Monitoring endangered wildlife utilising computer vision models

University of Twente

Bachelor Creative Technology

Februari 2025

Arne Arends
s2829495

Supervisors
dr. Faizan Ahmed
dr. Deepak Tunuguntla

Critical Observer
dr. Marcus Gerhold

Abstract **This research created a semi-automated image classification pipeline which significantly reduces the manual labor required for image classification, with a specific focus on supporting the Boeren, Burgers en Buitenbeesten (BB&B) project. To address this challenge, a systematic literature review was conducted to evaluate current state-of-the-art methods and identify best practices in automated camera trap image classification systems and models.**
These insights guided the design of a flexible workflow in a Jupyter Notebook setup and led to the selection and fine-tuning of four different candidate models. Among these, a customized ConvNeXt model from Schneider et al. [1], retrained on a limited dataset of mainly the BB&B own dataset, achieved the highest accuracy of 95.45 percent across 22 classes, underscoring the effectiveness of the model. This outcome confirms the successful implementation of the model, which can be utilised inside of the semi-automated classification pipeline for reducing manual classification effort.

Acknowledgement
Before the start of this thesis, I want to thank my supervisors Dr. Faizan Ahmed and Dr. Deepak Tunuguntla for their guidance during the time of this project. Even though I was sometimes stubborn and unsure, their reassurance as I was along the way have enabled me to create the research as presented here.

1 - INTRODUCTION

The meadow birds in the Netherlands are endangered. Due to predator pressure and extensive use of farm land, the meadow bird population is declining. Several initiatives have seen the daylight in recent years, but have not yet been fruitful in stopping this decline. Currently, due to the small population of meadow birds, the focus is on predator pressure. The predator pressure depends on two main points. The amount of predators and potential prey in the breeding season, as well as the land use all year round of other species.

This graduation project is a small link in the chain of this bigger overarching subsidized project Boeren, Burgers and Buitenbeesten [1]. This initiative is seeking to create an all year round monitoring system, regarding the behaviour of animals, the area quality and management thereof, which is non-existing right now. This graduation project focuses on one of the steps inside the animal monitoring system. Part of the Boeren, Burgers and Buitenbeesten aim is to provide a tool to municipalities and

organisations like Staatsbosbeheer with a heat map of the likelihood of a predator being in a certain area. For example, the stone marten's territory is between 80 - 700 hectares [2]. This heatmap will be based on camera traps standing in the area and identifying these species, so that this heatmap can predict the chance of an animal being in a certain area. And that is exactly where this graduation project comes in.

Identifying species through image classification systems. The amount of images retrieved from the camera traps is tremendous and not feasible to determine by hand. It is a highly labour intensive task. Therefore this graduation project seeks to find a solution within the image classification area through machine learning models. Throughout the Boeren, Burgers and Buitenbeesten (BB&B) project, the camera traps have gathered over 500 GB of data, containing over a million images. The aim of this project is to use this dataset to create a semi-automated image classification system that is able to confidently identify species.

2 - BACKGROUND RESEARCH

2.1 Problem Statement

The vision of this project is to create a custom wildlife machine learning classification pipeline that can be utilized by someone with low skills in machine learning. The requirements inhibit: A trained model on as many as possible species found in the datasets provided by the Boeren, Burgers and Buitenbeesten project. Depending on the state-of-the-art through a systematic literature review, the focus should be on creating a classification model that is contained in an pipeline.

The challenge that needs to be solved is that there are currently no machine learning models out there that can identify the sought after species:

Birds: buzzard, goshawk, hen harrier/marsh harrier, sparrowhawk
Mammals: fox, beech marten, polecat, weasel, ermine, rat, house cat.

The machine learning model has to fit into a wildlife classification workflow, which should be able to run by any person.

2.3 Systematic Literature Review

2.3.1 Introduction

This Systematic Literature Review focuses on tackling the current state-of-the-art of

machine learning in the camera trap images environment of the past five years. The aim of this Systematic Literature Review is to get a better understanding of the current developments in wildlife trap camera image classification pipelines. Also, what kind of classification models and pre-processing mechanics are being used and are promising in the future.

2.3.2 Findings

This section analyses the methodologies, decisions and applications presented in recent literature on wildlife image classification systems. The analysis focuses on workflows and pipelines, data pre-processing techniques, the impact of transfer learning and the performance of various wildlife image classification systems.

WORKFLOWS AND PIPELINES

There are two workflows considered in wildlife classification tools. Generally, they fall under two umbrellas: a semi-automated workflow or a fully-automated workflow. As Vélez et al. [3] point out, a fully-automated workflow is only feasible if the model displays an accuracy that complies with the user needs or for systems that require near-real-time detection. Guo et al. [4] argue for an automatic workflow. Their model called VCRPCN was able to achieve a mean average precision (mAP) of 78.6% on animal detection, on five animal categories and one empty category. A fully-automated workflow does have its benefits when low-cost, high-volume systems are requested where the accuracy and detection have less consequences. Despite this, the mAP score highlights that further improvements are necessary for models to achieve sufficient accuracy for autonomous operation. Generally, due to the occasional poor quality images captured by trap cameras, there is a good chance of encountering a few bad classifiable images that require human intervention.

Semi-automated workflows present a more achievable solution towards wildlife image classification. A good example of a semi-automated workflow is proposed in Böhner et al. [5], their workflow processes raw images, and classifies them with the ability to quality check the image labels manually. Celis et al. [6] apply this as well. Miao et al. [7] build upon this further and give the user the power to check low-confidence predictions flagged for human annotation. Brook et al. [8] take a different approach to the semi-automated workflow. They have created a modular system for non-programmers to perform image-data preparation and AI-model training pipeline through a command line

interface. This system offers flexibility, enabling users to execute any part of the workflow individually, or execute more or all parts at once.

