

Integrating Activities for Advanced Communities



D6.3- Demonstration Report on Artificial Intelligence and Machine Learning

Project No.871120– INTERACT

H2020-INFRAIA-2019-1

Start date of project: 2020/01/01

Duration: 48 months

Due date of deliverable: 2022/02/28 (M26)

Actual Submission date: 2022/02/18

Lead partner for deliverable: AFRY

Authors: (In alphabetical order) Erman Maria, Skogsmo Markus, Sundström Carl, Örn Fredrik

| Dissemination Level | | |
|---------------------|---|---|
| PU | Public | X |
| PP | Restricted to other programme participants (including the Commission Services) | |
| RE | Restricted to a group specified by the Consortium (including the Commission Services) | |
| CO | Confidential, only for members of the Consortium (including the Commission Services) | |

Table of Contents

| | |
|---|-----------|
| Publishable Executive Summary..... | 3 |
| Table of Acronyms | 4 |
| 1. Introduction..... | 5 |
| 1.1. Background | 5 |
| 1.2. Purpose and Goals..... | 5 |
| 1.3. Tasks..... | 6 |
| 1.4. Justification of Chosen Applications | 6 |
| 2. Agenda | 8 |
| 2.1. Module 1 – Computer vision for Camera Trap Footage | 8 |
| 2.1.1. Consistent and Effective Wildlife Monitoring | 8 |
| 2.1.2. Cameras in Cairngorms..... | 8 |
| 2.1.3. Thesis Work..... | 10 |
| 2.1.4. MegaDetector – The Machine Learning Animal Detector | 10 |
| 2.1.5. Camelot – User Friendly Implementation | 11 |
| 2.1.6. Summary..... | 12 |
| 2.2. Module 2 – Crowdsourcing Computer Vision and Building Your Own Model. | 13 |
| 2.2.1. Crowdsourcing Computer Vision | 13 |
| 2.2.2. Training Your Own Model..... | 14 |
| 2.2.3. Summary..... | 16 |
| 2.3. Module 3 – AI with Satellite Data for Earth Observation..... | 17 |
| 2.3.1. Satellite Data..... | 17 |
| 2.3.2. Examples of Available Services | 17 |
| 2.3.3. Earth Observation Examples..... | 18 |
| 2.4. Module 4 – Digitizing Text with OCR and Summarization with NLP..... | 19 |
| 2.5. Module 5 – Azure Machine Learning Designer on Tabular Data | 21 |
| 2.5.1. Cloud Computing – Azure | 21 |
| 2.5.2. Azure ML Designer | 22 |
| 2.5.3. Summary..... | 23 |
| 3. Conclusions and Future Work..... | 24 |
| 4. Bibliography..... | 25 |

Publishable Executive Summary

This document compiles the topics presented during the demonstration on February 7th, 2022 by work package 6, comprising deliverable 6.3. Station managers and researchers engaged in the International Network for Terrestrial Research and Monitoring in the Arctic (INTERACT) were invited.

One of the goals of work package 6 is to raise awareness of and develop usage of artificial intelligence (AI) and machine learning (ML) to researchers and station managers. This includes demonstrating applications and areas, where these techniques excel, their constraints, as well as introducing pre-requisites needed to use them. A goal of WP6 is also indirectly to incite change by showcasing how employing AI/ML techniques does not have to be a huge undertaking, and that small efforts may greatly benefit researchers.

An aim of D6.3 was to demonstrate how ML techniques can be used on example data to showcase specific algorithms and methods, and the outcomes thereof. A wide variety of ML techniques have been used for this demonstration, such as Computer Vision (CV), Natural Language Processing (NLP) and decision trees in application areas such as classifying animals and mushrooms, using satellite data for earth observations, digitizing and summarizing logbooks, as well as predicting and classifying tabular data.

This demonstration has shown that many exciting possibilities exist for INTERACT researchers to use AI tools to analyze data and create new insights, with or without coding. These possibilities should not be overlooked and it is highly recommended for smaller research projects to start utilizing the demonstrated techniques.

Table of Acronyms

| | |
|-----|-----------------------------------|
| AI | Artificial intelligence |
| API | Application programming interface |
| CV | Computer vision |
| DL | Deep learning |
| GPS | Global positioning system |
| IoT | Internet of things |
| ML | Machine learning |
| NLP | Natural language processing |
| OCR | Optical character recognition |
| WP | Work package |

1. Introduction

1.1. Background

There has been great progress by academia in analysing and documenting climate change. However, with powerful tools such as Artificial Intelligence (AI) and Machine Learning (ML), it has now been accepted that there are many ways of knowing. Technological advances and cutting-edge research have increased the demand for using AI and ML tools to combat climate change. In addition, there are hidden resources on environmental change that can complement conventional and traditional knowledge.

Essentially, AI is about mimicking human intelligence, with ML being a set of tools, techniques and algorithms to achieve AI [1]. ML models may not only provide new techniques for aiding climate researchers efficiently in a data-driven way [2] but it also shines when it comes to automating manual and tedious work already performed by humans today. It does this by learning from sources of information and applying this knowledge to new information. As such, application areas are as diverse as image recognition, object detection, self-driving cars, optical character recognition (OCR), syntax analysis, sentiment analysis of text etcetera.

A demonstration that comprises D6.3 was arranged by AFRY and was held virtually. This demonstration was based on input and feedback gathered from D6.1, a preliminary study of inquiries and needs from station managers and researchers.

AFRY is a Swedish-Finnish engineering, consulting and design company within the fields of technology, energy, industry and infrastructure. It was founded in 1895 in Sweden and now has around 19000 employees globally, and offices in more than 40 countries [3].

1.2. Purpose and Goals

One of the many goals of INTERACT III is to develop usage of AI and ML from the invaluable data that is collected by the many dedicated researchers involved in INTERACT III. Several research stations undertaking work in the Arctic regions, already possess large volumes of data dating several decades back [4]. These data volumes have often been collected without AI or currently modern techniques in mind; however, with modern computing power and techniques, they may be of potentially great use today in different contexts.

The purpose of arranging a demonstration was to use ML techniques on example data in order to showcase specific algorithms and methods and to demonstrate the outcomes thereof. A wide variety of ML techniques have been used for this demonstration, such as Computer Vision (CV), Natural Language Processing (NLP) and decision trees in application areas such as classifying animals and mushrooms, using satellite data for earth observations, digitizing and summarizing logbooks as well as predicting and classifying tabular data.

1.3.Tasks

The aim of D6.3 is to raise awareness of ML techniques and algorithms to researchers and station managers in INTERACT III. This includes demonstrating applications and areas, where these techniques excel, their constraints, as well as introducing pre-requisites needed to use them. The aim of WP6 is also indirectly to incite change by showcasing how employing AI/ML techniques does not have to be a huge undertaking, and that small efforts may greatly benefit researchers. Another goal has been to expound on the latest AI trends and algorithms available today that can be of benefit to Arctic researchers, as well as giving a taste of what may be available in the future.

1.4.Justification of Chosen Applications

AI/ML is a large concept and has many possible applications. Given the scheduled time for the demonstration, limitations and selections had to be made. Elaboration of how selections were made can be looked into results and conclusions from previous work in WP6. The delivery of the Pre-study (D6.1), on inquiries and needs from identified station managers and researchers, laid the foundation for the demonstration (D6.3). The results of this preliminary study formed the basis of what applications and choice of AI and ML to be utilized for the demonstration.

The results of the quantitative study clearly show that the majority of the respondents have an interest in image recognition and detection, with applications ranging from satellite imagery to scanning old photos for salient information. The pre-study showed that image data were commonly brought up with regards to both camera traps, i.e., motion-sensitive cameras, where both animals and sometimes people appeared, as well as conventional cameras. These data were also used to study the phenology of plants, as well as studying the arrival and departures of various bird species. There was also interest in the usage of satellite data and synthetic aperture radar to study geological hazards, events and snow cover.

