Tampere University

JUHA NIEMI

Convolutional Neural Network Based Automatic Bird Identification and Monitoring System for Offshore Wind Farms

Tampere University Dissertations 346

Tampere University Dissertations 346

JUHA NIEMI

Convolutional Neural Network Based Automatic Bird Identification and Monitoring System for Offshore Wind Farms

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty of Information and Communication Technology of Tampere University,
for public discussion in the auditorium 126,
at Tampere University of Technology – Pori,
on 11 December 2020, at 12 o'clock.

ACADEMIC DISSERTATION
Tampere University, Faculty of Information and Communication Technology
Finland

Responsible supervisor and Custos	Prof. Tarmo Lipping Tampere University Finland	
Supervisor	Dr. Juha T.Tanttu Tampere University Finland	
Pre-examiners	Prof. Mohsin Jamali The University of Texas Permian Basin USA	Dr. Panu Somervuo Aalto University Finland
Opponent	Prof. Petri Välisuo University of Vaasa Finland	

The originality of this thesis has been checked using the Turnitin Originality Check service.

Cover design: Roihu Inc.

ISBN 978-952-03-1775-1 (print)
ISBN 978-952-03-1776-8 (pdf)
ISSN 2489-9860 (print)
ISSN 2490-0028 (pdf)
http://urn.fi/URN:ISBN:978-952-03-1776-8

PunaMusta Oy – Yliopistopaino
Vantaa 2020

This dissertation is dedicated to my late labrador Niilo

PREFACE/ACKNOWLEDGEMENTS

The work presented in this thesis has been carried out in the Doctoral School of Industry Innovations (DSII) at Tampere University (TAU), Finland, during the years 2016-2020.

First, I would like to thank my supervisor, Dr. Juha T. Tanttu, for his tenacious effort to start this project initially, providing me the opportunity to do this research, and his help throughout this project.

I would also like to thank my second supervisor, Prof. Tarmo Lipping for his constant support during my work at Tampere University. It has been a pleasure working with the current and former colleagues in the Data Analysis Optimization group.

Further, I thank Suomen Hyötytuuli Oy for providing all the resources needed to accomplish this interesting and absorbing real-world research project.

Finally, I want to express my deepest gratitude to my wife Anna for all her support throughout this entire process.

ABSTRACT

Collisions between birds and wind turbines can be a significant problem in wind farms. Practical deterrent methods are required to prevent these collisions. However, it is improbable that a single deterrent method would work for all bird species in a given area. An automatic bird identification system is needed in order to develop bird species level deterrent methods. This thesis describes the first and necessary part of the entirety that is eventually able to monitor bird movements, identify bird species, and launch deterrent measures.

The objective of this thesis is twofold: it has to detect and classify the two key bird species, and secondarily to classify maximum number of other bird species while the first part still stands. The system consists of a radar for detection of the birds, a digital single-lens reflex camera with a telephoto lens for capturing images, a motorized video head for steering the camera, and a convolutional neural network model trained on the images using a deep learning algorithm for image classification.

Imbalanced data are utilized because the distribution of the captured images is naturally imbalanced. Distribution of the training data set is applied to estimate the actual distribution of the bird species in the test area. Several architectures were tested on species identification and the best results were obtained by the image classifier that is a hybrid of hierarchical and cascade models. The main idea is to train classifiers on bird species groups, in which the species resemble more each other than any other species outside the group in terms of morphology (colouration and shape).

The results of this study show that the developed image classifier model has sufficient performance to identify bird species in the test area in the offshore environment. When the hybrid hierarchical model was applied to the imbalanced data sets, the proposed system classified all of the white-tailed eagles correctly (TPR = 1.0000), and the lesser black-backed gull achieved a classification performance of 0.9993.

