of sight of the drone. This was discussed in more detail by Muribø in his thesis.[16]

The drone is taking only pictures of type IR and Visual every few seconds, which is stored on a portable device. The data is retrieved post flight for analysis and predictions.

The first reason for taking pictures compared to video is that video recording are of 1920x1080 pixels while a pictures are of 4000x3000 pixels, this means that the picture is able to capture much more details then the video. This is important to prior mentioning that if two different objects are observed to far away they almost become identical because of lack of information i.e pixels. This will mean that if the video is used as material for image detection the drone needs to fly closer to the target(sheep), which will affect the effectiveness of the drone.
The second reason for opting for images instead of video is the nature of the challenge. We want to locate sheep with a drone, the sheep tend to not move to fast in a normal situation. So there is no need to capture the animals 30 frames per second as it will just end up having a typically several hundreds of frames or more of a sheep not moving. This will then tax the system for storage and processing power of unnecessary frames, which are per definition pictures.
The third reason for choosing images over video is the time perspective of this solution. While the drone is far away on the pasture, and identifies a sheep which is missing it is not needed for the farmer to have a video recording or highly precise a few second delay real time identification of a animal.
This is because it would most likely take the farmer himself to get to the sheep up in the highlands quite some time. This will be something that differs between location to location as some highlands are having regular roads, touristic roads, service roads or the farmer have made his own road that could be utilized and sometimes the farmer will need to walk on foot.
However this is not a big issue as the sheep tends to not move at high pace. This means that the farmer can arrive to the destination of the identified sheep hours later and the sheep will be, in normal conditions in it surroundings. The sheep will most likely be a family-group which has bells that make sound or a single individual that are lost which again could have a bell, if not it is assumed the farmer has fairly knowledge of how to look for a sheep in small area and most likely is accompanied with a shepherd dog. [1]
Because of these reasons there have only been used pictures in this project. There have been recorded a few videos on some of these excursions, but have chosen to exclude those in this project in the process of reaching projects goal.

4.2 Dataset

A good dataset should preferable be well maintained, diverse and large. By meeting these criteria it will be the most realistic representation of a real world scenario, as the real world is large and diverse.
There is a direct relation between quality of the dataset and the performance of the system that use the dataset for training.
The more diverse the dataset is the more generalized the system's training will become. This is important when considering that the application of the system might be noisy and difficult, as often the real world applications are.
Generalization is to which extent the system is able to perform on unseen data.

While a system that is poorly generalized is referred to as over-fitted. Then the system will be performing well on training data, but when represented with unseen data it will under perform.
It is hard to obtain good and clean datasets which achieves good results. Therefore there exists datasets which have been worked on for years, even a decade, and are maintained and worked on every day.
These datasets of such a high quality are used as benchmark within the industry to first measure systems up against each other.
It also enables researchers to their research as one of the most limiting factors with computer vision, is the vast amount of data needed and good data.

An example of this is the famous dataset ImageNet which is influential within deep learning and contributed with multiple breakthroughs over the years.[44]
ImageNet have been often used for classification and contains approximately 14.2 million labeled pictures. [44]
There are several other dataset, like MS COCO(Microsoft Common Object in Context) which is used more or less for object detection, with 328 thousand pictures [45], CIFAR(Canadian Institute for Advance Research) which is a dataset often used for image generation, and consist of 60 thousand tiny 32x32 images. [46]
All the mentioned datasets have in common that it is being used for computer vision and its many architectures.
The mentioned datasets are not locked to one specific application or usage. It is just that the MS COCO dataset is better object detection dataset instead of ImageNet because of the previous work done to adding additional metadata of bounding boxes and keypoints.
While it is possible to run a object detection algorithm on ImageNet you will need to do more preliminary work before training the system.
Certain research or research groups make their own datasets for the project, but this is often very time consuming and often a limitations in computer vision as the data needed is vast.
By constantly having public available datasets the research community can all mature the effort of data gathering.
When it comes to benchmarks it also puts the systems on a leveled plainfield where comparisons can be measured to better understand computer vision systems. It is not common that a original designer of a architecture or a system is the one who creates the best solutions or application for it.
These datasets also enables researchers to have access to good clean data which otherwise would be time consuming or nearly impossible to come by. e.g. if creating a computer vision system for detecting differences between lemon, orange, apple, melon, eggfruit and mango you would probably be wise to have at least hundreds times 6 classes, of pictures with labels and so on.

4.3 Location

The pictures have primarily been gathered in Storlidalen, Oppdal in several different excursion where each excursion have involved several sessions and some excursions have lasted over two days.
The excursion to Storlidalen 21-22.08.2019 is the excursion that is the most substantial contributor of images.
The images have been gathering around 'Storli Gard' and other locations in the valley.
The sheep that have been recorded are in some of the pictures free range sheep within a fenced area and some outside fenced area.
The goal of the excursion has been to collecting images of sheep in representative surroundings of a typical pasture which sheep could be present in during the summer season. The locations are shown in figure 4.1.
There are several other locations that have been used where the terrain have been different.
One is a field, another is around a cabin.
A field is maybe not representative surrounding of a sheep on the pasture, however this is not so important as it would contribute to generalize the system better.
The other pictures have the terrain of highland, road, cabin and fenced grass field.

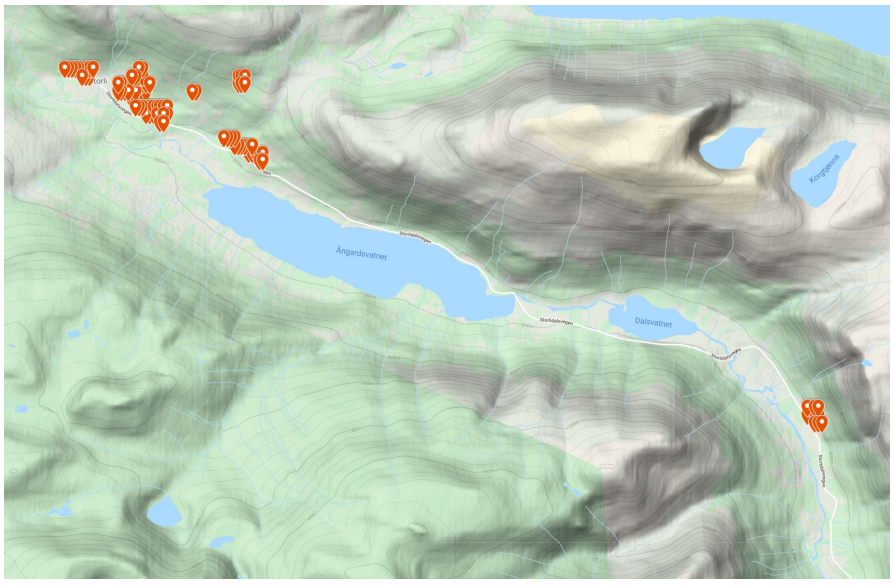


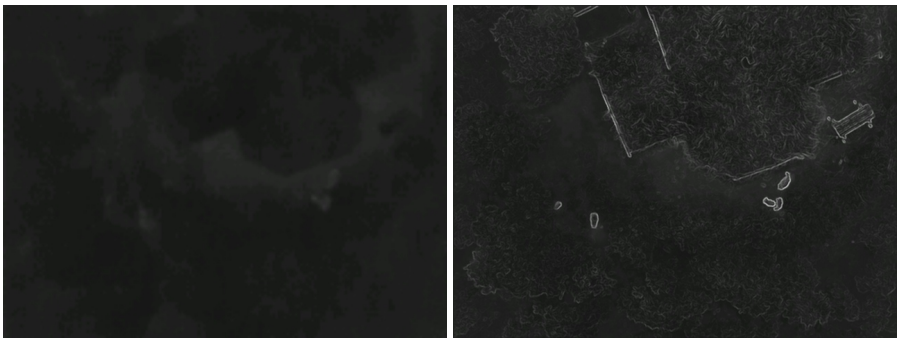
Figure 4.1: Map showing the location of the drone during data gathering in Storlidalen, Oppdal in the period of 21.08.2019-25.10.2019.