Additionally, the topic of digitizing old data was an emerging theme, and as one informant put it:

We have some old data, that would be nice if someone would have time [to digitize]. There are some projects that we know what has been done, we know the sites and the station is old enough [...] so it would be nice to compare this with the present situation. (P14)

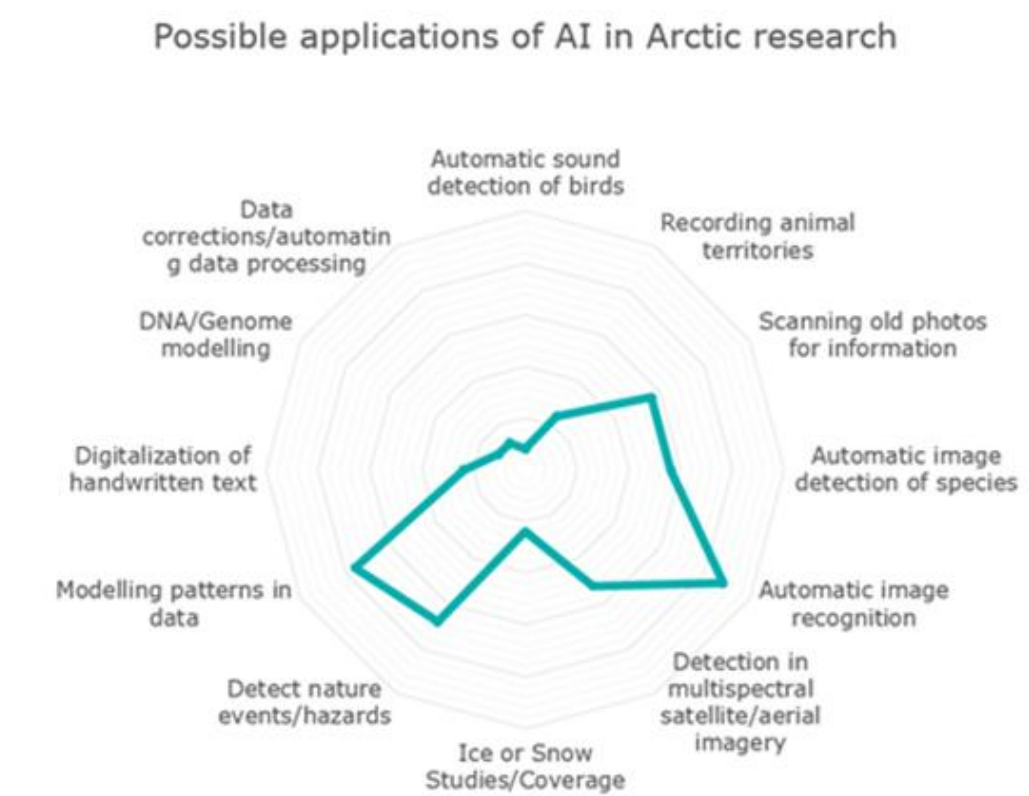


Figure 1: Possible applications of AI in Arctic research from the pre-study

In addition, AI/ML can be used in different ways. By training, a model gains experience to be able to automatically make predictions from or analysis of data. Another way to use AI/ML is to use the results that AI/ML has produced, which can be a result of what is called Feature Learning. Feature Learning can discover patterns and draw conclusions for detection or classifications from sample data, and therefore reduce manual investigations and studies. Depending on the sample data used for Feature Learning, results can be used to perform specific tasks. Both ways were demonstrated.

Regarding the choice of tool and programming languages, Python is currently the most popular programming language used for AI/ML. The main reason for this is that many Python libraries are easy to use, while still being powerful. However, not everyone can, or have the time to learn how to program Python. One of the observations from the pre-study was that it was stressed that a way to utilize AI/ML without having to program would be preferred. One researcher also mentioned that not everyone had enough experience in mathematics to grasp AI, while others stated that they in general were unsure of how to approach AI practically. For that reason, the demonstration was focused on presenting tools that would require as little programming skills as possible. Nevertheless, AI/ML is a field that today generally needs some programming skills.

2. Agenda

The digital demonstration was in total three hours long and was divided into five different modules, each demonstrating how different AI applications could be utilized by INTERACT researchers. During the presentation the participants were invited to engage and interact using the tool Mentimeter. The whole session was recorded and is available on INTERACT's YouTube channel[5].

2.1.Module 1 – Computer vision for Camera Trap Footage

Monitoring the fauna in an area can provide important insights into the health of the ecosystem, which is why it is included in the "INTERACT Minimum Monitoring Programme", described in INTERACT's *Research and Monitoring*, from 2015. However, such monitoring can be time-consuming and require skilled staff to do a large amount of field work. An interesting way to counter this is to employ the use of camera traps combined with Computer Vision. This approach was covered in the first module of the demonstration[6].

2.1.1.Consistent and Effective Wildlife Monitoring

The camera traps are automatically triggered cameras mounted in the habitats of species that researchers want to monitor. As cameras have become better and cheaper, the method has rapidly increased in popularity among ecologists as a way to sample populations. Another key strength is that the method is very consistent. Even if the camera traps are imperfect and miss a few animals, researchers will get data continuously, as opposed to sporadic field observations. Historical use of camera traps and the evolution of the field is well-covered in *Camera Traps in Animal Ecology* [7].

Regarding the best way to set up a camera trap survey and what factors to consider when assessing the results are covered in the review article *REVIEW: Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes* [8], which is highly recommended for INTERACT stations that want to set up a bigger survey.

The main drawback with large camera trap surveys is that they generate a large number of images that need to be processed by staff before they can be turned into valuable insights. Camera traps also tend to produce a lot of blank images, in which no animal can be seen. In one of the largest and most well-studied public camera trap datasets, *Snapshot Serengeti*, more than 70 % of the images are blanks [9]. A good way to counter this is to train a machine learning model to recognize images containing animals so that they can be filtered out. Researchers can then spend all their time on images of interest. For surveys with enough data, it is also possible to train the model to identify different species, and even individuals.

2.1.2.Cameras in Cairngorms

At least one of the stations in the INTERACT network, ECN (Environmental Change Network) Cairngorms in Scotland, is already using camera traps in their research [10]. Their cameras have been in use since 2010 and

has mainly been used to study the relationship between human activity and wildlife [11]. Some examples of images from different locations used in the survey can be seen in Figure 2.

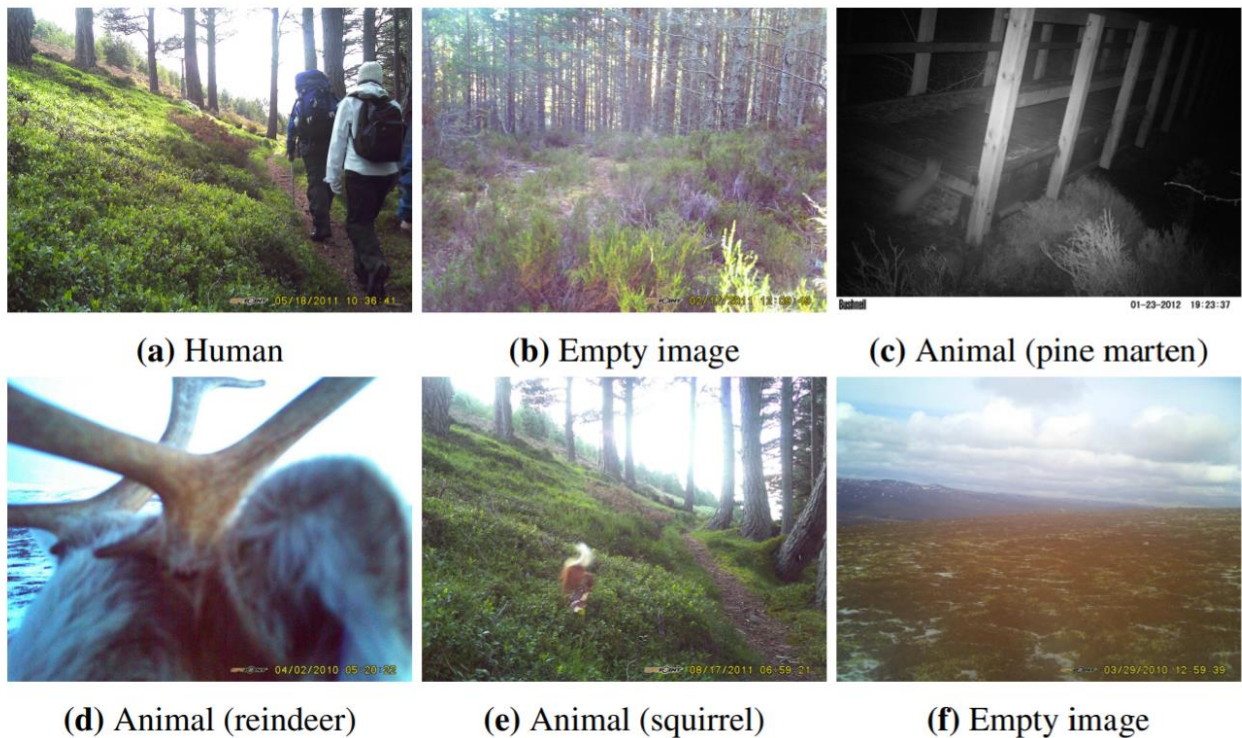


Figure 2: Photos from the Cairngorms data set.