CONTENTS

List of Figures

ABBREVIATIONS

AdaBoost	Adaptive Boosting
AI	Artificial intelligence
ANN	Artificial neural network
API	Application programmable interface
AUC	Area under the curve
CIE	Commission internationale de l'éclairage
CMFs	Color matching functions
CNN	Convolutional neural network
DMS	Degrees, minutes, and seconds
DSLR	Digital single-lens reflex camera
DT	Decision trees
ECN	The energy research centre of the Netherlands
FF	Full frame
FN	Number of false negatives
FPR	False positive rate
Haar-like	A technique for extracting features from images, used in object recognition
HOG	Histogram of oriented gradients
HSR	Horizontally scanning radar
IP	Internet protocol
LAN	Local area network
LRDP	Learning rate drop period
LED	Light-emitting diode
LRDS	Learning rate decay schedule
LSE	Least squares error
LSM	Least squares method
ML	Machine learning
ReLU	Rectified linear unit

RF	Random forest
RGB	Red, Green, and Blue
ROC	Receiver operating characteristic
SCADA	Supervisory control and data acquisition
SLR	Single lens reflex
SVM	Support vector machine
TCP	Transmission control protocol
TP	Number of true positives
TPR	True positive range
UDP	User datagram protocol
VSR	Vertically scanning radar
WGS84	World geodetic system 1984

ORIGINAL PUBLICATIONS

This thesis is based on the following publications that will be referred to as publications [**P1**] to [**P5**]:

[**P1**] J. Niemi and J. T. Tanttu. Automatic bird identification for offshore Wind farms: a case study for deep learning. ELMAR-2017, *59th IEEE International Symposium ELMAR-2017.* 2017. DOI: 10.23919/ELMAR.2017.8124482.

[**P2**] J. Niemi and J. T. Tanttu. Automatic Bird Identification for Offshore Wind Farms, In Wind Energy and Wildlife Impacts. *Wind Energy and Wildlife Impacts.* Ed. by R. Bispo, J. Bernardino, H. Coelho and J. L. Costa. ISBN 978-3-030-05519-6. 2019, 135–151.

[**P3**] J. Niemi and J. T. Tanttu. Deep Learning Case Study forAutomatic Bird Identification. *Applied Sciences* 8.11 (2018). DOI: 10.3390/app8112089.

[**P4**] J. Niemi and J. T. Tanttu. Deep Learning Based Automatic Bird Identification System for Offshore Wind Farms. *Wind Energy* 23.6 (2020). DOI: 10.1002/we.2492.

[**P5**] Niemi, J., Tanttu, J.T. (2019). J. Niemi and J. T. Tanttu. Deep Learning Case Study on Imbalanced Training Data for Automatic Bird Identification. *Deep Learning: Algorithms and Applications.* Ed. by C. Shyi-Ming and W. Pedrycz. Springer-Verlag, 2019. DOI:10.1007/978-3-030-31760-7.

1 INTRODUCTION

This doctoral study is closely related to the first offshore wind farm in Finland. The authorities are concerned about the possible bird mortality caused by the constructed wind turbines of the height of 130 m. This has resulted in explicit statements in the environmental license, which obligate the operator of the wind farm to monitor bird movements in the area, and to mitigate, or prevent if possible, collisions between birds and the wind turbines. The authorities have announced two key bird species for particular monitoring in the area: the white-tailed eagle (*Haliaeetus albicilla*) and the lesser black-backed gull (*Larus fuscatus fuscatus*). This demand requires an automatic bird identification system to be developed prior to any measures can be launched to monitor, and to deter, the birds in the area. The alternative would be manual observation by humans which is expensive and inaccurate. The proposed system for controlling wind turbines in its entirety consists of four components: a separate radar system, a bird identification and control unit, a camera unit, and a separate supervisory control and data acquisition system (SCADA). The proposed system is depicted in Fig 1.1. This study is for developing and implementing the bird identification and control unit, and the camera unit.

A deterrent method is any technique that prevents birds to approach wind turbines, e.g. by intimidating them so that they change their current trajectory to avoid collision with a wind turbine. Many different methods such as sounds and various light sources have been applied in land based wind farms. However, these techniques are always applied to all birds species in a given area without testing a different deterrent method to different bird species or species groups [3]. An obvious reason for this is that no feasible method for bird species identification automatically has been proposed.