4.4 Preprocessing

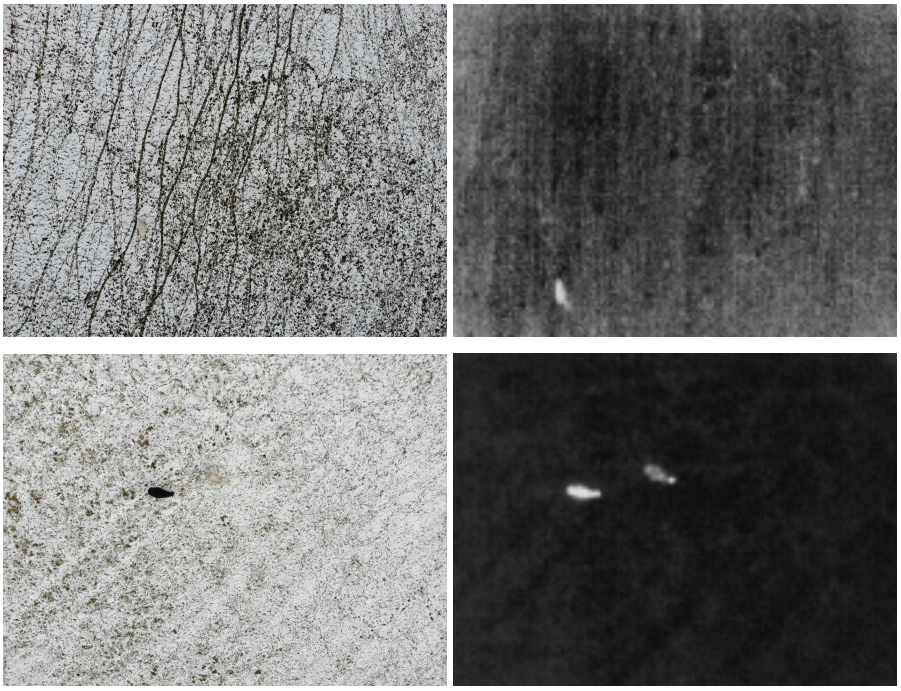
The pictures differ in quality when it comes to the IR pictures, as they are harder to calibrate, and people who have used the drone in the past have had little knowledge to how to operate the drone and adjust the IR camera and learn on the go. This is shown in Figure 4.2a where the first picture is of a blurry nature and the second one is of a sharp one.
This is related to the on board edge detection algorithm that works with the IR camera.
It is a little uncertain to why this happens but the general pattern that could be found by viewing the IR images is that different people with different knowledge have been using the drone with mixed results, as this is not the only occurrences. In figure 4.2b you can see that the drone deals very different when presented in a winter environment.
These pictures are taken on 25.10.2019 on the same day in two different sessions.

The data was provided by Magnus Guttormsen, he was a previous student and have written a thesis on the project. The data are located on Microsoft OneDrive under his name, and I was granted an access upon starting this project with a invitation link from my supervisor, Svein-Olaf Hvasshovd. The folder contains both documented and not documented images captured by different students over the years.
I choose to go through everything as a my system would benefit greatly from more images I was able to obtain.
The pictures where sorted in a few cases but this was still raw images from excursions and the labeling needed to be done.
The way I did labeling was to open each individual picture to identify if it existed a sheep in it or not.
As I have a binary classification algorithm I only needed to have positive(with) and negative(without) pictures.
So I needed to split them into *sau* and *not_sau*. This proved to be quite a task as the amount of total pictures of visual I ended up including in my dataset are 3 516 and IR pictures was similar number.
I realized quickly that the IR pictures was providing a lot of unnecessary noise, as some noise is good but when larger part of the IR pictures are distorted and blurry this can make it more challenging for the training sequence of the algorithm.
So the after sorting the IR pictures a second time I ended up with 2 728 IR images, and 3 516 visual images.

A pattern seen very strongly in the dataset I had gathered, was that very many of the pictures are very similar.
Certain scenarios the drone would just fly up 10-30 meter higher while taking 50 pictures of the same animal.
This is making the dataset to be very very little diverse.
The total of 3 516 visual and 2 728 IR pictures might sound a lot, but because of



(a) IR camera have difficukties with edge detection algorithm on images taken right after each other. Images were captured 21.08.2019 Storlidalen, Oppdal.



(b) IR camera having challenges with seemingly similar snowy surroundings. Images were captured 25.10.2019 in Storlidalen, Oppdal.

Figure 4.2

this the dataset is not the best.

To approach a solution to this I used the program [47] which is a simple tool that uses fuzzy logic to remove duplicates, where it is possible to remove also pictures that are to a certain degree similar.
I performed this removing of similarities, but took into account the second sorting of IR pictures and if I had blurry pictures or not included.
I ended up with a dataset with several sub-samples with a different preprocessing of the images or pre-sorting if you like.
The program did 3 different levels of tolerance in similar comparison when it reached a lower tolerance the image would be deleted. The levels was following Loose, Basic and Strict where strict had the lowest tolerance of difference. This means that strict will only exclude picture which are identical or a have a few pixels values in difference, while basic and loose have a greater tolerance of difference and will exclude more pictures.
This resulted in datasets containing different amount of pictures.

- IR original and second sorting(SS)
  - IR original - 2728
- IR SS with blurry
  - IR SS with blurry removed duplicate STRICT - 1980
  - IR SS with blurry removed duplicate BASIC - 1548
  - IR SS with blurry removed duplicate LOOSE - 592
- IR SS without blurry
  - IR SS without blurry removed duplicate STRICT - 1186
  - IR SS without blurry removed duplicate BASIC - 846
  - IR SS without blurry removed duplicate LOOSE - 236
- Visual
  - Visual original - 3516
  - Visual removed duplicate STRICT - 3515
  - Visual removed duplicate BASIC - 3511
  - Visual removed duplicate LOOSE - 3504

4.5 Experiment Structure

In this experiment I choose to go with google's Tensorflow, a open source platform for machine learning applications.
The Tensorflow platform is designed to push the state of the art machine learning and make it accessible and easy to build and deploy such applications. [48]
I choose the newer sub-platform that have only been around the few recent years,

the Tensorflow-lite.
The idea with Tensorflow-lite is that the communication between mobile devices and servers was to much of a bottleneck in performance when using machine learning applications on a mobile device often referred to as on the edge or on edge device.
They decided to create a mobile device friendly version so the platform can be built on the device itself.
This was not possible earlier because of the limitations in the mobile devices processing power, but have in the last 4 years changed as cell phones, tablets, SoC's(system on a chip) and SoM(system on module) are on par with laptops and are exceeding them considering processing power per wattage, which is very relevant for remote, hard to access applications.

4.6 EfficientNets

I decided to use a CNN architecture called EfficientNets, which consist of several different CNN adoptions.
The adaptations of the efficientNet approach are scaleable where the nets adjust, where some nets are wider and deeper then the other as other are thinner and shallower. [49]
As mentioned in chapter 3.4 deeper CNN are sensitive to the vanishing gradient descents problem and the gain of accuracy are often saturated after a certain steps.
The authors behind the efficientNets made the two following remarks after about CNN's after observing and studying several different approaches to scale up CovNets(convolution networks).
Several prior research was done to improve performance of different covNets. [50] [51] where either the length or the size of input was scaled up for positive performance gains, since AlexNet's win of the ImageNet competition and its [14] breakthrough advocated for deeper network.
In 2014 ImageNet competition winner was GoogleNet [52] with 6.8M parameters 74.8 top-1 accuracy on the ImageNet, while the winner in 2017 achieved 82.7% top-1 accuracy with 145M parameters, and 84.3% in 2018 with 557M parameters.
This last models was so huge that to run it, a specialized pipeline that could distribute the work to several different accelerators across the network was used.[53]

- 1 Scaling up the networks width and depth and increase resolution will result in improved accuracy, but diminishing accuracy for larger models.
- 2 To seek increased accuracy and efficiency it is important to harmonize all dimensions of a network during a covNet scaling operation.

The authors found that following compound scaling method which scales the network width, depth and resolution in a uniformly method. This will avoid the network to become to large, and to slow for its use case and therefore be mostly balanced for its task.

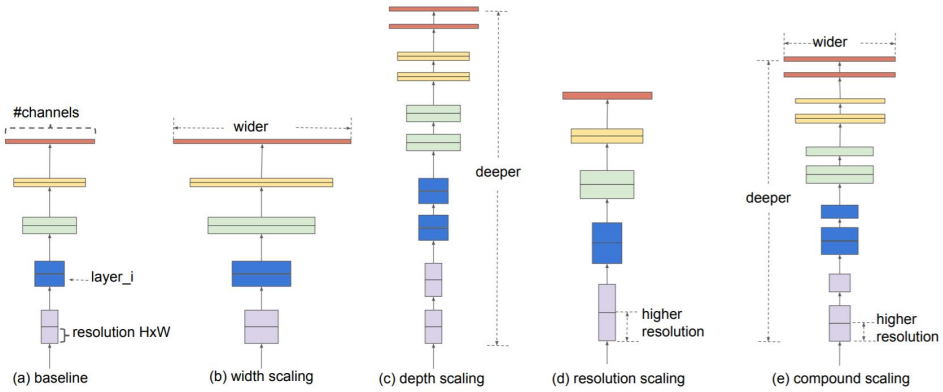


Figure 4.3: (a) Represent a baseline network, (b)-(d) are the more classical method for scaling and (e) is the authors suggestions as an better alternative. [49]

The method implies that doubling the network depth will double the FLOPS[1] and doubling width or resolution will increase it 4 times.

$$\begin{aligned} \text{depth: } & d = \alpha^{\phi} \\ \text{width: } & w = \beta^{\phi} \\ \text{resolution: } & r = \gamma^{\phi} \\ \text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ & \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned}$$

Figure 4.4: This method ensure the scaling of the CoveNet is balanced, where ϕ is a user specific coefficient, that is based on the choice of the user will scale up the network by 2^{ϕ}.

The balance is between model accuracy and model size trade-off as the larger it gets the more accurate but at the cost of being a large less efficient network.
The efficientNet baseline model is to create a set of layers of the multi-objective neural architecture search. Where each layer focuses on accuracy and keeping the FLOPS low.
This is very similar to a previous defined network called MnasNet. [54] Where in

[1] FLOPS(Float Operations Per second) are how many operations a machine is able to process per second. It can be considered as number to determined performance of a piece of hardware or in this context how much hardware is needed to run the network. When the CoveNets grows gradually they demand more FLOPS to be running. As systems and networks have a very different approaches and a lot of details, FLOPS is a very precise measurement of costly it is to run the network. Similar to gas per 10 kilometers, kilo watt per hour and so on, that are used to measure cost of running a device, without going into details of how it was constructed.