DATA PRE-PROCESSING

The first step in a classification pipeline is pre-processing the dataset. As Reynold Xin [9] and many others have said, “A machine learning model is only as good as the data it is fed”. Which showcases the importance of pre-processing. Pre-processing techniques aim to address common challenges such as class imbalance, image noise, variability in image conditions and artefacts, resulting in a more effective and accurate model. Islam and Valles [10] argue to start off with preprocessing the data with formatting, cleaning and sampling the dataset to enable the classification algorithm to retrieve information from the data. The formatting of the dataset is necessary to make the data suitable for the deep learning model. Cleaning data is done to eliminate bias and irrelevant information, but also entails splitting datasets in training and test sets as Sharma et al. [11] notes. Data sampling a smaller, random representative sample can be an option if the computational time and memory requirements are limited (Islam and Valles [10]). Building on these first steps of data pre-processing, data sampling can further influence the performance of image classifications models.

CLASS IMBALANCE

The most common method of tackling class imbalance is through data augmentation. Faizal and Sundaresan [12] and Sharma et al. [11] noticed a significant class imbalance between the classes which could lead to biases in the model. Both augmented the underrepresented classes, for instance Faizal and Sundaresan [12] apply data augmentation through rotating, flipping and zooming the samples to increase the number of class images. Similar techniques are also used by Simões et al. [13] and Islam and Valles [10]. However, Simões et al. [13] observe that regardless of the data augmentation they applied, the ‘human’ class is over-represented in comparison to the other classes. Indicating that data augmentation in their case was not sufficient to prevent heavy biases for the ‘human’ class. The consequence is that the model will overfit and will cause the model to work ‘too well’ on the training data, but fails to generalise on validation data. Schneider et al. [14] demonstrate a different interpretation of data augmentation resampling. They made use of a so-called data generator that samples an equal number of images for each class

in each training epoch. This method ensures that species with a limited amount of data receive relatively higher sampling rates. Sajun and Zualkernan [15] attempt a different way of dealing with data augmentation. They try to over- and undersample based on the class weight assigned from the imbalance distribution. Consequently, this is only feasible if each class has a workable number of images to allow for meaningful sampling. In conclusion, while data augmentation serves as a valuable tool in mitigating class biases and enhancing the models generalisation abilities, a critical look is important to notify of signs of under- or overfitting of the model at hand. Fortunately, additional techniques such as image enhancement can further improve the effectiveness of image classification models.

IMAGE ENHANCEMENT

There are a lot of image enhancement techniques. Most of these techniques focus on noise removal. Islam and Valles [10] point out that frequent noise problems such as Gaussian, salt-and-pepper and speckle can evoke performance degradation which can be caused by unfavourable temperatures, poor illumination or noisy transmission causing sensor noise. The standard method of image-denoising is to clear the image by boosting its edges and outlines and suppressing the details, which will result in noise suppression. Chen et al. [16] and Zurita et al. [17] demonstrate image removal techniques through similar methods as gaussian blurring, brightness adjustment and colour enhancement methods. Most papers have used one or multiple colour enhancement techniques of the likes described here [18], [17], [19]. The frequency of these used methods still show their practicality and effects on data enhancement.

ADVANCED TECHNIQUES

In addition to these standard noise removal and colour enhancement techniques, advanced methods keep on emerging from the literature. For instance, [19] and [17] make use of the Cutout technique. It uses fixed-size rectangular squares to randomly fill a region, masking irrelevant information which contributes to the generalisation ability of the model. Moreover, Yang et al. [19] take this a step further. They also use CutMix, which is similar to the Cutout technique, but the squares proportion may vary and the real frame image, labelled with the cropped data set images, is utilised for filling. These sections were designed to avoid the main information of the picture, using

the Intersection over Union ratio. The data augmentation pre-process of Yang et al. [19] showed an overall increase of accuracy, recall and mean average precision between 3 and 6 percent. Furthermore, Bhargavi et al. [20] describes an image enhancement technique called Lanczos interpolation that can scale images precisely and improve the image quality of the image overall. This is a promising technique that can help prevent aliasing and sharpness problems in images.

Another ingenious procedure is the dual-input channel method. Applied by Curran et al. [21], It is noted that both papers use a dataset of camera trap images where the camera, when motion-activated, takes three pictures. They describe the method effectively, the method is split in two streams, the first stream applies the structural similarity algorithm to produce artefacts. The second stream selects all sequences that meet the requirements from the first stream. The first stream produces these artefacts based on the differences between these images. Since three pictures are shot at once and the position of the camera is fixed, an animal that moves through the sequence would then be highlighted through this structural similarity algorithm. To avoid pixel shifting, both papers choose to only use the first two images of a sequence. This can be a fantastic method to highlight moving objects in camera trap images. Curran et al. [21] reported a significant improvement on this dual-input channel CNN model compared to a regular ResNet-50 model, demonstrating how data pre-processing can improve wildlife image classification models tremendously.

BACKGROUND SUBTRACTION

A good strategy to improve generalisation of a model is the use of Background Subtraction (BS). For BS, Chappidi and Sundaram [22] and Sá [23] use a similar approach, Celis et al. [6], Schneider et al. [14] and Brook et al. [8] as well, Faizal and Sundaresan [12] used a different method. For instance, Chappidi and Sundaram [22] use segmentation via the Superpixel-Based Fast Fuzzy C-Means (FCM) method to detect wildlife animals, and thus remove the background. Sá [23] shows that substantial improvements were observed across all YOLO versions. However, Sá warns against the blind integration of BS in any model, as their Faster R-CNN model proved. Faizal and Sundaresan [12] propose a segmentation via OpenCV, called K-Means. They found that with a value of K = 3 the background removal was sufficient enough. Finally, Brook et al. [8] uses an effective approach where they make use of the MegaDetector

v5a, a deep-learning object detection model that locates images in four categories: animal, blank, vehicle and humans. MegaDetector can have different use cases for instance animal presence detection, here it is used to retrieve the bounding-box coordinates of the animals in the images. After this has been done, ‘Snip’ is used to isolate the animal and resize the new image. All three show an effective method to remove the background from the animals in order to improve generalisation of the model. Where Chappidi and Sundaram [22] and Faizal and Sundaresan [12] use segmentation methods, Brook et al. [8] uses a deep learning approach (MegaDetector) to do this, which may become even more effective in the future due to its widely usage.