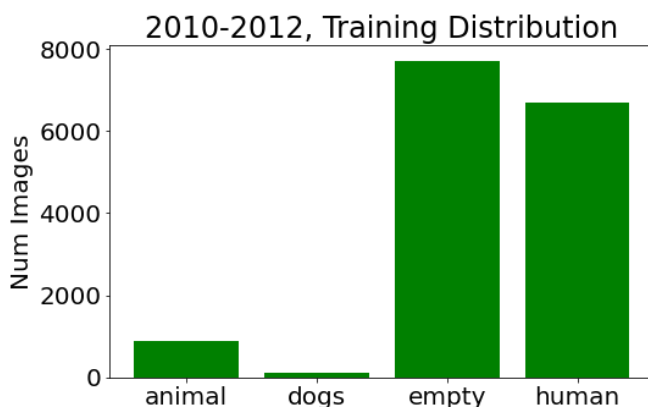


Figure 3: Data distribution in Cairngorms data set.

The Cairngorms survey has had at most nine different camera trap locations. Just as for many other camera trap data sets, more than 50% of the images are empty, as seen in Figure 3. Only about 6 % of the images featured wild animals, the rest were images where human activity had triggered the cameras. This is due to the fact that the national park is popular among tourists.

Regarding camera hardware, ECN uses automatic cameras of two different brands, Bushnell and Spypoint [12]. Camera traps of these brands have price ranges of roughly \$100-\$300. However, the models currently on the market have better resolution than the older ones used in Cairngorms.

2.1.3.Thesis Work

As part of AFRY's work package, a master thesis project was conducted on the subject *Computer Vision for Camera Trap Footage* [13], since the pre-study had found it to be one of the most promising possible applications of machine learning within INTERACT. The project used the images from ECN Cairngorms and focused on comparing two different Computer Vision approaches, object detection and classification. During the project, an object detector using the YOLOv3 model was trained and achieved decent performance [14]. However, a pre-trained model named MegaDetector, developed by Microsoft's AI for Earth team achieved better performance. Combining the detections of animals and humans with the timestamps of the photos also allowed to filter out images with dogs.

The thesis project could also find some weaknesses in the model, e.g., its tendency to detect other man-made, but static structures, as vehicles. Since the detections of this structure are static, recurring false positives of this type can be filtered out.

2.1.4.MegaDetector – The Machine Learning Animal Detector

MegaDetector is the main part of AI for Earth's ongoing camera trap project, which can be followed through its Github page. MegaDetector has been trained on millions of camera trap images from different locations over the earth, which has made it efficient at detecting animals and humans in a variety of settings. In their proposed workflow, MegaDetector is the first step that filters out blanks and leaves you with bounding boxes for all images that are left that describe what part(s) of the image that contains a detection and if it is considered an animal, a human or a vehicle, the three detection categories [15]. A typical detection output can be seen in Figure 4.

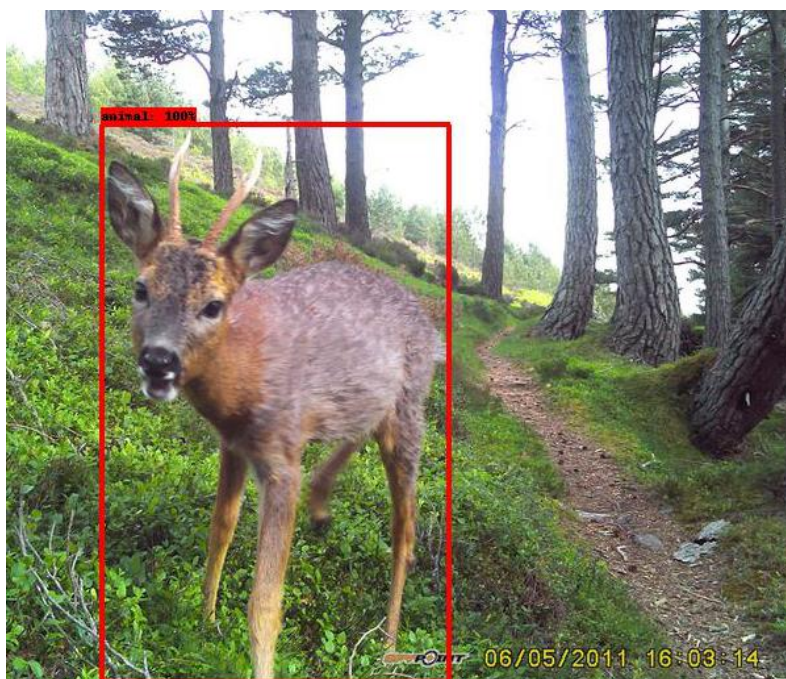


Figure 4: An animal detection on a picture from the Treeline camera in Cairngorms.

To use MegaDetector, there are a few different ways to get started. To get a quick view of the model in action, it is recommended to take a look at Google Colab notebooks [16]. This shows code, albeit only requires the user to run pre-written code with clear instructions. However, it is not a method suited for bigger projects, since Colab times out if longer sessions are run - which is required if there are more than a few hundred images.

Another alternative is to run MegaDetector locally, i.e., on your own computer. This enables further possibilities with an output, such as cropping images to bounding boxes and training a species classifier on specific data. However, it also requires more programming knowledge and preferably a GPU, to process all images at a sufficient pace. A possible way around that for big data sets with 100,000+ images is to get in contact with the AI for earth team and get assistance with processing via the cloud.

The last alternative, and the recommendation from the AFRY team is the application Camelot [17].

2.1.5.Camelot – User Friendly Implementation

Camelot is a tool developed for managing camera trap data and it has built-in support for MegaDetector. It is free and developed by two volunteers. It allows users to load images into the application and with a rather simple interface to go through images with many different filtering options. If detection is enabled, the images will be uploaded Azure Blob Storage and then be processed by MegaDetector via Microsoft's cloud API, all for free. Depending on how many other users are uploading images at the same time, the time periods for getting the detection results can vary, but it's typically from a few minutes up to a few hours. The application doesn't have to be running once the images have been uploaded, so it will be free for other work until the results are back. Another thing that can possibly limit the detection speed is of course the Internet connection, since all images have to be uploaded before to Azure.

Once the images of interest have been filtered out and any additional information needed has been added directly via the UI (e.g., species, age, sex) it is easy to generate different reports in csv format. All information can be exported directly and then be processed in Excel, but there are also pre-built reports that cover e.g. a specific site or a species. Another perk with Camelot is that it is well documented and has an active forum where other users and the two developers help out with questions and issues.