MnasNet they are targeting latency and accuracy, while in a efficientNet target is accuracy and total FLOPS.
Meaning the network has a max cost to process, while MnasNet only look for latency but at the cost in hardware/power.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 4.5: The table shows the different layers in the baseline network. Stage i with L layer, F operations, H,W resolution, C output channels.

The authors approach the base model with following approach. set ϕ to be equal 1.
When scaling we assume that resources available are double then the previous step. And then do a small grid search of α, β, γ based on equations.
They then found that for the base model, EfficientNet-B0 are $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$ under the constraint of $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$.
The α, β, γ is not re-evaluated on each step which would be beneficial but because it is quite computational expensive it is only done in the first layer.
This ensures efficient scaling at the start of the network.
The authors also found that their network out performed other CNN networks on 5 out of 8 well known datasets with an average of 9.6 times smaller network measured in millions of parameters.

Chapter 5

Results

I chose Python as the script language as I am comfortable in it and it is fairly straight forward with little boiler code necessary.
I also knew about Tensorflow platform and knew it was highly regarded as a good platform to build, train and deploy deep learning networks on.
I chose to train the model on NTNU HPCG(High Performance Computing Group) [55] in the belief that this was necessary for training the algorithm. This gave me access to several high performance accelerators to my disposal. With this I was able to train, test and experiment with my networks.

I ended choosing a EfficientNet-lite version to best possible mimic the possibility that this network could run on a smaller device like IoT/mobile devices and be carried by a drone.[56]
I chose the EfficientNet-lite4 network provided by the Tensorflow-lite library in the module tflite-model-maker.
This adaptation of the network consist of a width $= \beta = 1.4$, depth $= \alpha = 1.8$, resolution $= \gamma = 300x300$ and a dropout $= 0.3$
This a quite larger model than the base model witch would have been: width $= \beta = 1.0$, depth $= \alpha = 1.0$, resolution $= \gamma = 224x224$ and a dropout $= 0.2$.
Dropout is applied to several layers to counter overfitting, the dropout does is that it drops out random nodes in the network, thus preventing the network for memorizing the training data as this is often more likely to happen with larger networks.
It also makes up a lot of noise to prevent the network from repeating a pattern.
I ended up adjusting the network for dropout rate to 0.5, learning rate to 0.002 (which is quite high) this was to avoid getting stuck in a local maximum.
Augmentation was applied, where each pictures are copied and rotated and flipped to generate more of a variety in the dataset.
I choose to run the network for 500 epochs as I saw it was quite common to run networks pretty far when looking for subtle changes in metrics over longer periods of time.
I split the dataset in the following way: 70% training, 15% validation and 15% for testing post training the algorithm i.e. inference of the network.

Most of my runs was similar but I will mention some differences between the runs when presenting them.

5.1 RGB results

RGB or visual images where quite more computational heavier then IR images. As their size are many times larger then their respective sibling IR images, it takes quite a lot of resources in processing and when training the network.
I was so fortunate that I had access to NTNU IDUN HPCG and used a GPU as well as a CPU components of a professional level and scale.
This enabled me to run the training for 500 epochs just under 30 hours on a single node with 10 cores CPU and a GPU with 16GB memory.

5.1.1 RGB results

I choose to train the whole model or all the layers resulting adjusting all the weights in the network.
This is not smart considering the model is trained on very good quality dataset prior to me adjusting weights, but since my problem space are of a binary problem, two classes of not_sau or sau. This is not to complex as it is only two classes that need to be identified.
This proved to be true as the **??** shows that the accuracy during training was very high, meaning the network needed to adjust very little for each epochs in its back-propegation step.
The model shows that in the prior of about 150 epochs the network starts to get overfitted, as the training accuracy is surpassing the validating accuracy.
On epoch 150 the network achieved train-accuracy = 0.9803, train-loss = 0.2304 and validation-accuracy = 0.9785.
Where is a slight indication of overfitting as the training values are higher then validation accuracy at this stage. As the graph also shows, this becomes more and more subtle later on.
After that the network was trained for 500 epochs with train-accuracy = 0.99, train-loss=0.2128 and validation-accuracy=0.9785, validation-loss = 0.2435.
This indicate that the network was trained to long and should have put in even more methods for avoiding overfitting even though the dropout rate was more then doubled then original design of the network.
Finally after training the inference is done on data that have never been processed by the network, thus being the most realistic performance of the network. The accuracy = 0.9697 and loss = 0.2549. This is very good results!
The inference was having a dataset of 525 pictures that was inferred in 29 seconds total and 18.1 picture per second. This is quite fast and is forecasting a possibility of running the model on IoT or mobile devices.

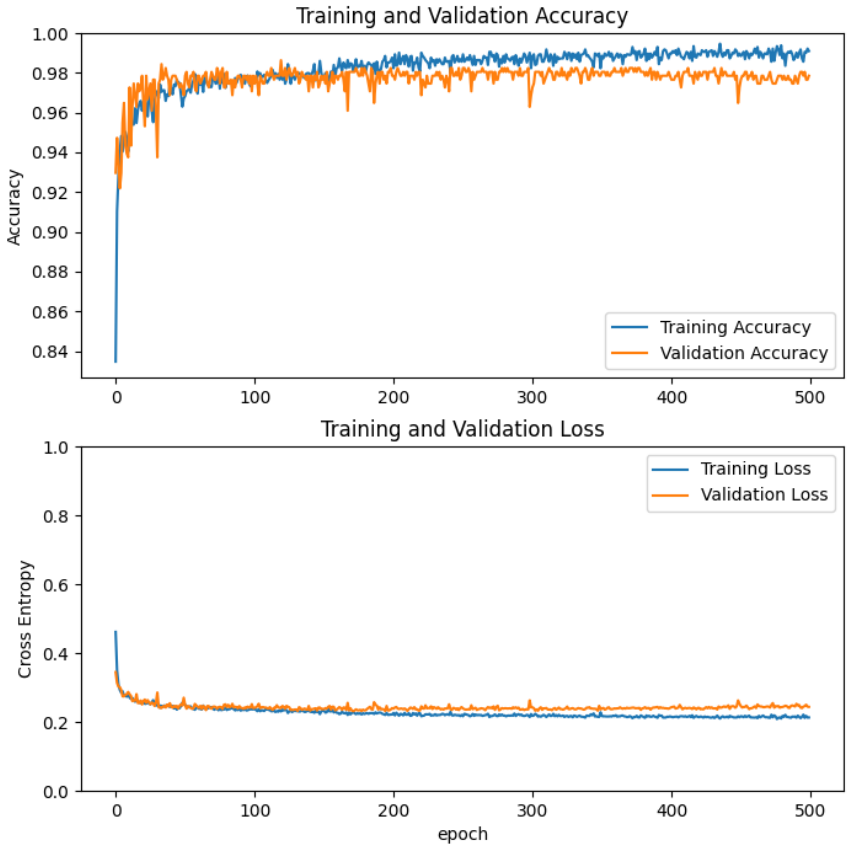


Figure 5.1: The upper graph shows the accuracy done during training and validation between each epoch, the higher value the better. Lower graph shows the loss function where lower value is better. There trend shows sing of overfitting at epoch 150 and further as both accuracy and loss graph shows training and validation going further apart.

5.1.2 RGB Results: Transfer Learning

In this run I choose to take advantage of the transfer learning.
I also changed the numbers of epochs to 200 as I believed there is fewer parameters to optimize and therefore it is not necessary to train the network for excessive amount of epochs. The trend in the graph display a clear point, it is flat, there is

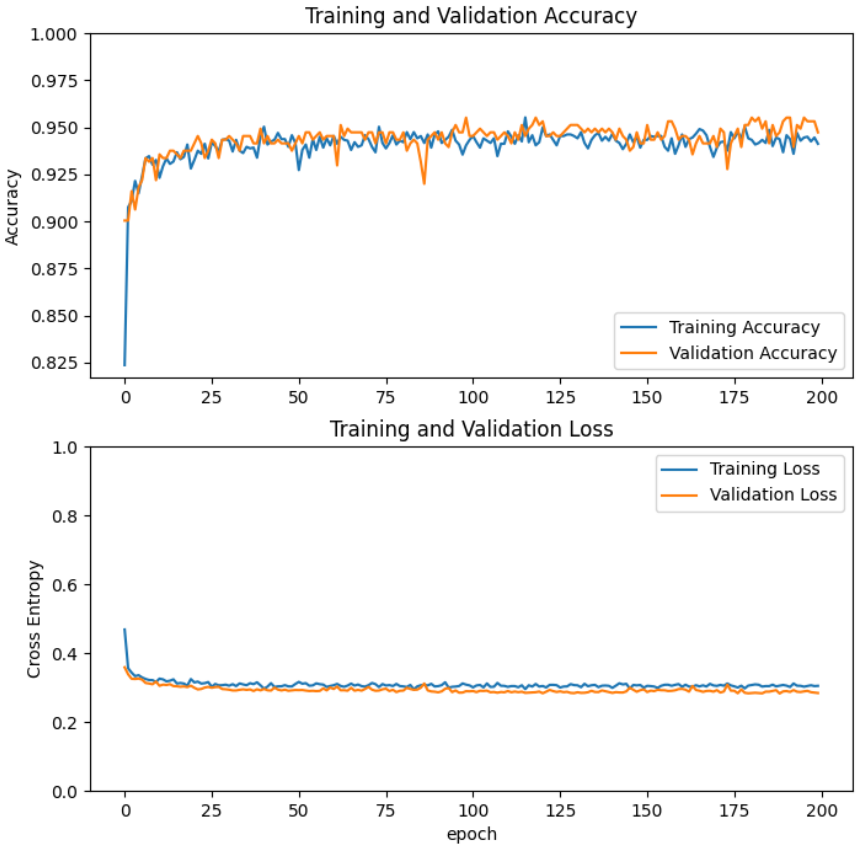


Figure 5.2: The upper graph shows the accuracy done during training and validation between each epoch. The evaluation is done between every epoch. Lower graph shows the loss function where lower value is better.