TRANSFER LEARNING

Transfer learning creates a positive effect on image classification systems.
Thangaraj et al. [24] explain this method accurately, describing transfer learning as essentially fine-tuning pre-trained models. Two techniques are used extensively: deep feature extraction and fine-tuning. With deep feature extraction, the low-level features of the pre-trained model serve as a foundation of knowledge without training from scratch. Generally these layers are frozen during fine-tuning. Allowing the final layer(s) to be retrained using a new dataset, with a new output layer based on this new dataset. Of the 29 papers analysed, 14 papers incorporated transfer learning [4], [8], [11], [12], [13], [14], [16], [18], [20], [23], [24], [25], [26], [27]. All have claimed an improvement on one or several levels.

With the basic framework of transfer learning clarified, its utility plays a significant role in enhancing the performance and efficiency of wildlife image classification models. Guo et al. [4] and Gurule [26] demonstrate that a significant advantage of transfer learning is the substantial reduction in training time. Seljebotn and Lawal [25] identify that this has to do with a significant reduction of trainable parameters. Furthermore, Gurule [26] also observes a great improvement of approximately 9 percent increase in Mean Average Precision (mAP). These findings are supported by other studies [11], [12], [13], [27]. This indicates that the backbone model’s ability of complex pattern recognition contributes to the model's accuracy. In conclusion, using transfer learning within wildlife image classification systems leads to an improvement in performance, efficiency and time management.

MODELS

Comparing wildlife classification models is challenging due to the diverse datasets, pre-processing techniques, and pipelines used across different studies. However, comparisons within papers are a bit more robust and can give us a roadmap to which models perform well. For instance, Schneider et al. [14] note that the ConvNeXt models achieved a mAP of 97.91% on a validation set, and perform slightly better compared to other used models, the Swin Transformer and EfficientNetV2. The Resnet models perform worse, but still achieve acceptable results. Interestingly, Brook et al. [8] achieved similar results when various EfficientNet versions were compared to ConvNeXt-Base and ViT-Base. The ConvNext-Base performed slightly better, however this was only seen after the decimal point.

Curran et al. [21] demonstrate that their own convolutional neural network (CNN) using a custom dual-input channel performed well over alternatives like DenseNet-121 and ResNet-50. Highlighting the potential of specialised architectures in certain contexts, since the setup of the camera allowed for a different method. Faizal and Sundaresan [12] and Thangaraj et al. [24] both observed the InceptionResNetV2 model outperformed other models. Faizal and Sundaresan [18] compared the model with models like VGG19, VGG16, MobileNetV2 and V3, while Thangaraj et al. [12] compared InceptionResNetV2 against DenseNet-169, ResNetV2-50 and Xception. All models are pre-trained on the ImageNet dataset, but are fine-tuned on different sets. This showcases the power of InceptionResNetV2. Ansari et al. [27] compared several pre-trained models, such as VGG16, ResNet-50, InceptionV3 and EfficientNetB0. Where EfficientNetB0 achieves the best performance with an F-score of 0.9. Similarly, Sharma et al. [11] affirms this in a different study, where VGG16 is compared to EfficientNetB0 with a top-5 accuracy of over 90 percent. To conclude, it is hard to make actual comparisons between models, since each model has its strengths and weaknesses. However, some of the found models like EfficientNetB0, ConvNeXt and InceptionResNetV2 show potential.

2.3.3 Conclusion, Discussion & Future recommendations

The goal of this systematic literature review was to get an overview of the current developments and practices on wildlife machine learning system pipelines, regarding pre-processing techniques and model types. From the research, it is found that there are indeed a few best practices such as data augmentation types as Background Subtraction, Cutout

and CutMix methods and data enhancement techniques, which can make a difference for a model's effectiveness. Subsequently, transfer learning and specialised architectures show that there is not always a ‘one size fits all’ image classification system. However, the variability of pre-processing techniques and model architectures across the found studies make it hard to compare them and make definitive conclusions about these methods.

A limitation of the research is that the research approach is applied to other papers that have created wildlife image classification systems, without looking for specific papers that do regard a best practices method for a specific problem. For instance, having a paper regarding data augmentation applied to the field of wildlife image systems would give a deeper understanding of techniques and methods that could be used in this field. However, due to the scope becoming too wide this was disregarded. Furthermore, in the found papers there was generally not a lot of attention to comparisons, a lot of the research focused more on one specific technique or issue. This makes it hard to derive conclusions. Finally, having a scope of 5 years in this field (from 2019 onwards) may have contributed to finding 29 papers. Initially there were 40 papers assumed to pass the criteria, but after further examination setting up the Literature Matrix this was discovered. Time constraints made it impossible to do another critical look at the search string. Before this final attempt another Literature Matrix was already created, which also took time.

An interesting future research direction is a search through pre-processing techniques that are not yet used in image classification systems, that may have the power to make a difference on classification’s accuracy. The image processing branch is quite old and may have fruitful solutions and practices for newly specialised wildlife image classification systems.

3 - METHODS AND TECHNIQUES

3.1 CRISP-DM

For this project, an adapted version of the CRISP-DM methodology was used. From the initial systematic literature review, two papers integrated the workflow inside of their project already [15] [23]. CRISP-DM is widely used throughout data science projects, as it offers a clear, structured framework for project execution [28].

3.2 Research Tools

Several tools have been equipped throughout this thesis. For the literature review, the platform ResearchRabbit [29] was utilized to efficiently discover both

cited and referencing works. The model training was done on the University of Twente's Jupyter notebook environment, which provides remote access to several GPU cores, such as the NVIDIA A10 and A16 tensor core GPUs. Perfect for machine learning training.

