Preliminary title

Svensk titel
Klassificering av noshörningar via kamerafällor och faltningsnätverk med driftsättning i en kritisk miljö
Engelsk titel
Rhino classification in camera traps with CNN, deployed in a critical environment

Background

Biodiversity is declining at a rapid pace because of human settlement expansion, illegal wild-life killing, and climate changes. Conservationists are faced with the tough quest of surveying the wildlife populations. Data in the form of images is the most powerful tool to estimate the wildlife population, however, today most images are annotated manually. With computer vision, the time to annotate the images can be reduced and the conservation decision-making can be accelerated. AI for Earth is an initiative by Microsoft which provides AI tools and cloud solutions in the hands of those working to solve environmental challenges. In particular, they develop image-based tools for surveying biodiversity.
Black rhinoceros is one of these critical populations. Ngulia is a rhino sanctuary in Kenya and home to about 80 individuals. These animals are vulnerable to poaching which drives Project Ngulia, a public-private partnership, to combat this problem. Today, the rangers use smartphones with an app to report whatever they see during their patrols, and the officers can in real time see their reports and summaries from the system. Also, there are camera traps on site. However, these cameras run on batteries and simply store their data on memory cards which must be replaced on a weekly basis. Another drawback is the manual work to review the recordings and register moments when rhinos or humans pass the camera trap. Consequently, information about interesting observations will not reach rangers in real time. This time delay can be crucial for the survival of these animals as they generally are killed when the rangers are out of reach - and a motivation to automate this process.

Aim

The aim is to develop a convolutional neural network (CNN) that automatically analyses the video content from camera traps that are located in a rough environment. Focus is on classifying the rhinos against other animals and in the best of cases, identify the individuals. The CNN model will be trained on annotated images from Kolmården Zoo and other images of black rhinos. The camera traps will be placed in Ngulia together with a microcontroller. Included in the scope of this project is to also investigate all constraints necessary to deploy the system. This includes functionality on the server-side and to consider security aspects, as well as to configure the microcontroller to work with the camera and classification model.

Research questions

Based on the aim, five research questions are set:

- What constraints applies to the CNN implementation due to its goal platform, a microcontroller with limited performance?
- How can it be evaluated that the hardware can sustain the rough environment in the Kenyan Savannah?
- Is it possible to classify the rhinos individually? If so, what approach will work as the individuals accessible for testing are of another species of rhino.
- How can a camera observation over a time interval be assembled into a single event? More specifically, if an animal moves out of image, how long time can pass before it is considered to be a new event when it appears again?
- What images are interesting to send to the server based on the needs from the officer in charge and how can they be extracted?

Theory

The major part of literature and theory to review covers the topic of deep learning, and in particular object detection and classification. Relevant to this work is also video tracking and understanding about implementation on microcontrollers. Grouped by subject area, algorithms and insights and will be given from the research papers below.

Deep learning

In the field of machine learning, deep learning is a method based on artificial neural networks. When dealing with images, in particular, a convolutional neural network (CNN) is the most common approach. The state of the art CNN's achieve many times an accuracy close to 100% on complex tasks, a convincing number for picking it as the core method of this project. Its shared-weights architecture is suitable to detect features invariant of its placement in the image. CNN's take advantage of the fact that images can be interpreted as a hierarchical pattern. That is, if the actual image contains 4 faces and the network is trained on real images, some of the final filters may respond to faces while its preceding filters represent simpler patterns. Those early filters may appear as circles which eyes match to, horizontal edges for a mouth, and so on. During training the weights for each filter updates continuously after each example along with the ground truth are given to the network. A CNN is formed by a stack of layers that typically appears in a similar order in many implementations, however with variations. Common are still the main "building blocks" which are layers of the following types: convolutional layers as the core learnable filters, pooling layers for down-sampling which typically follows a convolutional layer, and ReLU layers to remove negative values as well as it makes the network nonlinear which is an important property. Finally, after rounds of these building blocks just mentioned, a fully connected layer is applied to summarize all activated layers, resulting in a vector of real numbers. To predict a class based on this vector, it is often normalized into a probability

distribution by the softmax function. This probability distribution can be interpreted as for example: with probability 0.97 it is a rhino and with probability 0.03 it is an elephant.

Many species can be identified on an individual level by analysing surface patterns. Rhinos have no salient pattern in their skin, thus theory on pattern identification cannot be directly applied. However, rhinos do have attributes pattern unique enough to possibly identify individuals, as the shape of their ears and their skin folds. In related work, algorithms to detect a part on the animal are used, where this part depends on the species. The detected part is then compared to records in a database. Data augmentation is also a central part of deep learning with rare classes, as the amount of data available is often low. Other research has presented demonstrative results with as low as 687 images.