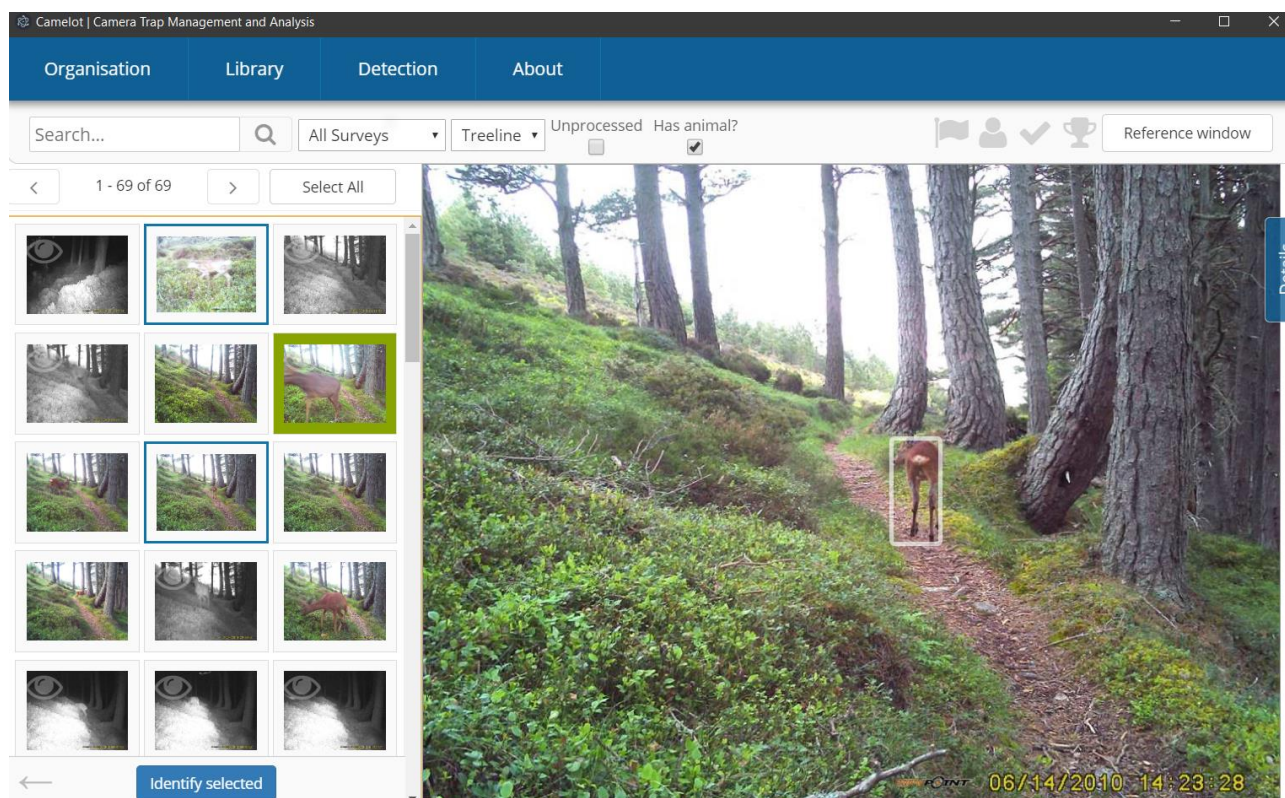


Figure 5: The Camelot UI. Here, two filters are applied: a specific camera location (Treeline) and images with animal detections.

Even though Camelot is a great tool, it is important to be aware of potential weaknesses. Since the tool is free, it is brittle with respect to service continuity. The developers are not obligated to continue providing the service. It also relies on grants to be able to provide the detection service, which requires a fair bit of cloud computing. That should not stop scientists from using the service, but it is good not to become dependent on it.

One should also be wary of uploading any potentially privacy-sensitive images (pictures of people) if detection is activated. The images will be uploaded to an Azure blob storage where they can be accessed by the two developers. It will automatically be deleted after detection is completed or after 60 days, whichever comes first. Make sure that this is in line with local laws and regulations before uploading such footage.

2.1.6.Summary

To summarize, using camera trap surveys combined with Computer Vision models, can give ecologists a great tool to improve wildlife monitoring on their stations. It is by no means perfect, but can give consistent and cost-effective results without a complicated setup. The AFRY team recommend interested stations to try it out by purchasing a single camera trap and then processing the images with the MegaDetector model.

2.2.Module 2 – Crowdsourcing Computer Vision and Building Your Own Model.

Identifying and understanding patterns in data is an integral part of science and engineering, and it has been for a long time. For example, Carl von Linné found patterns and characteristics in living organisms. He classified and categorized them into different taxa. This requires manual labour, making it time-consuming. Today, with the aid of computers to process and identify patterns in large datasets, several of these processes can be automated. In recent years, advancements have been made in fields such as ML and AI, bioinformatics, and image analysis. An area within image analysis and AI that has recently emerged and seen rapid growth is Computer Vision. This field aims to automate information collection from images, and in some way mimic human vision.

As the pre-study showed, there was a large interest in automatic image recognition. Hence, this module introduces two very different ways to work with AI and ML. The first way introduced is to join a community/social network like *iNaturalist* which provides the service of automatic species identification. The other is to train a model utilizing data available of your choosing. Both approaches have advantages and disadvantages, so picking the approach which suits one's needs is crucial for success.

2.2.1. Crowdsourcing Computer Vision

There exist several social networks for naturalists, citizen scientists, and biologists, but with 2.5 million registered users few are as large as *iNaturalist*. From crowdsourcing, over 340 000 different species have been documented. The span of organisms from all different kingdoms is unmatched compared to many different platforms. When a species have over 20 images of research-grade quality, it will be included when they update the image recognition model. Currently 38 000 species satisfy this criterion and can be identified with ML. There are several advantages and disadvantages of utilizing a service such as *iNaturalist*.

A disadvantage of utilizing *iNaturalist* is lacking control of the image recognition model. If you want to distinguish fungi by spores or include metadata, it might be advantageous to create your own model with the images you own, as it might not be the main priority of the developers of their model to incorporate specific details which suit one user's needs. Additionally, as the data is published online, it might not be recommended to use as main and only storage of your images, nor use it if the images should be private.

One major advantage is there being no need for programming skills to have advanced animal classification at your fingertips for many different species. It can also be difficult to reach the amount of data required to train models with high accuracy on your own, as this data collection is crowdsourced. Large amounts of data are collected every hour which can help improve the model accuracy. Hence, without any knowledge of programming, you can contribute to a larger dataset and better ML models used worldwide.

In all crowdsourcing media it is important to have a quality control mechanisms in place to reduce false identifications by less experienced users. *iNaturalist* classifies images as a casual observation and research-grade quality. This will also reduce the error in the ML model; because when a species has over 20 observations of research-grade, it will be included [18]. The verification process is shown in Figure 6 [19].

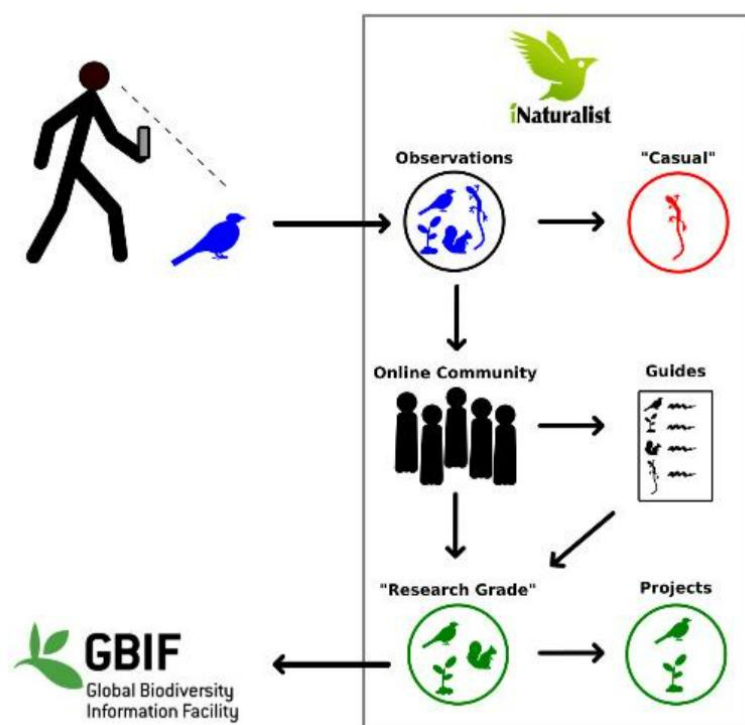


Figure 6: The verification process used by iNaturalist[19]

2.2.2. Training Your Own Model

In some cases it can be advantageous to create your own model. This section will provide information regarding the examples shown in the demonstration and where to find more information. All example images used in the demonstration to show the capabilities of the model are example data from Nina Filippova at Mukhrino field station within the INTERACT network [20][21].

In early 2022 Picek et al. achieved groundbreaking performance on the classification of fungi. The research group achieved more than 83.4% accuracy for the top 1 accuracy for 1604 danish fungi species. The group decided to open source two different datasets used in their study. Additionally, the group open-sourced the code for the training process and checkpoints for the model [22][23]. The process requires knowledge in setting up a docker container, jupyter notebook, and Python programming. One of the datasets contains over 30 000 images of 182 different species of fungi, while the other which is bigger contains over 300 000 images for 1 604 species.