During the preparation of this work the author used ChatGPT in order to improve sentences and investigate and solve bugs in the code. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

3.3 Evaluation & Metrics

The model's abilities to classify animals will be measured using the following metrics.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (1)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

where

TP = True Positive
FP = False Positive
FN = False Negative
TN = True Negative

4 - DATA UNDERSTANDING

4.1 Data collection

Three datasets have been used for the creation of the final training dataset. The primary dataset is the Boeren, Burgers and Buitenbeesten (BB&B) dataset [1], which consists of over one million raw images (more than 500GB of data) gathered by camera traps in the wild. In addition, an open dataset of European fauna collected by Schneider et al. [14] was included, and further cat and weasel images were drawn from various projects within eMammal [30].

Because the BB&B dataset is extensive and unprocessed, it is important to understand how camera traps operate. Camera traps are equipped with motion sensors that capture a burst of images whenever movement is detected. This triggers both daytime as well as nighttime captures, with an inbuilt infrared sensor for the night. Figure [4.1.1] illustrates an example of a night image of the BB&B dataset.

[4.1.1] Random night image of the BB&B dataset.

Alongside the captured image, each photo contains information regarding temperature, date, time and the specific camera ID (location).

Similar to night images,daytime images are captured in a similar manner, as is demonstrated in Figure [4.1.2].

[4.1.2] Random day image of the BB&B dataset.

However, not all images contain animals. False triggers can be caused by movement in the vegetation, or other environmental factors.

[4.1.3] Random occluded image of the BB&B dataset.

Moreover, bad weather conditions such as fog or extreme cold can significantly affect the image's quality, as seen in Figure [4.1.3].

Given the sheer size of the BB&B dataset, filtering and preprocessing are necessary steps to create a high quality dataset usable for machine learning. The following sections will describe the steps taken to filter, process and finally create a cohesive training dataset.

5 - DATA PREPROCESSING

This phase outlines the step by step process of transforming the BB&B dataset into an annotated dataset suitable for model training.

5.1 Object Detection

As shown in Figure 4.1.2, not all images contain animals, and those that do may vary in quality. This variability can affect model performance.

Previous studies have demonstrated that background subtraction can substantially improve classification accuracy [6], [8], [12], [14], [22], [23], [26]. Accordingly, bounding boxes around detected objects are utilised as the first step in the classification pipeline.

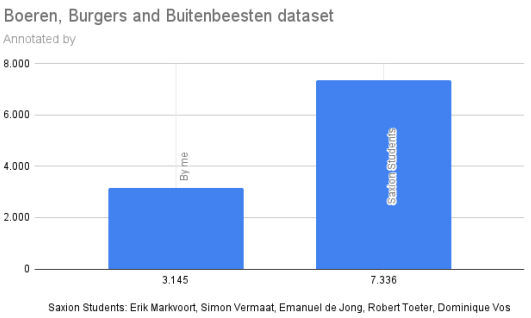
From the literature, an object detection model named *MegaDetector* [31] was identified as a suitable tool [6], [8], [13], [14]. MegaDetector classifies regions within images into three categories - Human, Vehicle, and Animal - and assigns confidence scores for each classification within an automatically generated JSON file, linked to the images. For this project, MegaDetector v5b was employed.

After running MegaDetector on the raw images, a confidence threshold of 0.5 for the “Animal” class was chosen to filter out false positives, and to maintain quality of the classification. According to the model’s creators, a threshold of 0.5 is generally effective at drawing bounding boxes that do contain an animal.

5.2 Annotating images

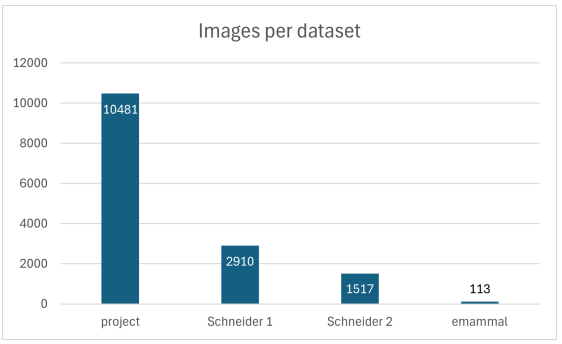
After the object detection step, bounding boxes for each image were annotated using *Label Studio* [32], an open-source data labeling platform. Label Studio allows for usage of .JSON files, thereby facilitating the import and annotation of bounding box data.

Several students at Saxion contributed to the annotation process, the annotations division can be seen down below.



[Figure 5.2.1] Annotation distribution of BB&B dataset.

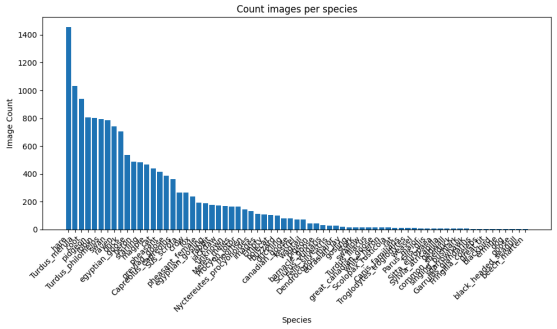
Similar processes in various states have been used for eMammal and Schneiders datasets, resulting in a total of 15.021 annotated images, see Figure 5.2.2.



[5.2.2] Images per dataset.

5.3 Training dataset

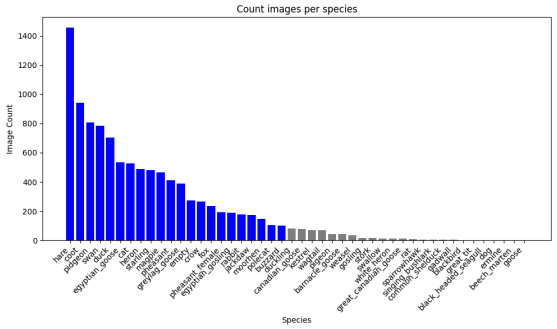
After annotation, a subset of 71 classes remained, including an ‘empty’ class derived from pieces of backgrounds of the BB&B dataset, see Figure 5.3.1. However, species absent from the BB&B dataset, or which are not considered key predators for this study were excluded for the final selection.