Deep learning literature:
• *Learning to detect unseen object classes by between-class attribute transfer*[1]
• *Automatic Detection and Recognition of Individuals in Patterned Species*[2]
• *Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna*[3]
• *Automated identification of animal species in camera trap images*[4]
• *A deep active learning system for species identification and counting in camera trap images*[5]
• *Synthetic examples improve generalization for rare classes*[6]

Video tracking

To avoid the system to act on every single frame where a rhino is encountered, methods within the subject of video tracking are relevant. The objective of video tracking is to locate a moving object(s) over time. Further, it can be combined with the task of object recognition, i.e. to also identify the object. Implementation of video tracking differs depending on the intended use which affects the two major components of the process, target representation and localization. They deal with the question about how to represent the moving target, e.g. using blob tracking or contour tracking. Blob tracking is as for example more adaptable to objects that may change orientation and thus profile dynamically.
In addition to target representation and localization, methods for filtering and data association are needed for more complex objects and/or objects that may move behind obstructions. Kalman filtering is an algorithm for this purpose which takes a series of

[1] C. H. Lampert, H. Nickisch and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," *2009 IEEE Conference on Computer Vision and Pattern Recognition*
[2] Cheema G.S., Anand S. (2017) “Automatic Detection and Recognition of Individuals in Patterned Species”
[3] Swanson, A., Kosmala, M., Lintott, C. et al.” Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna” Sci Data 2, 150026 (2015)
[4] Yu, X., Wang, J., Kays, R. et al. J Image Video Proc (2013) 2013: 52. https://doi.org/10.1186/1687-5281-2013-52
[5] Norouzzadeh M, Morris D, Beery S, Joshi N, Jojic N, Clune J. A deep active learning system for species identification and counting in camera trap images. arXiv preprint arXiv:1910.09716. 2019 Oct 22
[6] Beery S, Liu Y, Morris D, Piavis J, Kapoor A, Meister M, Perona P. Synthetic examples improve generalization for rare classes. arXiv preprint arXiv:1904.05916. 2019 Apr 11.

measurements observed over time and produces state estimates, where a state for instance can be position and velocity of the tracked object.

Video tracking literature:
• *Real-time video tracking using PTZ cameras*[7]
• *Video object tracking based on extended active shape models with color information*[8]
• *Real-time tracking of non-rigid objects using mean shift*[9]

Hardware - running om microcontroller

Having a pre-trained classification model to run on a microcontroller as Raspberry Pi has proven to work, however the model might need to be converted into an optimized model format. This conversion can be made with the use of Tensorflow, and if other machine learning libraries have this functionality will be investigated to be aware of all constraints. Moreover, quicker model predictions can be achieved by the addition of a USB accelerator connected to the microcontroller.

Literature regarding deep learning implementation on microcontrollers:
• *MobileNetV2: Inverted Residuals and Linear Bottlenecks*[10]

Other

In 2004, a similar work was done as a course project at Linköping University. While hardware differs and the scope of the project was smaller, valuable insights can be found in their paper, however it is not officially published.

[7] Kang, Sangkyu & Paik, Joonki & Koschan, Andreas & Abidi, Besma & Abidi, Mongi. (2003). Real-time video tracking using PTZ cameras. Proceedings of SPIE - The International Society for Optical Engineering.
[8] A. Koschan et al, "Video object tracking based on extended active shape models with color information," (2002)
[9] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000
[10] Mark Sandler et a. (2018) “MobileNetV2: Inverted Residuals and Linear Bottlenecks”

Method

Initially, research and data collection will be the major activity of the work. These two go hand in hand in this case, as for example more research on transfer learning is required if the amount of data available is low. Likewise, if it is possible to obtain annotated images of individual rhinos, research on methods for individual recognition will be applicable.

Data collection and preprocessing:
Preprocessing of data is known to require a significant amount of time. As the data comes from mixed sources, their annotation structure will differ or be missing completely. Often when classifying images, pure image processing operations are done to separate interesting objects from the foreground as an example. As part of the preparation work, methods to accomplish this separation will be investigated.
Kolmården Zoo has a population of four rhinos on an area of 1.5 km2. A camera-trap placed there will, therefore, produce a lot of images of rhinos in a Savannah-like environment. The images need to be annotated manually before training the model which will take time.