For this demonstration the use of the smaller dataset was chosen due to computation time to train the model. The instructions given by Picek et al. were followed to train the model for 5 epochs. This yielded the top 1 accuracy of 56%. Utilizing data loader images could be given to the model which would give a fungi species estimation. The whole output of the program can be seen in Figure 7, where an image of *Amanita*

muscaria is fed to the model. The output result can be seen above the image itself. For this case the model predicted the correct species and was very sure of its decision.

```
[38]: PATH='amanita_muscaria_02_YUGRA.jpg'
      ##Visa bra resultat
      d = {'taxonID': [0], 'image_path': [PATH]}
      df_input = pd.DataFrame(data=d)
      input_dataset = TrainDataset(df_input, transform=get_transforms(data='valid'))
      input_loader = DataLoader(input_dataset, batch_size=BATCH_SIZE, shuffle=False, num_workers=WORKERS)
      get_prediction(input_loader,PATH)

      Predicted top1 fungi: ['Amanita muscaria'] ,Psuedo Certainty: 99.99908447265625 %
```



Figure 7: Shows the output of the example model on *Amanita muscaria* from example data

In Figure 8 three images of fungi that were correctly classified by the model are shown. The image to the left depicts an *Amanita vaginata*, the center is a *russula vesca*, the left image is *Mycena gellericulata*. As the model had a top 1 accuracy of 56%, it will occasionally predict the wrong species. Figure 9 shows examples of images where the ML model got the result wrong. The image on the left shows a *Cortinarius sanguineus* and the image on the right is *Russula vesca* as in Figure 8. The reason for the model predicting the wrong result for *Cortinarius sanguineus* is easily explained by the fact that this species was not included in the dataset of 182 Danish species. The model has never seen this fungus before and therefore it made a

prediction based on what it knows. The model guessed this was a *Russula emetica*, while this is clearly not the case for a well-trained eye, due to the model never having seen any fungi in the family of *Cortinarius*, and having made a guess which is believed to be the most similar of what it had seen. This model had no option of unknown or outside of dataset option for the model. If this model was deployed for the public without any modification, this could result in misclassifications for fungi which can have devastating consequences, i.e., if it had predicted a poisonous mushroom species to be an edible one. This highlights the importance of a close working relationship between developers and researchers to prevent this from happening. The other case where the model predicted wrong is harder to understand, but more forgiving. The model had seen *Russula vesca* previously, but it mistook it for a relative in the same family which has similar visual characteristics.



Figure 8: Examples of fungi images where the model got the correct species in the top1 result.



Figure 9: Examples of fungi images where the model got the wrong species in the top1 result.

2.2.3. Summary

There are several ways to utilize computer vision models and to contribute to their improvement. One way is to join an international crowdsourcing community like iNaturalist. This can be done without any previous experience and still let you have an intuitive user experience and give you powerful results. To get more

specific models created for your need, it is often better to train the model yourself. This can result in very precise results for your needs. However, it also requires lots of data. It is also important to remember how to handle false classifications especially if it is a model which is going to be used by the public and not experts. How to use iNaturalist was demonstrated during the demonstration, and the result of the self-trained model was presented on INTERACT research station data.

2.3.Module 3 – AI with Satellite Data for Earth Observation

Using satellite data for information retrieval brings many opportunities and can efficiently assist climate research.

2.3.1.Satellite Data

Satellites are often equipped with sensors that can collect data that can be used to retrieve important climate or Arctic information. Examples of the sensors can be Synthetic Aperture Radar (SAR) or spectral bands outside of what is visible for humans. More data increase the possibility to find more information. Also, wide swaths of satellite sensors and the perspective, give larger and orthogonal aerial imagery.

2.3.2.Examples of Available Services

There exists a lot of satellite data and providers of services derived from satellite data access, although most are expensive. Regarding this statement, there exist examples, like the European Union's Earth Observation program called Copernicus. The program offers data and information services. There exists a vast amount of global satellite data, which can be accessed by different free and openly accessible services [24].

Two different examples of services were mentioned in the demonstration: SentinelHub and EO Browser [25]. The services are both developed by the company Sinergise and aim to give real-time visualization of multi-spectral data supports, building new earth-observation products [26]. In these tools, there are many possibilities to explore and investigate for data exploration and information retrieval. One of the most interesting uses of these tools is to incorporate customized scripts [27]. The scripts are often results of studies or the community revolving around Earth Observation, and are ready-to-use software as well as free to share and adapt (licensed under a Creative Commons Attribution-ShareAlike 4.0 International License) [28].

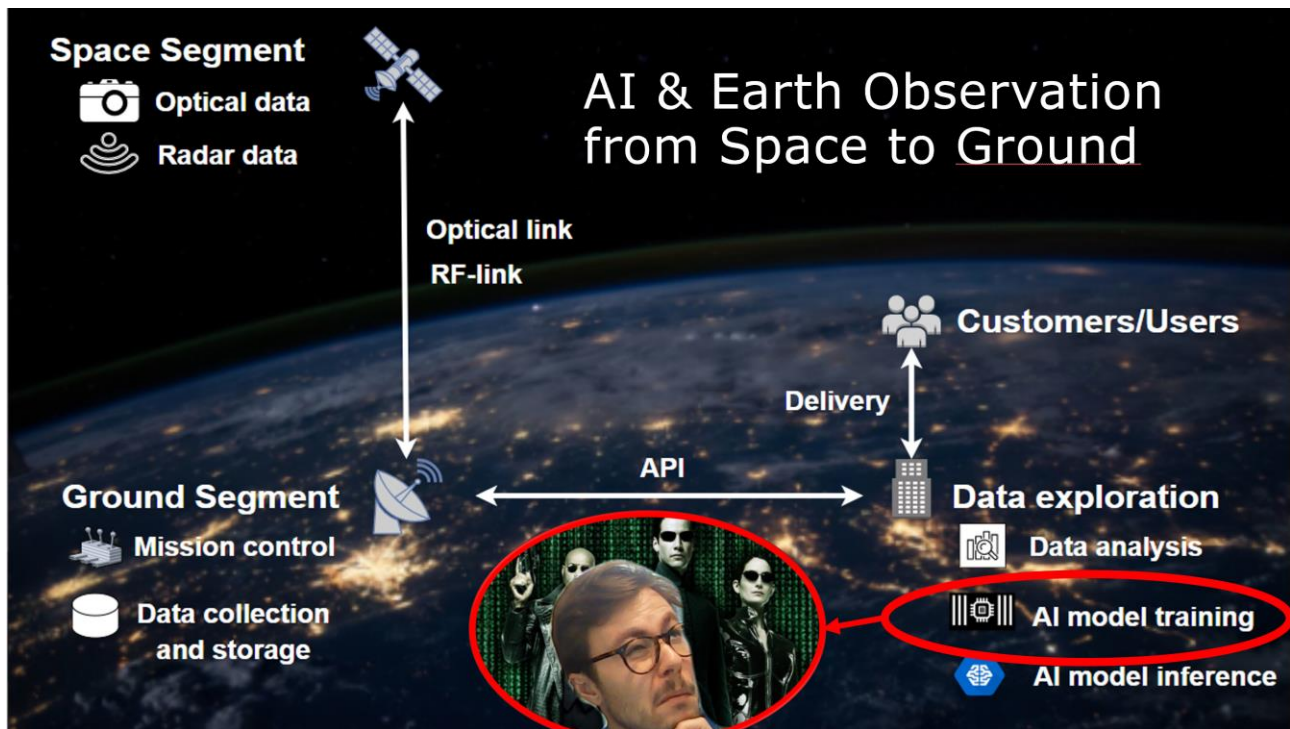


Figure 10: Slide from PowerPoint, emphasizing that training AI models can be difficult, with an example of general data to information flow can look like with satellite data.

2.3.3. Earth Observation Examples

Using the results is advantageous since incorporating AI/ML in an easier way for analyzing satellite data requires some coding skills. Some examples, emphasizing some use cases for Arctic research were presented. Among these examples were the scripts analyzing to detect and visualize snow cover change detection, normalized difference chlorophyll index, evapotranspiration levels composite and deforesting. All examples were made adjacent to INTERACT III stations' positions. Although many of the available scripts work fine; during the demonstration, it was pointed out that the quality varies, depending on where and when to use the script. One of the primary reasons is that many scripts have been developed for one particular use-case on a specific place on Earth.