[Figure 5.3.1] Annotations per class.

A study by Shahinfar et al. [33] suggests that 150 to 500 images per class can achieve a practical level of accuracy for camera trap models. In this dataset, certain focal species (e.g. buzzard and polecat) had between 100 and 150 images. Given their relevance to the project's objectives, these classes were still included. Consequently, 22 classes were retained, see Figure 5.3.2.

To prepare for classification, background subtraction was applied to the bounding boxes, producing images in a uniform shape of 224x224x3, which matches the expected input dimensions for the classification models discussed in the next chapter.



[Figure 5.3.2] All blue classes are taken into account for the final dataset.

After filtering, 9.859 images remained across 22 classes. The hare class had the highest representation of 1.455 images, whereas the buzzard class had only 100 images. This class imbalance has to be accounted for, to ensure robust model performance.

5.4 Processing of the final dataset

Prior to model training, the dataset was normalized to help the model generalize better and to enhance performance. Because of the pronounced class imbalance, as can be seen in Figure 5.3.2, three measures identified in the literature were implemented.

Oversampling: The first measure is to oversample rare classes. All classes have been oversampled to 500 images, provided they did not already meet this count. While oversampling can balance class frequencies, it may also lead to overfitting if applied excessively.

Class Weights: Secondly, during training normalized class weights were applied to penalize misclassifications of rarer classes more heavily. This approach helps the model avoid bias toward classes with larger sample sizes.

Data Augmentation: Additional transformations were performed "on the fly" during training using the *ImageDataGenerator* of TensorFlow. After experimentation, the following parameters, as seen below, have shown to be working.

```
train_datagen =
ImageDataGenerator(
rotation_range=20,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest'
)
```

[Figure 5.4.1] Data Augmentation during training.

Finally, the dataset was split into 72% training data, 18% validation data and 10% test data. The slightly larger validation set was chosen to ensure sufficient representation of the smaller classes. This strategy helps mitigate potential overfitting and provides a clearer understanding of model performance on underrepresented species.

6 - MODELING

From the systematic literature review covering state-of-the-art methods in the past five years, several machine learning models were identified as potentially useful for this project.

Given the relatively small size of the dataset, transfer learning emerged as the most suitable approach. Models trained through transfer learning leverage weights pretrained on very large datasets, which can substantially improve performance when fine-tuned on a smaller dataset.

In particular, models pre-trained on ImageNet-1000 [34] are frequently mentioned in the literature, for instance by [12], [20], [25], [27]. ImageNet comprises over 1.2 million images across 1000 classes, providing machine learning models with pre-knowledge regarding low and high level image features.

Table 6.1.1 lists the well performing models identified by the systematic literature review. These architectures trained on ImageNet-1000 will serve as backbone models for training new classification models applied to wildlife camera trap images.

Model	Parameters	In paper
InceptionResNetV2	56M	[12], [24]
EfficientNetB0	4M	[11], [20], [27]
ConvNeXt model	87M	[8], [14]
Schneider (ConvNeXt) model	87M	[8], [14]

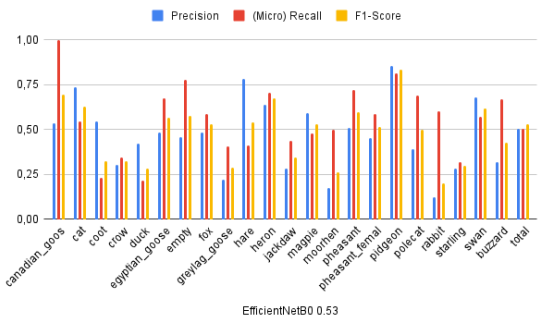
[Table 6.1.1] Selected machine learning models and their respective parameters count, and references.

The fourth model in this table is a ConvNeXt variant provided by Schneider et al. [14] who have released its weights publicly. This is particularly relevant because the model was trained on images of European wildlife, which closely aligns with the species present in our camera trap dataset. This already adapted model may improve the performance on the targeted classes.

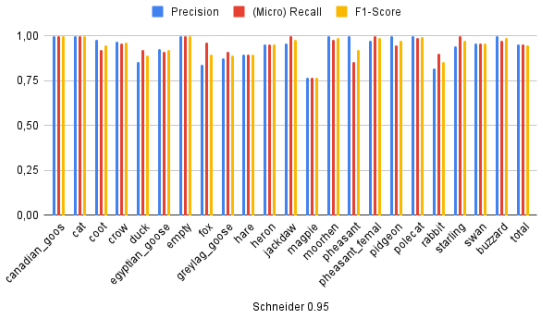
7 - EVALUATION

7.1 Model performance

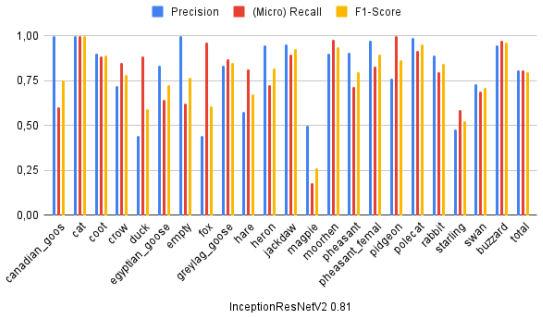
Using the evaluation metrics defined in Chapter 3, the models introduced in Chapter 6 were trained and tested on the refined dataset.



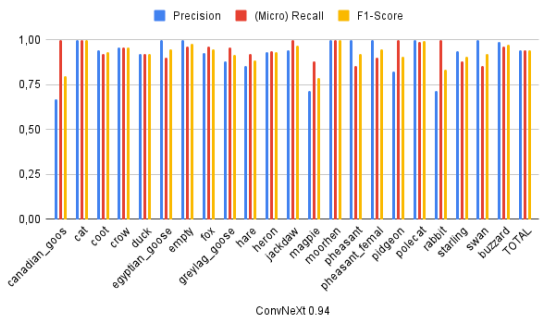
[Figure 7.1.1] Performance of EfficientNetB0 model.