CNN implementation
With preprocessed data in hand, the implementation of a CNN model can start. It is common and often recommended to pick a pre-trained model to start with, and continue with further tuning with application specific data. This pre-trained model should have been trained on a large dataset similar to solve a similar problem, in this case object detection with real images. The open source community within deep learning is generous and pre-trained models and image databases are easily accessible. Models to look further into for the scope of this project are as for example ResNet50 or the small-sized model MobileNetV2 suitable for embedded applications.
When the model manage to classify an arbitrary rhino, the challenge to identify individuals will be tackled, corresponding to the goal of the third research question. In this case, there are no known approaches for rhinos in particular, however methods for identifying other animals will be modified and tested.

Hardware setup
There are a few hardware setups available for this project and their pros and cons will be compared during the research phase. To mention an example of a possible constraint, the bandwidth may differ between the hardware options and thus limit how much data can be sent. The task to get a program running with the camera will include work itself, as for example pure setup configurations as well as tests regarding what is possible with the processor speed and available RAM. Insight regarding the first research question regarding what constraints applies to the CNN implementation will be achieved in this stage. The performance on a deep CNN as ResNet50 running with “unlimited” processing power can be compared to the performance of what model size can operate on the microcontroller.

Video tracking implementation
As the theory section about video tracking describes, video tracking will be implemented using classification results as the series of measurements. As described, there are several approaches

and to answer the fourth research question it will require implementation tests to figure out the best option for rhinos in the scenario where individual recognition is of importance.

Server communication and UI
As the equipment works along with the classification model, the whole system including communications between cameras and server will be set up, as well as a user interface to visualize the incoming camera observations. Answers towards the last research question regarding what data interesting to send to the server will emerge in this stage, keeping a dialog of the end users

Activities
When the classification model is performing good on the training data, the model can be transferred into a microcontroller to evaluate the real-time result on a camera setup in Kolmården. Stress-tests are crucial to make sure the system can maintain during a long time period, exposing the camera setup for situations that may appear in the real rough environment - as heavy impacts from curious animals. These tests are relevant to embrace the second research question which covers the topic about how can it be evaluated that the hardware can sustain. However, more research has to be done to distinguish what stress-tests are most critical, it might be something yet not covered in this suggested method.
Ideally, towards the end of the project a field trip to the rhino sanctuary Ngulia in Kenya will take off. In preparation, development of the complete system and testing will be in focus.

Delimitations

As the goal is to put this system into action, several constraints has to be taken into account to meet the real world conditions. The camera traps will be placed in a rough environment, occasionally with extreme heat and dust storms. Thus, neither smartphones or other non-robust devices are considered in the development of the system. The classification model might not correspond to the state of the art performance for this kind of problem if it requires too much bandwidth or RAM for a microcontroller. Moreover, the full solution would need to be cost-efficient.
Further the prediction accuracy might be limited by the preconditions regarding training and testing data. Kolmården Zoo which provides always available training and testing data has only white rhinoceros, while the rhinos in Ngulia are black rhinoceros.

Time plan

Below is a preliminary plan of activities and milestones.

Vecka	Aktiviteter och milstolpar
5 (27-31 jan)	Uppstart, Efterforskning Förbehandling av data
6 (3 feb - 7 feb)	Träffa samarbetspartner: Axis Förbehandling av data
7 (10 feb - 14 feb)	Påbörja klassificeringsmodell **Skicka in planeringsrapport**
8 (17 feb - 21 feb)	Ha en första klassificeringsmodell
9 (24 feb - 28 feb)	Design av systemarkitektur
10 (2 mars - 6 mars)	Test med live-data från Kolmården
11 (9 mars - 13 mars)	Överför modell till mikrokontroller
12 (16 mars - 20 mars)	Implementation av server-kommunikation
13 (23 mars - 27 mars)	Implementation av video tracking
14 (30 mars - 3 april)	**Halvtidskontroll**
15 (6 mars - 10 april)	Implementation av *video tracking* + igenkännande av individer
16 (13 mars - 17 april)	
17 (20 mars - 24 april)	Utvecklande av fullständigt system
18 (27 mars - 1 maj)	Utvecklande av fullständigt system + test
19 (4 maj - 8 maj)	
20 (11 maj - 15 maj)	Resa till Kenya, förhoppningsvis
21 (18 maj - 22 maj)	Rapport
22 (25 maj - 29 maj)	
23 (1 juni - 5 juni)	
24 (8 juni - 12 juni)	
25 (15 juni - 18 juni)	**Framläggning** och **opponering**