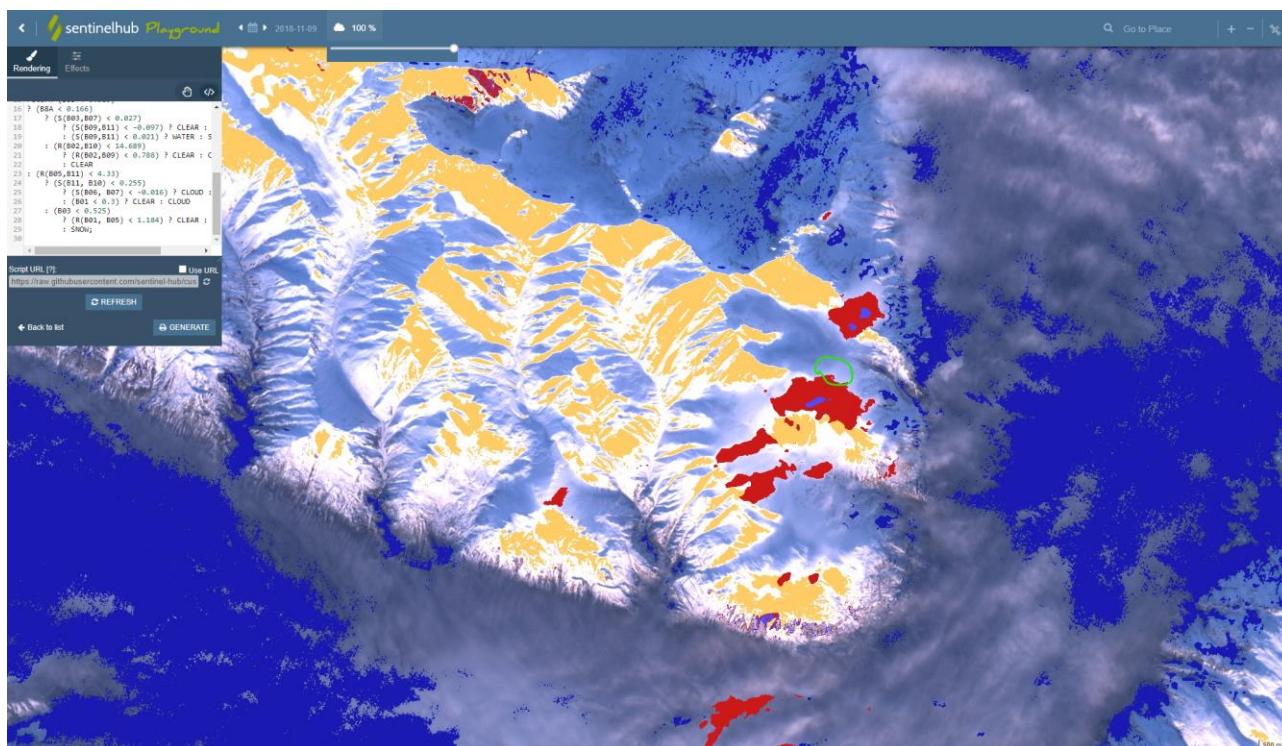


Figure 11: An example output of using a custom script in SentinelHub; green circle is the position of Kluane Ranges, Canada.

After highlighting some example cases of how one could use satellite data to retrieve information, a proposed way to become an Earth Observation Expert was presented by giving some examples of how one can use the same services and data in more elaborated manners.

2.4.Module 4 – Digitizing Text with OCR and Summarization with NLP

This part of the demonstration focuses on using Optical Character Recognition (OCR) and Natural Language Processing (NLP) to digitize and summarize analog text documents, by incorporating State-of-the-Art tools.

The AI/ML tools used were Google’s Vision AI and OpenAI’s GPT-3, accessed by application programming interfaces (APIs)[29][30][31][32]. The Integrated Development Environment (IDE) used was Google Colab - which allows combining executable code and rich text in a single document [33]. In ten steps, a Google Colab Notebook sequentially presented how one can get started, use and retrieve results with these tools [34].

Regarding using the code, it was brought up that the OCR works with both handwritten and printed digital text and that there could be a need to choose both OCR and NLP parameters for optimal outputs – given use case.

This demonstration was made possible by Maja Linderholm, an AFRY Master’s Thesis student, working for INTERACT III and Stockholm University/Tarfala Research Station providing the data for Maja’s project.

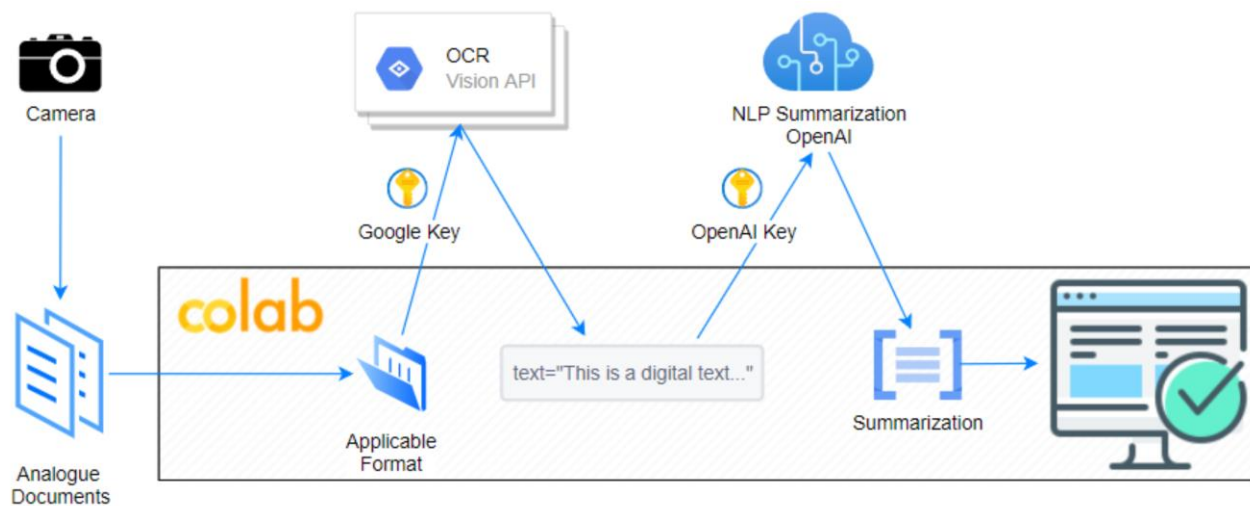


Figure 12: A diagram showing the flow of data for the demonstration and the Master's Thesis project.

Instructions on how to access the tools were provided since there is a need to have both Google Vision and OpenAI API keys - which can be created on their websites by creating an account and is free to some extent[35][36][37][38].

The digitization with OCR and summarization with NLP were then run in code showing the capabilities of the tools. Input data and results can be observed below in Figure 13. Additional examples of use-cases and applications of OpenAI's GPT-3 were presented and discussed [39]. The Colab Notebook can be found at this url: https://colab.research.google.com/drive/1C_q8gzfxMOxpcj0dCWNbWBEHleuKcLhq?usp=sharing