[Figure 7.1.2] Performance of Schneider's et al. ConvNeXt model.



[Figure 7.1.3] Performance of InceptionResNetV2 model.



[Figure 7.1.4] Performance of InceptionResNetV2 model.

From these figures, it is evident that the two ConvNeXt models show a significant performance gap to the EfficientNetB0 model, as well as the InceptionResNetV2 model. The Schneider et al. model consistently shows the highest results, achieving 95.45% accuracy, 95.58% on precision and 94.59 on (micro) recall, as seen in Table 1.1. One possible explanation is that Schneider's model used the ConvNext model trained on ImageNet-1000 as a backbone, and then refined it on more specific wildlife species closer related to those in this thesis, thus leading to better transfer learning outcomes.

By contrast, EfficientNetB0 shows the lowest overall performance. With approximately 4 million parameters, this architecture may not have sufficient depth to understand the data through transfer learning. Presumably, the depth combined with the small training dataset combined with suboptimal use of hyperparameters may have caused this drop in performance.

The InceptionResNetV2 model generally performs well but does not match the accuracy of the ConvNeXt architectures. Moreover, InceptionResNetV2, EfficientNetB0, and to a lesser extent the ConvNeXt models, struggle with certain classes. For instance, small birds like the magpie and starling are often misclassified. This issue could be attributed to the bounding boxes generated during data preprocessing, which may not account for the smaller size of certain animals, leading to low resolution crops. Additionally, animals that appear similar from afar and lack distinctive coloration, such as the magpie and starling, which are both very dark, may have further reduced the model's ability to accurately identify them.

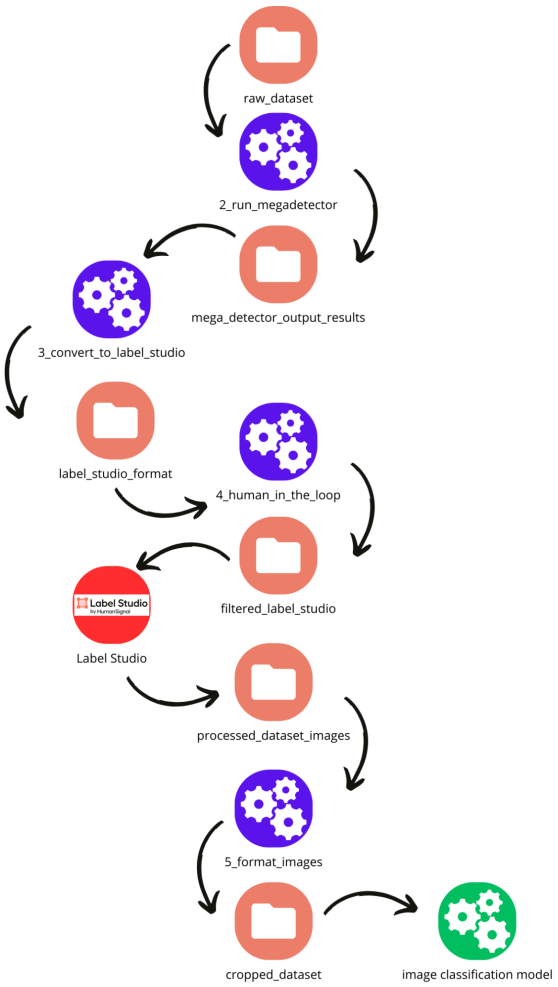
It is necessary to interpret these results with caution, because the test set originates from the same overall dataset used for training and validation. Which are from the same sources. Consequently, there is possibility of nearly duplicate images across subsets, due to the very nature of wildlife camera dataset images. However, preliminary testing on small never before seen images of one of the camera traps indicate that the models can

classify animals and thus show its capability to generalize.

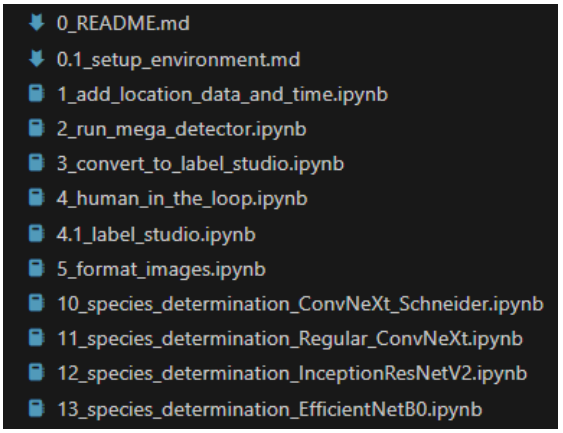
8 - DEPLOYMENT

8.1 Deployment

To facilitate the actual semi-automated pipeline, a dedicated GitHub repository was established. This repository hosts a series of Jupyter notebooks that collectively form an image classification workflow. Users can input their own raw dataset into adjacent folders, which these notebooks process and eventually generate annotated JSON files containing classification results. The modular design of these notebooks provides users with great control over each stage of data processing, and annotation. Figure 8.1.1 shows a good overview of the workflow in Jupyter Notebook.



[Figure 8.1.1] Overview pipeline Jupyter Notebooks.



[Figure 8.1.2] Oversight of the jupyter notebook workflow.

To install and run the workflow, users should set up the environment, using Python 3.10 or 3.11. The dependencies in windows can be installed through pip or conda, specified in *requirements.txt*. Label Studio has to be installed to enable local data annotation. These steps are explained within the GitHub repo.

After executing these notebooks, two notebooks within the folder ‘graphs’ can be utilized to retain metadata of the original dataset and can be mapped back on the cropped images. Now, graphs combining annotations, confidence levels, location, date, and time can be created. Resulting in a quick overview on the output.

9 - CONCLUSION, DISCUSSION AND FUTURE WORK

9.2 - Conclusion

This thesis has presented a semi-automated image classification pipeline for wildlife trap camera images, integrating transfer learning and a Jupyter notebook workflow. By leveraging the ConvNeXt backbone model from Schneider et al., an accuracy of 95.45% was achieved, trained on a relatively small dataset of 9.859 images across 22 classes of which four are primary predators of meadow birds.