not necessarily to train the network past 25 epochs as the improvements are at most shallow and not improving across the additional 175 epochs.
The graph show no tendency to overfitting but it is interesting to see that the loss in network are quite higher then in figure 5.1 so there are clearly advantages in training the whole network in this situation.
This can also be seen on that in figure 5.1 on epoch 150 is where the loss was the

at its best prior to overfitting.
While on figure 5.2 the network is already on its best loss value at epoch 25, but this value is almost 0.10 points higher.
It is interesting to see the accuracy being in 0.95 low values across the board.
This proves the concept the authors behind the efficientNet architecture.
In this training mostly of the network was disabled for adjusting, which means the part of the network that is trained is quite small and therefore the lower accuracy and higher loss value.
While deactivating the transfer-learning the model is performing much better but at the trade-off in cost of computation and time.

5.1.3 RGB Results: Transfer Learning 25 epochs

In this training we can really see the benefit of it being a lite weight architecture as the network after just 25 epochs are 0.04 points off the network that was adjusting all its weights for 500 epochs.
The training time really show how efficientNet is efficient, with an accuracy = 0.9715 and loss = 0.2669 on the test_set or the inference of the network prior to training it.
The inference was also run on different hardware with a peak performance of 10 Terra FLOPS achieved in a inference time of 69 seconds on 525 pictures, 7.6 pictures a second was evaluated and classified.

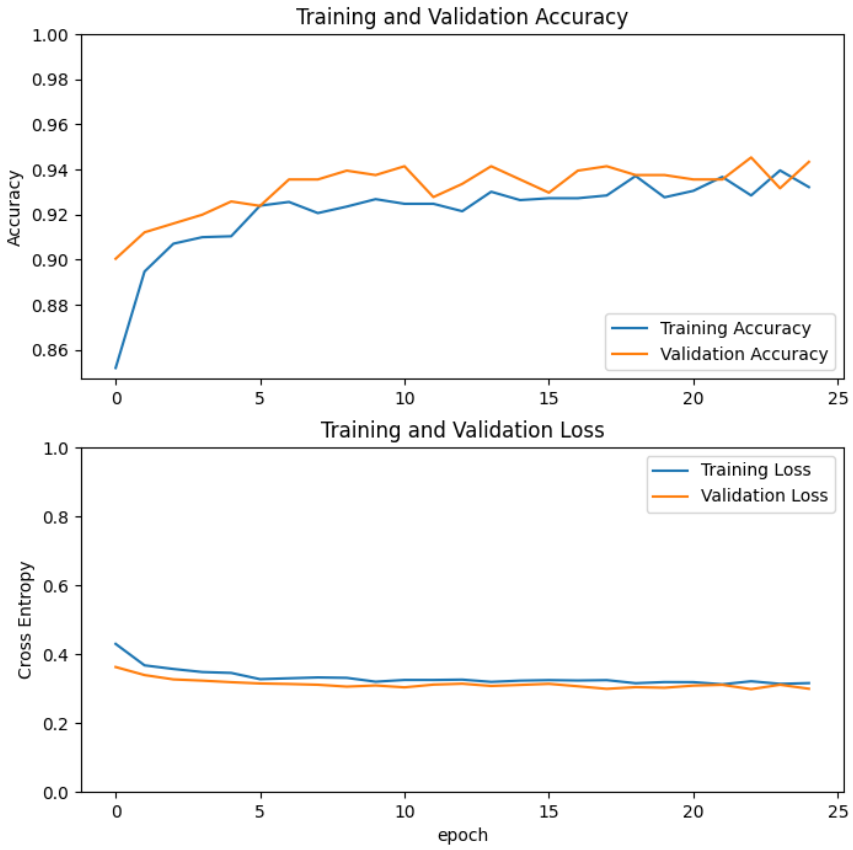


Figure 5.3: The upper graph shows the accuracy done during training and validation between each epoch. The evaluation is done between every epoch. Lower graph shows the loss function where lower value is better.

5.2 IR results

As the decision of filtering data prior to running the algorithm to look for performance change according to RQ3 in chapter6. This resulted in a few different challenges as the IR version of the dataset was restricted quite a lot, due to the manual and automatic preprocessing conducted. Some of these challenges are shown in 4.2a and the problem of quite many of the images was duplicate or to close in similarity, and was sorted with sorting tool VisiPics. [47] This resulted in pictures that are not exactly the same but similar to be excluded out of the dataset. This affected the dataset, and they ended up being drastically different in size, between which preprocessing was performed on it. See figure 4.4.
All the different networks are trained with transfer learning deactivated.
The results are reflecting some of the problems in the dataset that I will mention in the conclusion chapter. What can bee seen immediately in the graph differences is that the smaller datasets of IR are easily having problems as a result that the dataset is directly to small and lack diversity.

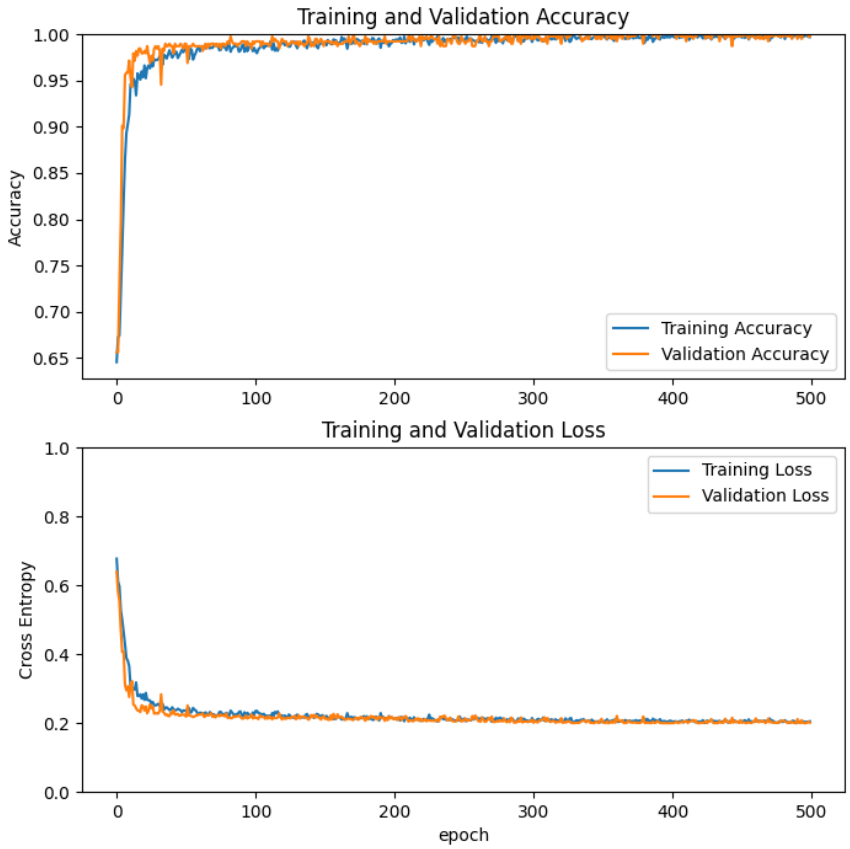


Figure 5.4: IR original Second sorting. Performance on inference on test-dataset was accuracy = 0.9927 and loss = 0.2151 on 6s, 68.2 pictures per second

5.3 Results Overview

Dataset	Epochs	Accuracy	Loss	Dataset size
RGB	500	0.9697	0.2549	3516
RGB	200	0.9521	0.3478	3516
RGB	25	0.9715	0.2669	3516
IR OR	500	0.9927	0.2151	2728
IR DR-WB-Strict	500	0.9933	0.2058	1980
IR DR-WB-Basic	500	0.9871	0.2159	1548
IR DR-WB-Loose	500	0.9551	0.3091	592
IR DR-WoB-Strict	500	0.9944	0.2049	1186
IR DR-WoB-Basic	500	1.0000	0.1997	846
IR DR-WoB-Loose	500	1.0000	0.1999	236

OR = Original, DR = Duplicate Removed, WB = with Blurry, WoB = With out Blurry

My code can be accessed at github: hakonbockman
The dataset are of 120GB and will be given access to upon request due to its size.