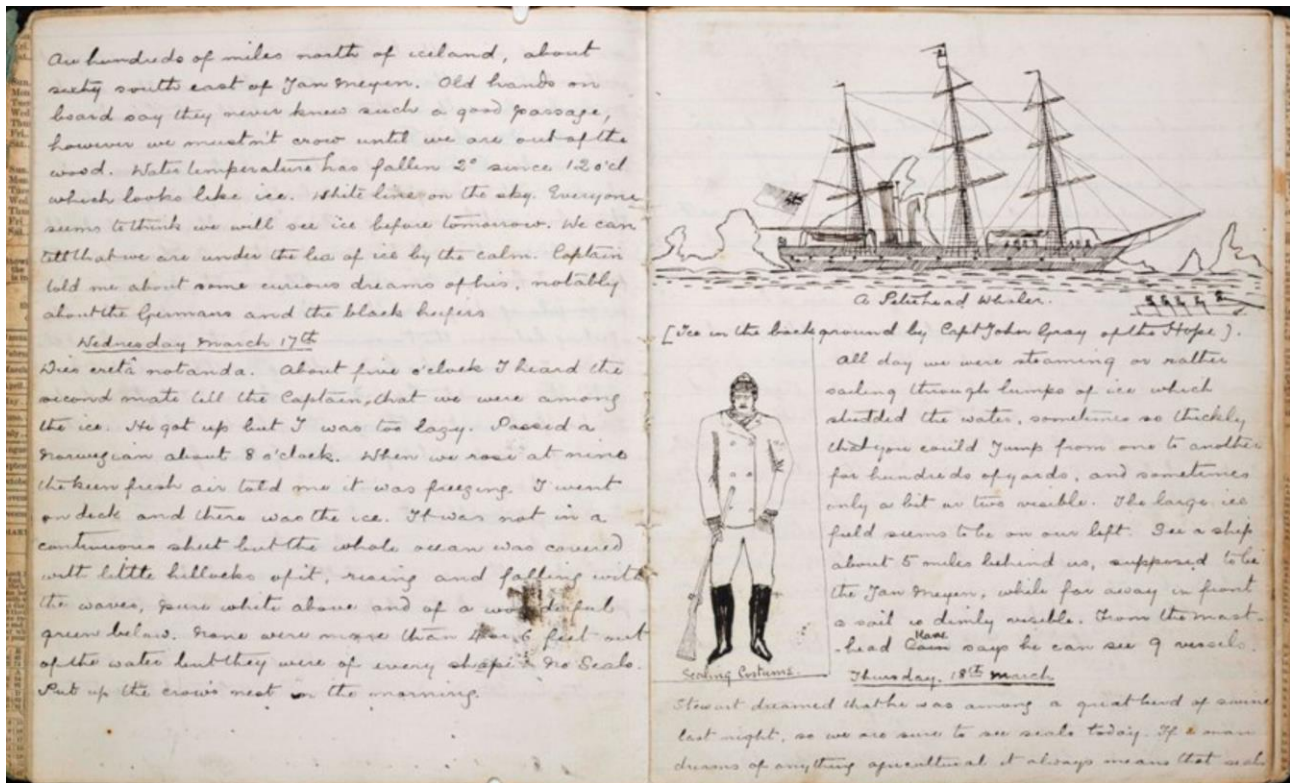


Figure 13: The image shows the input image. The resulting output was: “The crew of a ship sailing north of Iceland near Jan Mayen Island observed what looked like ice and woke the captain, who confirmed that it was in fact ice. The ship continued to sail north and eventually encountered more ice, which surrounded the ship and stopped its progress.”.

2.5. Module 5 – Azure Machine Learning Designer on Tabular Data

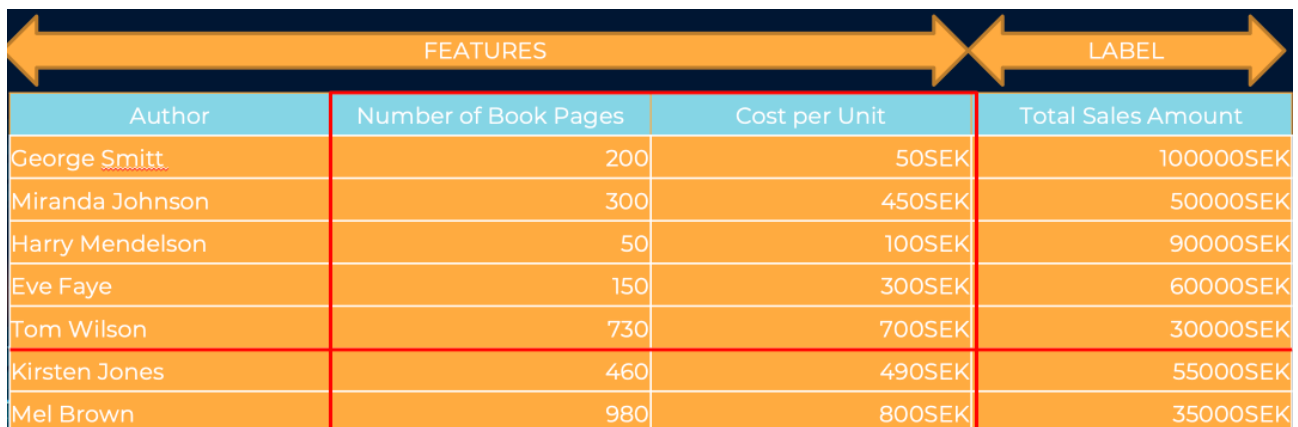
In this section, a powerful and flexible machine learning tool was employed. It is called the Azure Machine Learning Designer [40] and it is a graphical tool, incorporating ML algorithms by connecting modules on a visual canvas. The Azure ML Designer offers not only an intuitive way of learning or teaching ML, but combined with cutting-edge cloud computing, it is also possible to professionally deploy your ML model in production.

2.5.1. Cloud Computing – Azure

Azure is Microsoft’s cloud computing solution. Other popular examples are Amazon Web Services and Google Cloud. Cloud computing is the delivery of computing services including servers, storage, software, analytics, intelligence etcetera [41]. A benefit is that you only pay for what you need, and it is possible to scale up or down depending on required resources. In other words, there is no need to invest upfront in capital expenses without knowing whether these will be properly utilised, or not be sufficient. As a cloud computing solution, Azure is unique in the way that it offers a graphical tool that helps visualise an ML pipeline with a drag and drop interface.

Mathematically, ML can be thought of as a function $f(x) = y$, where x is considered features, y is considered the label, and the function is the ML model. In most cases data require pre-processing, e.g., identifying relevant features, cleaning the data from irregular/unwanted values, taking care of missing values, normalising the data etcetera. Thereafter, an algorithm is applied to fit the data, and the model is trained. Afterward, the performance of the model should be evaluated. The model can then be used to make predictions from your input values or features. Azure also provides the option to deploy the model in production.

Figure 14 shows an example tabular data set where features or input values, and the label have been denoted. Before training a ML model it is worth asking what features are relevant. In Figure 14 the number of book pages, and the cost per unit could be relevant; however, the name would not make sense to use as a feature to train the model and should thus be excluded. Furthermore, the data set will be divided into a training and test data set row-wise, but it is worth taking into consideration that the division of the rows is random. Usually, a good starting point is to use 70% of the data set for training and 30% for testing.



| Author | Number of Book Pages | Cost per Unit | Total Sales Amount |
|-----------------|----------------------|---------------|--------------------|
| George Smitt | 200 | 50SEK | 100000SEK |
| Miranda Johnson | 300 | 450SEK | 50000SEK |
| Harry Mendelson | 50 | 100SEK | 90000SEK |
| Eve Faye | 150 | 300SEK | 60000SEK |
| Tom Wilson | 730 | 700SEK | 30000SEK |
| Kirsten Jones | 460 | 490SEK | 55000SEK |
| Mel Brown | 980 | 800SEK | 35000SEK |

Figure 14: Features and label of an example data set, where both the relevant features are outlined, as well as dividing a data set into training and test data sets respectively

2.5.2. Azure ML Designer

Figure 15 shows a standard ML pipeline. The first module dragged and dropped to the Azure ML Studio pipeline is the dataset, after which the module 'split data' is connected, which divides the data into a training and test data set. The Train Model module takes an algorithm (of which there are many included in Azure ML Studio with regards to classification, clustering, regression, anomaly detection etcetera), in this example, a Two-Class Boosted Decision Tree, and the training data set as inputs. The Score Model module scores how well the trained model performed compared to the test data. Finally, the Evaluate Model module provides valuable metrics and statistics to help evaluate the ML model.



Figure 15: Pipeline in Azure ML Designer

During the demonstration it was also shown how to set up a powerful computing cluster, to which the ML model was deployed to production. Essentially, an endpoint was established to which API calls can be made.

2.5.3.Summary

Cloud computing is the most suitable method for using ML models in professional settings. Module 5 of this demonstration showed how to utilize Microsoft's cloud solution, Azure, and its graphical tool the ML Designer to create an ML pipeline. This pipeline consisted of dragging modules onto a visual canvas. The modules had as their constituent parts the dataset, splitting the dataset into a training and a test data set, the algorithm and scoring and evaluating how well the ML model performed compared to the test data.

The Azure ML Designer, given its drag and drop interface, can also be used to learn and teach ML in an intuitive way.

3. Conclusions and Future Work

In the pre-study, delivered in D6.1, several potential use cases for AI and ML applications for Arctic science were presented, along with the needs identified within the INTERACT network. In this demonstration (D6.3) several specific use cases and methods accommodated to these needs were presented.

The first two modules of the demonstration covered several approaches of Computer Vision being utilized to analyze image example data from various research stations within the network. Various tools, requiring varying degrees of technical knowledge, were used. It is worth noting that stations can implement the most simple tools immediately, which will aid in generating new insights. These modules also showed how the research community and voluntary contributions can assist in making new scientific progress; in this case with regards to the tools Camelot and iNaturalist.