A key feature of the proposed pipeline is its ability to map the metadata (e.g., timestamps, date and location) back to the final cropped images. This design facilitates efficient validation of classifications and quick creations of custom graphs, combining the classifications, confidence scores metadata in a quick overview. In addition, each model used in the pipeline has a dedicated notebook, allowing users to configure their own workflows. The resulting system offers a substantial reduction in manual annotation time, as the bulk of image classification tasks can be performed automatically.

9.3 Discussion & Future work

Although the pipeline demonstrates its robust performance, several limitations must be addressed. As discussed in Section 7.1, the training and test subsets both originate from the BB&B dataset, potentially leading to biased performance estimates. Furthermore, the bounding boxes generated by the object detection model were not filtered by their size, increasing the likelihood of poor quality cropped images, particularly problematic for smaller species such as the magpie and starling. Or any animal in the distance.

Moving forwards, several improvements can be made in future refinements. As just

has been mentioned, incorporating bounding box size thresholds could help exclude very small crops, which tend to degrade the models performance. The models utilised in this thesis have an input size of images of around 224x224x3, resulting in significant reduction of quality images, when the bounding boxes are quite small.

Although this thesis demonstrates that small datasets can generate strong results on classification basis, many species remain underrepresented in the current data. Sourcing additional images, potentially of Youtube videos, can help increase the total pool of images. Searching for certain species, combined with ‘camera trap’ result in various lengthy videos with good footage. It might be very beneficial to get into contact with these channels, to use their videos and sample frames into images. Countless hours of footage can be found regarding fauna in this project’s scope.

Additionally, while simple data augmentation methods have improved performance, employing more advanced image generation tools (e.g., generative ai) which could slightly alter images in a way, for instance backgrounds, lighting and morphology altercations, further enhancing model robustness.

Finally, not all species are represented among these 22 target classes. Including an additional “unknown” class, trained on all animals found in the dataset, but not passing the 100 images threshold can be gathered in this one big class, to oppose misclassifications.

By addressing these improvements, the system can evolve into a more accurate, and helpful solution for wildlife monitoring and research in the Netherlands. The findings underscore the value of carefully processing data methods and transfer learning techniques, offering a promising framework for future deployment in ecological research settings.

APPENDIX

model	InceptionResNet V2	EfficientNetB0	ConvNeXt model	Schneider (ConvNeXt) model
batch size top layer	16	32	16	16
epochs top layer	7	8	4	4
learning rate	0.0001	0.00005	0.0001	0.0001
optimizer	Adam	Adam	Adam	Adam
learning rate finetuning (full model)	0.0001	0.000001	0.001	0.0001
epochs finetuning	1	10	1	1
batch size finetuning	16	32	16	16
optimizer finetuning	Adam	Adam	Adam	Adam
Accuracy	0.5051124744376279	0.8068181818181818	0.9442148760330579	0.9545454545454546
Precision	0.5719204398668878	0.84452032957989	0.949892366227274	0.9558172649946199
(Macro) Recall	0,5581901323	0.7918231346265777	0.9442073345441728	0.9458527010060878
F1-Score	0.509977864872993	0.8111006900866636	0.9452870638200039 5	0.9547922796075989
mAP	0.2923799832930706 3	0.6555939533000489	0.8685463579435987	0.8951787089747382

[Table 1.1] The parameters and performance of the best fine-tuned models.

REFERENCES

[1] noardlike fryske walden, ‘Boeren, Burgers en Buitenbeesten’.

[2] V. Zoogdier, ‘Zoogdier vereniging’, Steenmarter (Stone marten). [Online]. Available: https://www.zoogdiervereniging.nl/zoogdiersoorten/steenmarter

[3] J. Vélez *et al.*, ‘An evaluation of platforms for processing camera‐trap data using artificial intelligence’, *Methods Ecol. Evol.*, vol. 14, no. 2, pp. 459–477, Feb. 2023, doi: 10.1111/2041-210X.14044.

[4] Y. Guo, T. A. Rothfus, A. S. Ashour, L. Si, C. Du, and T. Ting, ‘Varied channels region proposal and classification network for wildlife image classification under complex environment’, *IET Image Process.*, vol. 14, no. 4, pp. 585–591, Mar. 2020, doi: 10.1049/iet-ipr.2019.1042.

[5] H. Böhner, E. F. Kleiven, R. A. Ims, and E. M. Soininen, ‘A semi-automatic workflow to process camera trap images in R’, Oct. 07, 2022, *Ecology*. doi: 10.1101/2022.10.05.510927.

[6] G. Celis *et al.*, ‘A versatile semiautomated image analysis workflow for time-lapsed camera trap image classification’, Dec. 30, 2022, *Ecology*. doi: 10.1101/2022.12.28.522027.

[7] Z. Miao, Z. Liu, K. M. Gaynor, M. S. Palmer, S. X. Yu, and W. M. Getz, ‘Iterative human and automated identification of wildlife images’, *Nat. Mach. Intell.*, vol. 3, no. 10, pp. 885–895, Oct. 2021, doi: 10.1038/s42256-021-00393-0.

[8] B. Brook, J. Buettel, and Z. Aandahl, ‘A user-friendly AI workflow for customised wildlife-image classification’, Dec. 12, 2023, *Biodiversity*. doi: 10.32942/X2ZW3D.

[9] R. Xin, ‘Interview with Reynold Xin, co-founder and Chief Architect at Databricks’, “A machine learning model is only as good as the data it is fed”. [Online]. Available: https://devm.io/machine-learning/apache-spark-machine-learning-interview-143122

[10] S. B. Islam and D. Valles, ‘Identification of Wild Species in Texas from Camera-trap Images using Deep Neural Network for Conservation Monitoring’, in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA: IEEE, Jan. 2020, pp. 0537–0542. doi: 10.1109/CCWC47524.2020.9031190.