Chapter 6

Conclusion

The idea behind the project is to develop a full system where the farmer will be notified about sheep on the pasture in the highlands which are vast and difficult to explore and cover in case of lost or missing animals.
It is thought to mainly utilize the solution during the roundup process where the farmer is collecting all his animals from the pasture in the fall, but I would argue it could also serve as a observation tool.
The idea is also that the UAV that is going to be finally appropriate will have several meters of wing span, run on petrol and can run for hours.
The UAV will consistently take visual and thermal pictures, and process them with an image recognition system that is carried by the drone.
The UAV will be autonomous or following a predefined fly path or flying pattern.
Upon detection of a sheep the drone will log the relative position of the sheep, and transmit the sheep location to a database.
The farmer will be using his phone, which runs an application that will have access to the same database.
The phone will be able to place a mark on a sufficient map that will represent "sighting of a sheep", with a time stamp and color representing how recently is was done, which will gradually fade to other respective colors depending on the current time.

Me and my supervisor agreed on that through a image classification solution the drone would never be able to separate if there are 10 sheep or 1 sheep in a image. This means that 1 sheep or 100 sheep in a image is considered the same by the algorithm.
Because of the sheep living in a family-group of 8-10 animals, this is very likely to happen, that several animals are present in the image that is detected for sheep.
Although this dose not pose a problem for the farmer as he or she have equal interest to come and collect one animal or several animals, as the farmer is responsible for every one of his animals of the herd. [1]
The precision of the detection location was not of such importance as the farmer wont be able to mobilize himself and arrive at the location where the sheep was

detected at, before some time has passed.
By the time farmer is present at the location it is most likely that the sheep or the whole family-group have moved further on. Also a sheep is not a wandering animal but rather prefer to stay in local environments, compared to a wolf who are migrating over huge areas as part of their habitat
Therefore the classification method to this problem space is a sufficient solution as a precise location finder. But is never optimal as we can not "ask the sheep to stand still" until the farmer is arriving at site.
The drone will fly over an area in a height of 50-120 meters, as allowed according to Norwegian flight regulations [57].

Before this project started it has been several other master projects where the students have tried to solve different aspects of the problem area.
My aspect in this project was the image recognition algorithm of a nature of light weight so it could possibly run on the drone, without any assistance from a server-park.
The reason for this was that the previous image recognition algorithm proposed and proven to work have been running on large components that are not suitable for a drone to carry.
I have used my algorithm to run on a desktop myself as a hardware example suitable for the task and the drone that would be appropriate for this project. However I have measured the FLOPS to highlight the computation needed to inference on the network I am suggesting as a solution to this project.

I believe all the results show that efficientNet-lite4 provided on the Tensorflow platform are providing an excellent choice when coming to a on board chip in a drone.
The advantages with Tensorflow it could be exported to Android Operating system and the platform comes in both java and C versions.
I choose to write in Python as I knew from talks with the supervisor that getting my hands on any suitable hardware would most likely not be possible. This would also be less relevant as the hardware will change year to year and there is today not decided or planned any investment plan towards buying a fixed wing drone of larger capacity.
Also because of Civil Aviation Authority Norway the law forces the project to be flown by a certified person, and also each flight needs to be coordinated with the agency, if the a larger drone will be bought.

- **RQ1:** How well do a lightweight classification algorithm to identify lost sheep on UAV footage perform?

Very good, the EfficientNet networks show all result over 95% accuracy with a low 0.30's loss values. 5

This definitely show how well suited the architecture is for such an application.5 The figure 5.3 shows that the architecture is able to get very good results with minimal training.
Because of the small size needed to train there is a possibility of in the future to set the mobile device running the algorithm on the drone to do training while the drone is busy refueling or training.
This could be done on a daily basis to keep the algorithm optimized as the terrain could change a little depending on which areas the drone is deployed in.

- **RQ2:** Will the performance change by filtering footage on quality and diversity prior to training at the loss of quantity of the dataset?

The IR dataset lack diversity, as the pictures have less information because the pictures have a much smaller resolution and many pictures where rendered to plain noise, as the optics are quite difficult to calibrate. This caused that larger portions of the dataset to be considered less valuable.
In figure A.7 the dataset is less then 600 images and figure A.4 the dataset is lower then 300 images.
This caused the network to become quite absurd, and noise become quite visible on the training performance.
This also ended up with the validation dataset that is used to counter overfitting to become so small that it was not representative of the actual dataset.
The same could be said about the inference done on both network where A.4 network had a 100% accuracy while A.7 was plagued with a lot of noise because within its small dataset consisted quite large portion off blurry images.
This caused the results to jump all over the place compared to other networks.
There exist good small datasets of the same size as mentioned above, but their diversity are great, compared to the one of the IR dataset.

Through my preprocessing steps I realized that this is a larger fault in both IR dataset but also in the RGB dataset and is a fault in the dataset overall which affecting the performance of the classification problem.
The problem is that during the collection of images, the operator of the drone have chosen to take several 10s of pictures or even sometimes almost 100s of pictures of the same sheep standing almost still while the drone have flown upwards.
This per say not a bad decision as could be valuable data to know how far you could get away from target before a detection problem would occur.
The problem is that $\approx$ 80% of the dataset consist of repeated footage of similar situations and objects. This causes the algorithm to train over and over on very similar images in the dataset.
This is something I tried to mitigate through my preprocessing steps, but was less unfortunate with the IR datasets ending up being almost not usable.
The lack of diversity in the dataset have also made the test dataset which the network is tested on after training, consist of quite many similar pictures that where

present in the training set. Even with shuffling prior to splitting the dataset, it persisted with lesser results.
I believe the lack of diversity in the dataset is so great that this is larger the reason why the network are performing so well.
I believe that through looking at the table in chapter 5.3 it can bee seen that the results for the IR dataset are so good that this point at the lack of diversity are even stronger in the IR dataset compared to the RGB dataset.
This is also related to the RGB pictures comes naturally with more information as they are 39.0625 times larger then the IR pictures. This causes the RGB to have quite different values if the drone moved 1-2 meters between the pictures, while in a IR picture this difference would almost not make any changes to image.

- **RQ3:** Can the lightweight classification algorithm potentially be operating on the drone with a low power consumption hardware? e.g., System-on-Module (SoM) like Coral's Dev Boards, Nvidia's Jetsons or a newer mobile phones?

The network shown in figure 5.3 was running on a computer of 10 Terra FLOPS, where the graph is already quite flat. This indicate that there is not much to gain from continuing training the network for this particular application.
The inference on the test dataset was achieving 7.6 pictures per second with a accuracy of 0.9715 and loss=0.2669.
This proves the network does not need to be so substantially trained to perform quite well. I am confident a mobile device of processing power of 2-4 Terra FLOPS will be sufficient enough to do so. As it would not be necessary to process more then ≈ 1 picture per second as the drone and the sheep are not moving very rapidly.
This could change in the future, but 2-4 Terra FLOPS is a quite low computational power and can be easily increased with correct components.
The Nvidia's Jetson AGX Xavier runs on 15watt and can deliver 70 Terra OPS, Jetson Xavier NX delivers 21 OPS and a kit cost $\approx$ 1500 USD. [58] The jetson nano could be sufficient enough, but could be on the weaker side, and needs more investigation.
I would opted for something little more powerful as the drone thought to suit this project is a fixed wing which can't fly so close to the target as some of the pictures present currently in the dataset.
This means it could be interesting to have better optics, which implies more input to the network to process, thus better hardware is needed.
There is also exist Google's Coral who is boasting of providing 2 TOPS/watt which is for only 60 USD and a developer board for 100 USD. [59]

6.1 Future Work

For future work I would want to export my network to a android device. Today's mobile phones are so powerful that one of the flagship phones with a suitable arm processor would be sufficient to run the algorithm fast enough.
I tried this with my older phone with an example network in the beginning of the project with mixed results, but is totally possible.
Then the phone could have been either taped, or connected to the drone that is used temporary to collect the data for this project.
This should be possible as the DJ Mavic drone have an extra carrying capacity of 200 grams, typically a phone weighs around this much.

I believe improvements of the dataset is critical.
As I mentioned earlier in this thesis, the networks performance is often tied closely to the data given and its quality.
I believe the dataset is of lesser quality and to improve it, the students should organize to go regularly or quite many trips to Storlidalen to take more pictures, of sheep.
This is time sensitive as the most important time to take pictures is during the roundup period. Then the pictures should be sorted, labeled and stored in the most convenient way possible for re usability.
The IR sensor, or gathering data should have certain guidelines, to better calibrate IR sensor, and new operators of the drone could easier gather data that would be representative and diverse.

When it comes to the algorithm choice, I believe there exist more alternatives to EfficientNets out there that could challenge its performance. e.g. I believe it could be interesting to see how well the MnasNet [54] which efficientNet is heavily inspired from, performs.

Bibliography

[1] O. Blix Anna; Vangen. (). 'Sau i store norske leksikon på snl.no.,' [Online]. Available: https://snl.no/sau. (accessed: 30.04.2021).

[2] (). 'Beitebruk,' [Online]. Available: https://www.bondelaget.no/beitebruk/. (accessed: 24.05.2021.

[3] J. R. E. Johanssen and K. Sørheim, 'Sau-atferd og velferd hos sau,' 2018. [Online]. Available: http://orgprints.org/33947/1/NORS%5C%C3%5C%98K%5C%20FAGINFO%5C%20Nr.%5C%205%5C%202018%5C%20Atferd%5C%20og%5C%20velferd%5C%20hos%5C%20sau.pdf.

[4] F. av Klima- og miljødepartementet. (). 'Forskrift om erstatning når husdyr blir drept eller skadet av rovvilt,' [Online]. Available: https://lovdata.no/dokument/SF/forskrift/2014-05-30-677/%5C%C2%5C%A76#KAPITTEL_2-4. (accessed: 08.06.2021).