The third module covered how public satellite data can be used to analyze changes in the landscape around the stations. It also illustrated how pre-existing scripts and ML-results can be used to meet specific needs, e.g., movement and texture changes in glaciers. For more advanced users, scripts could be updated or created from scratch to be especially suited to an uncommon task. However, researchers should start by looking at the pre-existing solutions to quickly get an overview of the possible insights and results suited to their research needs.

The fourth module described how OCR and NLP can be powerful tools for digitizing old data. It also showed how the free cloud computing service Google Colab can be used as a powerful tool for exploring ML methods.

The fifth and final module showcased how a powerful cloud computation model in Azure could be built from scratch without considerable previous knowledge. Azure also provides a graphical UI that visualizes the ML-model, which makes it easier to understand and update. Models like these can be trained to make predictions and analyses of different data types, and requires little more than an Internet connection to start developing.

To summarize, the demonstration showed that there already exist many exciting possibilities for INTERACT researchers to utilize AI tools to analyze their data and create new insights, even without coding. These possibilities should not be overlooked and the AFRY team highly recommends starting smaller research projects to utilize these techniques.

It is suggested to look into the demonstrated options, since these are options to achieve good results with minimum effort. However, if a specific need identified by several stations can not be handled by these pre-existing solutions, it would be recommended to put effort into developing a new model which would yield satisfactory results, for example, a Computer Vision model highly specialized on Arctic flora and fauna, or a script for tracking reindeer grazing with satellite data. If such a need is identified, powerful cloud computing services are a great way to get started with a new model.

4. Bibliography

- [1] J. Bughin *et al.*, “Artificial intelligence: the next digital frontier?,” 2017.
- [2] “Climate Change AI | Tackling Climate Change with Machine Learning.” <https://www.climatechange.ai/> (accessed Feb. 16, 2022).
- [3] “Providing leading solutions for generations to come | AFRY.” <https://afry.com/en> (accessed Feb. 16, 2022).
- [4] A. Buras, M. Hallinger, and M. Wilmking, “Can shrubs help to reconstruct historical glacier retreats?,” *Environmental Research Letters*, vol. 7, no. 4, 2012, doi: 10.1088/1748-9326/7/4/044031.
- [5] “AI & Machine learning demonstration - YouTube.” <https://www.youtube.com/watch?app=desktop&v=4pb5dvLI5E&feature=youtu.be> (accessed Feb. 16, 2022).
- [6] “Research and Monitoring - INTERACT.” <https://eu-interact.org/publication/research-and-monitoring/> (accessed Feb. 16, 2022).
- [7] A. F. O’Connell, J. D. Nichols, and K. U. Karanth, *Camera traps in animal ecology: methods and analyses*, vol. 271. Springer, 2011.
- [8] A. C. Burton *et al.*, “Wildlife camera trapping: A review and recommendations for linking surveys to ecological processes,” *Journal of Applied Ecology*, vol. 52, no. 3. Blackwell Publishing Ltd, pp. 675–685, Jun. 01, 2015. doi: 10.1111/1365-2664.12432.
- [9] “Snapshot Serengeti — Zooniverse.” <https://www.zooniverse.org/projects/zooniverse/snapshot-serengeti> (accessed Feb. 16, 2022).
- [10] “ECN Cairngorms - INTERACT.” <https://eu-interact.org/field-sites/ecn-cairngorms/> (accessed Feb. 16, 2022).
- [11] S. Sharp, “Evidence from Camera Trap Data on the Relationship between Human and Animal Presence in the Allt a’ Mharcaidh ECN Site Copyright Declaration,” 2019.
- [12] “Trail Cameras.” <https://www.spypoint.com/en/products/trail-cameras> (accessed Feb. 16, 2022).
- [13] F. Örn, “Computer Vision for Camera Trap Footage : Comparing classification with object detection,” no. 21037. p. 56, 2021.
- [14] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” Apr. 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [15] S. Beery, D. Morris, and S. Yang, “Efficient Pipeline for Camera Trap Image Review,” Jul. 2019, [Online]. Available: <http://arxiv.org/abs/1907.06772>
- [16] “Megadetector_Colab_test.ipynb - Colaboratory.” https://colab.research.google.com/github/microsoft/CameraTraps/blob/master/detection/mega_detector_colab.ipynb (accessed Feb. 16, 2022).
- [17] “Camelot Project – Make camera trap image organisation quick, easy and free.” <https://camelotproject.org/> (accessed Feb. 16, 2022).
- [18] “iNaturalist - Wikipedia.” <https://en.wikipedia.org/wiki/iNaturalist> (accessed Feb. 16, 2022).
- [19] M. E. Boone and M. Basille, “Using iNaturalist to Contribute Your Nature Observations to Science 1.” [Online]. Available: <https://www.gbif.org/>
- [20] “Mukhrino Field Station - INTERACT.” <https://eu-interact.org/field-sites/mukhrino-field-station/> (accessed Feb. 16, 2022).
- [21] “Occurrence search.” https://www.gbif.org/occurrence/gallery?dataset_key=d922b606-6c94-4d51-9277-36c9b03872a7 (accessed Feb. 16, 2022).

- [22] L. Picek, M. Šulc, J. Matas, J. Heilmann-Clausen, T. S. Jeppesen, and E. Lind, “Automatic Fungi Recognition: Deep Learning Meets Mycology,” *Sensors*, vol. 22, no. 2, Jan. 2022, doi: 10.3390/s22020633.
- [23] “GitHub - picekl/DanishFungiDataset.” <https://github.com/picekl/DanishFungiDataset> (accessed Feb. 16, 2022).
- [24] “About Copernicus | Copernicus.” <https://www.copernicus.eu/en/about-copernicus> (accessed Feb. 16, 2022).
- [25] “Sentinel Hub EO Browser.” <https://apps.sentinel-hub.com/eo-browser/?zoom=10&lat=41.9&lng=12.5&themeId=DEFAULT-THEME&toTime=2022-02-16T12%3A00%3A22.940Z> (accessed Feb. 16, 2022).
- [26] “What we do | Sinergise.” <https://www.sinergise.com/en/what-we-do> (accessed Feb. 16, 2022).
- [27] “Sentinel Playground.” <https://www.sentinel-hub.com/explore/sentinelplayground/> (accessed Feb. 16, 2022).
- [28] “Creative Commons — Attribution-ShareAlike 4.0 International — CC BY-SA 4.0.” <https://creativecommons.org/licenses/by-sa/4.0/> (accessed Feb. 16, 2022).
- [29] “Vision AI | Derive Image Insights via ML | Cloud Vision API | Google Cloud.” <https://cloud.google.com/vision> (accessed Feb. 16, 2022).
- [30] “OpenAI API.” <https://openai.com/api/> (accessed Feb. 16, 2022).
- [31] “OpenAI API Davinci.” <https://beta.openai.com/playground/p/default-tldr-summary?model=text-davinci-001> (accessed Feb. 16, 2022).
- [32] “Detect handwriting in images,” *Google*. Google. [Online]. Available: <https://cloud.google.com/vision/docs/handwriting>
- [33] “Welcome To Colaboratory - Colaboratory.” <https://colab.research.google.com/> (accessed Feb. 16, 2022).
- [34] “ocr_ & nlp_interact-demo - Colaboratory.” https://colab.research.google.com/drive/1C_q8gzfxMOxpcj0dCWNbWBEHleuKcLhq?usp=sharing (accessed Feb. 16, 2022).
- [35] “OpenAI API key.” <https://beta.openai.com/account/api-keys> (accessed Feb. 16, 2022).
- [36] “Quickstart: Setup the vision API,” *Google*. Google. [Online]. Available: <https://cloud.google.com/vision/docs/setup>
- [37] Openai, “Openai Sign up.” <https://beta.openai.com/signup> (accessed Feb. 16, 2022).
- [38] Google, “Google cloud platform.” <https://console.cloud.google.com> (accessed Feb. 16, 2022).
- [39] “Examples - OpenAI API.” <https://beta.openai.com/examples> (accessed Feb. 16, 2022).
- [40] “What is the Azure Machine Learning designer? - Azure Machine Learning | Microsoft Docs.” <https://docs.microsoft.com/en-us/azure/machine-learning/concept-designer> (accessed Feb. 16, 2022).
- [41] “Cloud Computing Services | Microsoft Azure.” <https://azure.microsoft.com/en-us/> (accessed Feb. 16, 2022).