[11] S. Sharma, S. Neupane, B. Gautam, and K. Sato, ‘AUTOMATED MULTI-SPECIES CLASSIFICATION USING WILDLIFE DATASETS BASED ON DEEP LEARNING ALGORITHMS’, *Mater. Methods Technol.*, vol. 17, no. 1, pp. 103–117, 2023, doi: 10.62991/MMT1996359772.

[12] Sahil Faizal and Sanjay Sundaresan, ‘Wild Animal Classifier Using CNN’, *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 233–239, Sep. 2022, doi: 10.48175/IJARSCT-7097.

[13] F. Simões, C. Bouveyron, and F. Precioso, ‘DeepWILD: Wildlife Identification, Localisation and estimation on camera trap videos using Deep learning’, *Ecol. Inform.*, vol. 75, p. 102095, 2023, doi: https://doi.org/10.1016/j.ecoinf.2023.102095.

[14] D. Schneider, K. Lindner, M. Vogelbacher, H. Bellafkir, N. Farwig, and B. Freisleben, ‘Recognition of European mammals and birds in camera trap images using deep neural networks’, *IET Comput. Vis.*, vol. 18, no. 8, pp. 1162–1192, Dec. 2024, doi: 10.1049/cvi2.12294.

[15] A. R. Sajun and I. Zualkernan, ‘Exploring Semi-Supervised Learning for Camera Trap Images from the Wild’, in *Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference*, Osaka Japan: ACM, Dec. 2022, pp. 143–149. doi: 10.1145/3582099.3582122.

[16] L. Chen, G. Li, S. Zhang, W. Mao, and M. Zhang, ‘YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n’, *Ecol. Inform.*, vol. 83, p. 102791, Nov. 2024, doi: 10.1016/j.ecoinf.2024.102791.

[17] M.-J. Zurita *et al.*, ‘Towards Automatic Animal Classification in Wildlife Environments for Native Species Monitoring in the Amazon’, in *2023 IEEE Colombian Conference on*

Applications of Computational Intelligence (ColCACI), Bogotá D.C., Colombia: IEEE, Jul. 2023, pp. 1–6. doi: 10.1109/ColCACI59285.2023.10226093.
[18] P. Edgar Berry, 'Automated species identification for camera trapping in the Iona Skeleton Coast Trans-Frontier Conservation Area', Namibia University of Science and Technology, 2020.
[19] W. Yang *et al.*, 'A Forest Wildlife Detection Algorithm Based on Improved YOLOv5s', *Animals*, vol. 13, no. 19, p. 3134, Oct. 2023, doi: 10.3390/ani13193134.
[20] I. Bhargavi, A. R. Pratap, and A. S. Sri, 'An Enhanced EfficientNet-Powered Wildlife Species Classification for Biodiversity Monitoring', in *2023 4th International Conference on Intelligent Technologies (CONIT)*, Bangalore, India: IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/CONIT61985.2024.10627148.
[21] B. Curran, S. M. Nekooei, and G. Chen, 'Accurate New Zealand Wildlife Image Classification-Deep Learning Approach', in *AI 2021: Advances in Artificial Intelligence*, vol. 13151, G. Long, X. Yu, and S. Wang, Eds., in Lecture Notes in Computer Science, vol. 13151. , Cham: Springer International Publishing, 2022, pp. 632–644. doi: 10.1007/978-3-030-97546-3_51.
[22] J. Chappidi and D. M. Sundaram, 'Novel Animal Detection System: Cascaded YOLOv8 With Adaptive Preprocessing and Feature Extraction', *IEEE Access*, vol. 12, pp. 110575–110587, 2024, doi: 10.1109/ACCESS.2024.3439230.
[23] C. Filipa Abreu Sá, 'Wild animals detection in Camera Trapping images A Machine Learning approach', NOVA Information Management School Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa, 2023.
[24] R. Thangaraj, S. Rajendar, S. M, R. S. K, S. Sasikumar, and C. L, 'Automated Recognition of Wild Animal Species in Camera Trap Images Using Deep Learning Models', in *2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India: IEEE, Jan. 2023, pp. 1–5. doi: 10.1109/ICAECT57570.2023.10117922.
[25] K. Seljebotn and I. A. Lawal, 'Machine Learning Tool for Wildlife Image Classification', in *2024 9th International Conference on Machine Learning Technologies (ICMLT)*, Oslo Norway: ACM, May 2024, pp. 127–132. doi: 10.1145/3674029.3674050.
[26] M. Gurule, 'The Impacts of Transfer Learning for Ungulate Recognition at Sevilleta National Wildlife Refuge.', University of New Mexico, 2023. [Online]. Available: https://digitalrepository.unm.edu/geog_etds/65
[27] M. Z. Ansari, F. Ahmad, S. Fatima, and H. Shakeel, 'Transfer Learning Framework Using CNN Variants for Animal Species Recognition', in *International Conference on Innovative Computing and Communications*, vol. 731, A. E. Hassanien, O. Castillo, S. Anand, and A. Jaiswal, Eds., in Lecture Notes in Networks and Systems, vol. 731. , Singapore: Springer Nature Singapore, 2024, pp. 601–610. doi: 10.1007/978-981-99-4071-4_46.
[28] Data Science PM, 'What is CRISP DM?' [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/
[29] *ResearchRabbit*. [Online]. Available: https://www.researchrabbit.ai/
[30] 'eMammal'. [Online]. Available: https://emammal.si.edu/
[31] M. Dan, Microsoft, and & more, *MegaDetector*. [Online]. Available: https://github.com/agentmorris/MegaDetector/blob/main/getting-started.md
[32] *Label Studio*. HumanSignal, Inc. [Online]. Available: https://labelstud.io/
[33] S. Shahinfar, P. Meek, and G. Falzon, 'How many images do I need? Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring', 2020, doi: 10.48550/ARXIV.2010.08186.
[34] 'ImageNet'. [Online]. Available: https://www.image-net.org/index.php