[5] N. N. Surveillance. (). 'Erstatning for sau,' [Online]. Available: https://www.rovbase.no/erstatning/sau. (accessed: 08.06.2021).

[6] B. Kristiansen, 'Driftsgranskingar i jord-og skogbruk-rekneskapsresultat 2019 account results in agriculture and forestry 2019,' *NIBIO Bok*, 2020.

[7] N. (S. og Geit). (). 'Verdisatser - norsk sau og geit,' [Online]. Available: https://www.nsg.no/a-a/okonomi/verdisatser/. (accessed: 13.5.2021.

[8] S.-O. Hvasshovd. (). 'Droner og sauog litt til !! anvendelser og muligheter,' [Online]. Available: https://www.statsforvalteren.no/contentassets/cbf122460efa4e37a051c17c07fade0d/droner-buskerud-2017.pdf. (accessed: 16.06.2021).

[9] F. Nex and F. Remondino, 'Uav for 3d mapping applications: A review,' *Applied geomatics*, vol. 6, no. 1, pp. 1–15, 2014.

[10] I. Colomina and P. M. de la Tecnologia, 'Towards a new paradigm for high-resolution low-cost photogrammetryand remote sensing,' in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS Congress, Beijing, China, XXXVII. Part B*, vol. 1, 2008, pp. 1201–1206.

[11] F. Al-Turjman, 'A novel approach for drones positioning in mission critical applications,' *Transactions on Emerging Telecommunications Technologies*, e3603, 2019.

[12] Z. Ullah, F. Al-Turjman and L. Mostarda, 'Cognition in uav-aided 5g and beyond communications: A survey,' *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 872–891, 2020.

[13] H. W. L. Amund Kvalbein and A. G. Lie, *Bredbåndsdekning 2020*. [Online]. Available: https://www.regjeringen.no/no/dokumenter/prop.-1-s-20202021/id2768453/.

[14] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'Imagenet classification with deep convolutional neural networks,' in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] E. A. Rognlien and T. Q. Tran, *Detecting location of free range sheep*, 2018.

[16] J. H. Muribø, *Locating sheep with yolov3*, 2019.

[17] M. Guttormsen, *Gjenfinning av sau ved hjelp av drone*, 2019.

[18] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, 'A density-based algorithm for discovering clusters in large spatial databases with noise.,' in *Kdd*, vol. 96, 1996, pp. 226–231.

[19] Paperswithcode.com. (). 'Real-time object detection on coco,' [Online]. Available: https://paperswithcode.com/sota/real-time-object-detection-on-coco. (accessed: 17.06.2021).

[20] C.-Y. Wang, I.-H. Yeh and H.-Y. M. Liao, 'You only learn one representation: Unified network for multiple tasks,' *arXiv preprint arXiv:2105.04206*, 2021.

[21] GSMA. (). 'Narrowband – internet of things (nb-iot),' [Online]. Available: https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/. (accessed: 29.06.2021).

[22] Telenor. (). 'Iot dekning,' [Online]. Available: https://www.telenor.no/bedrift/iot/dekning/. (accessed: 29.06.2021).

[23] GSMA. (). 'Mobile iot deployment map,' [Online]. Available: https://www.gsma.com/iot/deployment-map/. (accessed: 29.06.2021).

[24] Telia. (). 'Massiv iot - nb-iot lte-m,' [Online]. Available: https://www.telia.no/bedrift/digitalisering/iot/tilkobling-konnektivitet/massiv-iot--nb-iot--lte-m/. (accessed: 29.06.2021).

[25] Wikipedia. (). 'Narrowband iot,' [Online]. Available: https://en.wikipedia.org/wiki/Narrowband_IoT. (accessed: 29.06.2021).

[26] U. Raza, P. Kulkarni and M. Sooriyabandara, 'Low power wide area networks: An overview,' *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[27] D. K. Boccara. (). 'Digitalisation and the industrial iot revolution – why should you care?' [Online]. Available: https://ihsmarkit.com/research-analysis/digitalisation.html. (accessed: 29.06.2021).

[28] Telia. (). 'Telias iot-plattform,' [Online]. Available: https://www.telia.no/bedrift/digitalisering/iot/telias-iot-plattform/. (accessed: 29.06.2021).

[29] GSMA, '3gpp low power wide area technologies - gsma white paper,' GSMA, Tech. Rep., 2016.

[30] B. O. 7. 673. (). 'Elektronisk overvåking av husdyr,' [Online]. Available: https://telespor.no/. (accessed: 24.06.2021).

[31] FindMy. (). 'Findmy - satellite tracking technology,' [Online]. Available: https://www.findmy.no/. (accessed: 24.06.2021).

[32] SSB. (). '05985: Number of animals, by contents, domestic animals of various kinds and year,' [Online]. Available: https://www.ssb.no/en/statbank/table/05985/chartViewLine/. (accessed: 05.06.2021).

[33] K. H. Pollock, J. D. Nichols, T. R. Simons, G. L. Farnsworth, L. L. Bailey and J. R. Sauer, 'Large scale wildlife monitoring studies: Statistical methods for design and analysis,' *Environmetrics: The official journal of the International Environmetrics Society*, vol. 13, no. 2, pp. 105–119, 2002.

[34] D. D. Dolton and R. D. Rau, 'Mourning dove: Population status, 2002,' 2002.

[35] J. C. Hodgson, S. M. Baylis, R. Mott, A. Herrod and R. H. Clarke, 'Precision wildlife monitoring using unmanned aerial vehicles,' *Scientific reports*, vol. 6, no. 1, pp. 1–7, 2016.

[36] S. Ward, J. Hensler, B. Alsalam and L. F. Gonzalez, 'Autonomous uavs wildlife detection using thermal imaging, predictive navigation and computer vision,' in *2016 IEEE Aerospace Conference*, IEEE, 2016, pp. 1–8.

[37] S. Russel, P. Norvig *et al.*, *Artificial intelligence: a modern approach*. Pearson Education Limited, 2013.

[38] F. Rosenblatt, 'Of biological memory,' *Matematika*, vol. 5, no. 6, pp. 18–31, 1958.

[39] M. Minsky and S. Papert, 'An introduction to computational geometry,' *Cambridge tiass., HIT*, 1969.

[40] D. Scherer, A. Müller and S. Behnke, 'Evaluation of pooling operations in convolutional architectures for object recognition,' in *International conference on artificial neural networks*, Springer, 2010, pp. 92–101.

[41] D. website. (). 'Mavic 2 enterprise series built to empower. destined to serve.,' [Online]. Available: https://www.dji.com/no/mavic-2-enterprise. (accessed: 24.07.2021).

[42] M. P. Devices and M. A. Devices, 'Electromagnetic waves,' 2006.

[43] R. S. Berns, *Billmeyer and Saltzman's principles of color technology*. John Wiley & Sons, 2019.

[44] K. Yang, K. Qinami, L. Fei-Fei, J. Deng and O. Russakovsky, 'Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy,' in *Conference on Fairness, Accountability, and Transparency*, 2020. DOI: 10.1145/3351095.3375709.

[45] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, 'Microsoft COCO: common objects in context,' *CoRR*, vol. abs/1405.0312, 2014. arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.

[46] A. Krizhevsky, G. Hinton *et al.*, 'Learning multiple layers of features from tiny images,' 2009.

[47] J. Cristy. (). 'Welcome to visipics,' [Online]. Available: http://www.visipics.info/index.php?title=Main_Page. (accessed: 10.04.2021).

[48] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu and Xiaoqiang Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[49] M. Tan and Q. V. Le, 'Efficientnet: Rethinking model scaling for convolutional neural networks,' *CoRR*, vol. abs/1905.11946, 2019. arXiv: 1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946.

[50] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang and K. Murphy, 'Progressive neural architecture search,' in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19–34.

[51] K. He, X. Zhang, S. Ren and J. Sun, 'Deep residual learning for image recognition,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, 'Going deeper with convolutions,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[53] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu *et al.*, 'Gpipe: Efficient training of giant neural networks using pipeline parallelism,' *Advances in neural information processing systems*, vol. 32, pp. 103–112, 2019.

[54] M. Tan, B. Chen, R. Pang, V. Vasudevan and Q. V. Le, 'Mnasnet: Platform-aware neural architecture search for mobile,' *CoRR*, vol. abs/1807.11626, 2018. arXiv: 1807.11626. [Online]. Available: http://arxiv.org/abs/1807.11626.

[55] M. Själander, M. Jahre, G. Tufte and N. Reissmann, *EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure*, 2019. arXiv: 1912.05848 [cs.DC].

[56] Tensorflow-Gardener and A. Wang. (). 'Efficientnet-lite,' [Online]. Available: https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet/lite. (accessed: 20.09.2020).

[57] C. Norway. (). 'The civil aviation authority of norway's main objective is to contribute to safe civil aviation in norway.,' [Online]. Available: https://luftfartstilsynet.no/en/. (accessed: 15.07.2021).

[58] N. Corporation. (). 'A breakthrough in embedded applications,' [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/. (accessed: 20.07.2021).

[59] Coral. (). 'Products: Helping you bring local ai to applications from prototype to production,' [Online]. Available: https://coral.ai/products/. (accessed: 29.06.2021).

[60] SSB. (). '03791: Domestic animals, by contents, region, domestic animals of various kinds and year,' [Online]. Available: https://www.ssb.no/en/statbank/table/03791/chartViewLine/. (accessed: 30.04.2021).

Appendix A

Additional data f efficientNet

Figure A.1: A detailed illustration of the base-model in efficientNet architecture, it is very large vast even considering it being a small CNN. source: EfficientNet-B0

A.1 IR Images results

Placing the graphs here:

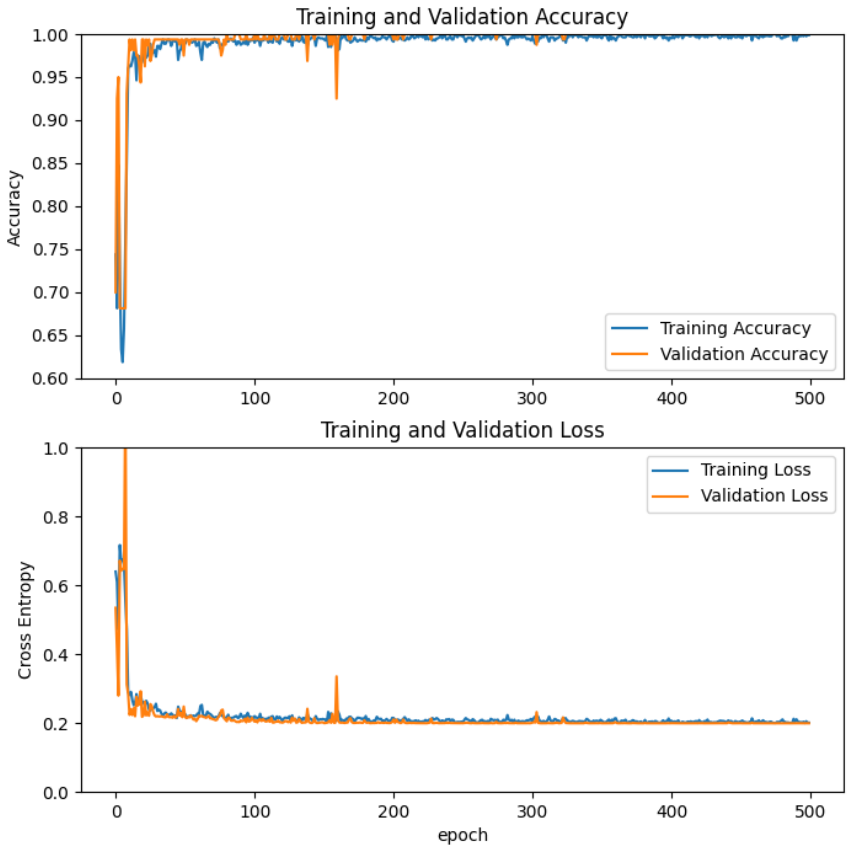


Figure A.2: IR duplicate strict. Performance on inference on test-dataset was 4s total loss = 0.2049 and accuracy = 0.9944

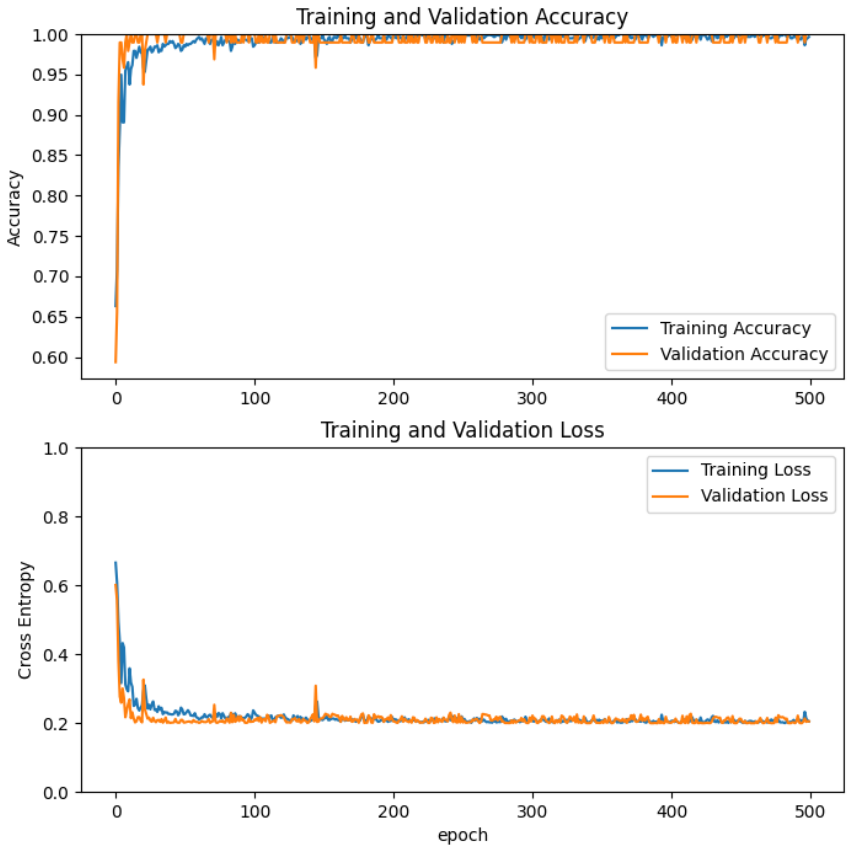


Figure A.3: IR duplicate basic. Performance on inference on test-dataset was 3s total, loss = 0.1997 and accuracy = 1.0000

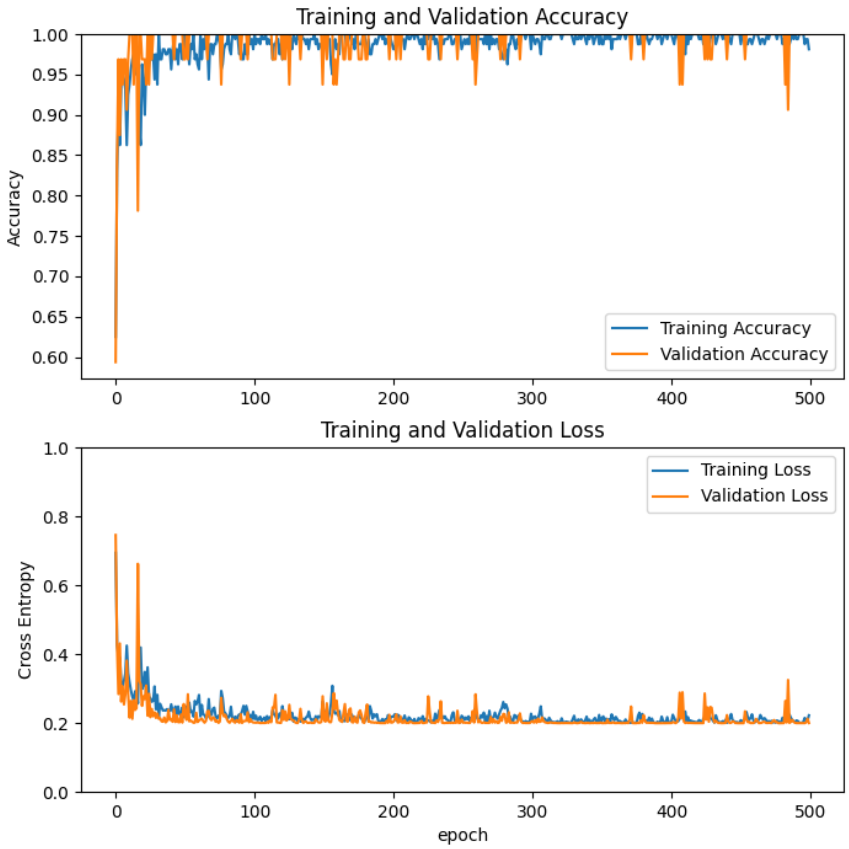


Figure A.4: IR duplicate loose. Performance on inference on test-dataset was 1s total, loss = 0.1999 and accuracy = 1.0000

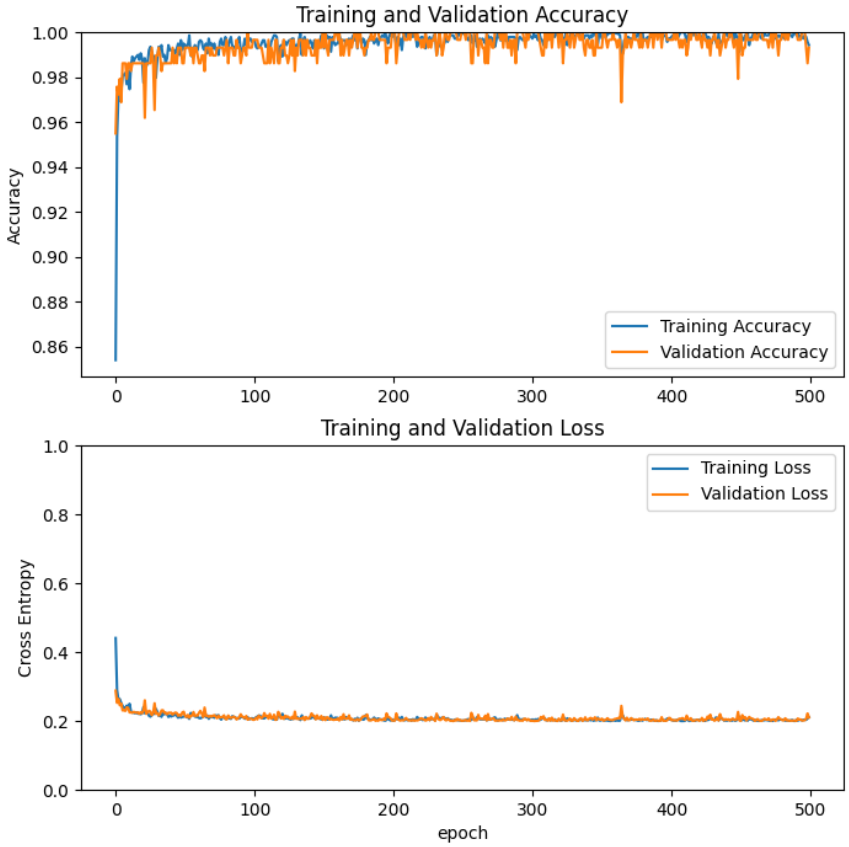


Figure A.5: IR duplicate loose. Performance on inference on test-dataset was 8s total, loss = 0.2058, accuracy = 0.9933 and epoch = 34s

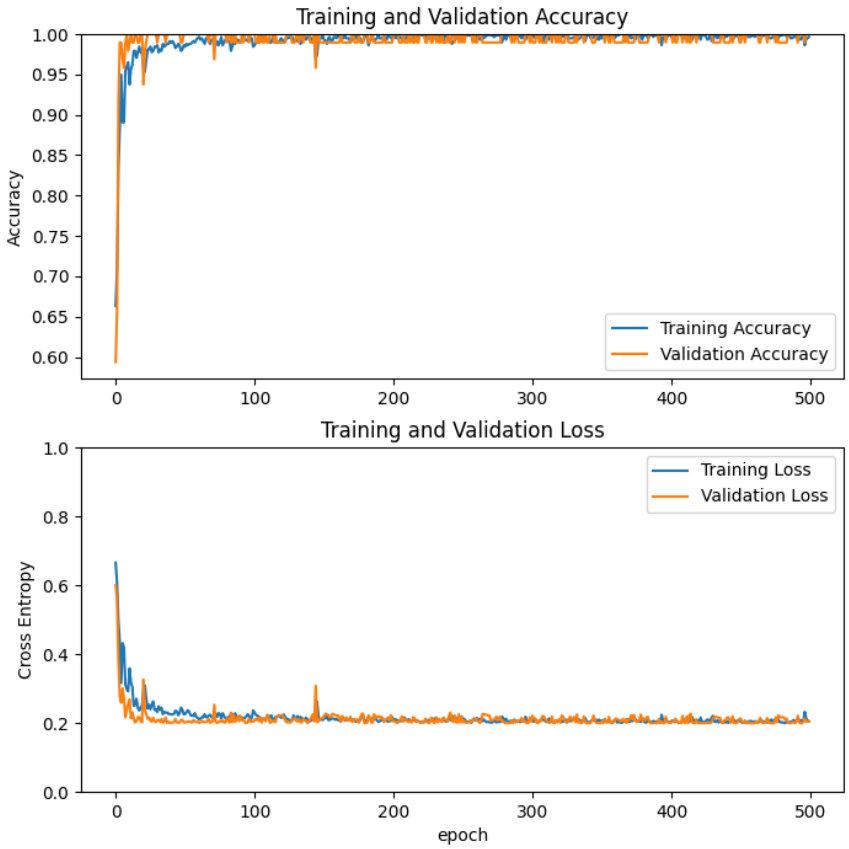


Figure A.6: IR duplicate loose. Performance on inference on test-dataset was 4s total, loss = 0.2159, accuracy = 0.9871 and epoch = 27s

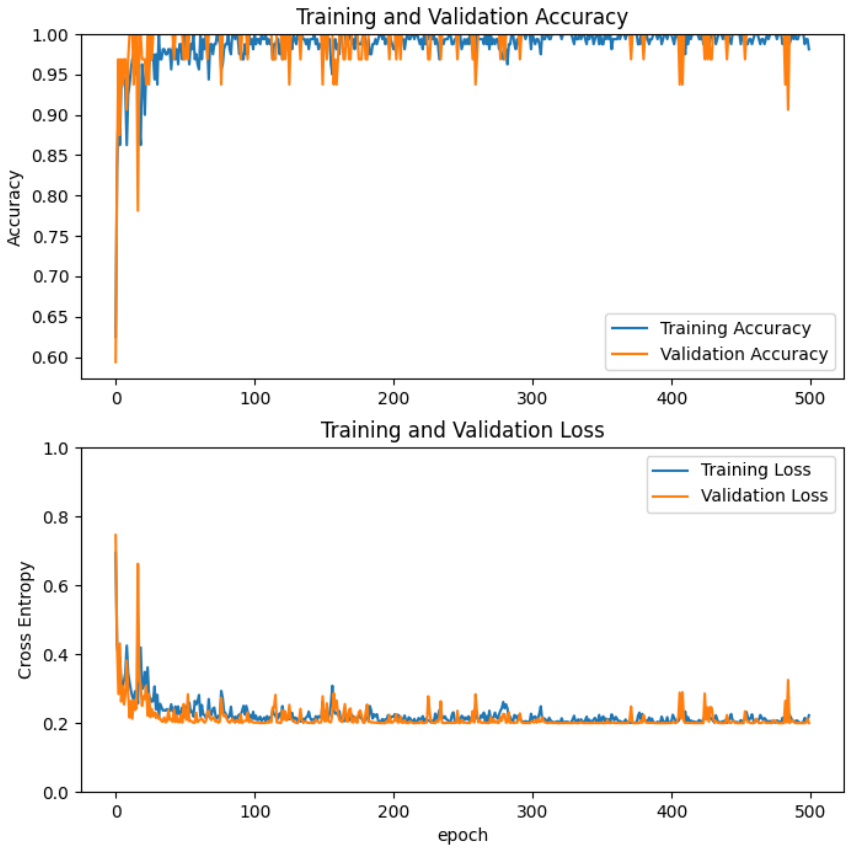


Figure A.7: IR duplicate loose. Performance on inference on test-dataset was 3s total, loss = 0.3091, accuracy = 0.9551 and epoch = 9s

Appendix B

Materials, graphs of sheep in Norway

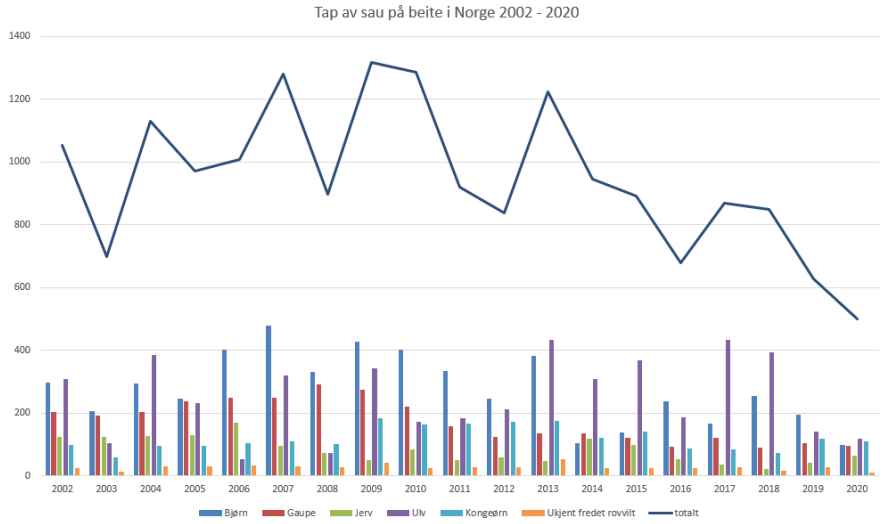


Figure B.1: Loss of sheep on the pasture 2002-2020. [5]

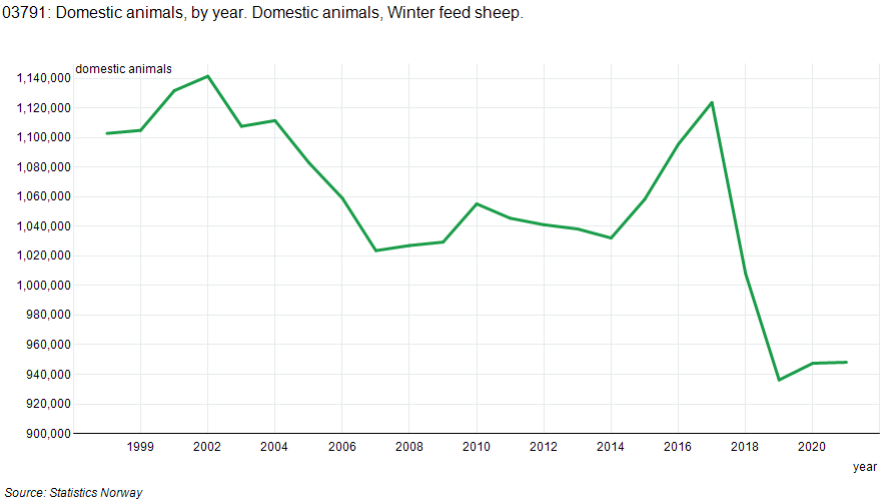


Figure B.2: Winterfed sheep by year 1999-2020. [60]

Figure B.3: Winterfed sheep per farm by year 1999-2020. [32]