A pilot wind turbine was erected in the wind park in 2010 for measuring wind and weather conditions. At present, a radar system controls directly the pilot wind turbine and shuts it down when any bird flies into a perimeter of 300 m to the wind

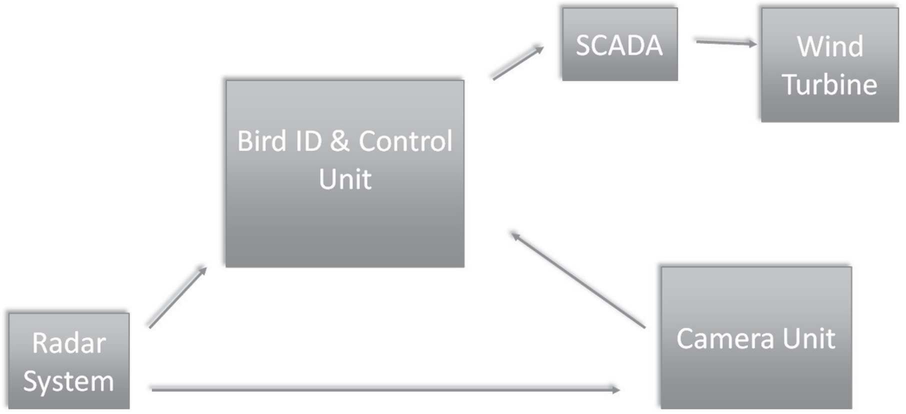


Figure 1.1 The proposed system for controlling wind turbines in its entirety.

turbine, which is the minimum distance in terms to have sufficient time to shut down the turbine. The number of fast restarts of wind turbines should stay as low as possible due to wearing of the mechanics and therefore this operation should be used as a last resort. Figure 1.2 shows the safety zones defined for each wind turbine. A solution to this is a suitable deterrent method but it is difficult to find only one applicable deterrence for all bird species in the wind farm area. According to [3], one extra problem is that the breeding birds may quickly become accustomed to, e.g., sounds as a deterrent method. At first stage, an automatically operating bird species identification system is needed in order to be able to develop such deterrent system. It makes sense, in order to implement this system cost-effectively, to build only one control system in such a location from where it is possible to monitor birds in the vicinity of all wind turbines of the wind farm.

The radar system used in this application is capable to detect birds and pass parameters such as WGS84 coordinates of the detected object. The radar system can also classify detected objects into five size categories. However, it is known that its actual identification capacity is limited, rendering impossible to classify bird species any further merely by this radar system [10, 54]. Obviously, external information is required and a conceivable method is to exploit visual camera images. In this work, we have used a digital single-lens reflex (DSLR) camera and 500 mm telephoto lens for capturing images. The camera has a sensor size of 5472 × 3648 pixels. This number of pixels is enough to constitute the bird in an image, when the image is taken from a long range.

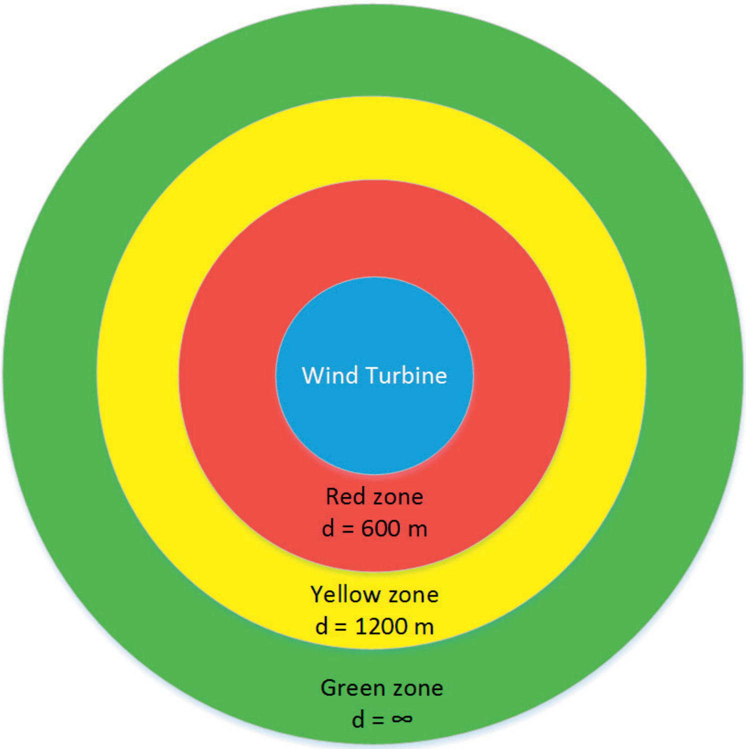


Figure 1.2 Defined safety zones for wind turbines.

1.1 Objectives

The objectives of this thesis were largely raised from possible negative impacts to the birds in the area. The general objective is to develop a system that is able to identify bird species in flight automatically in the offshore environment. This is especially important to Suomen Hyötytuuli Oy, which is the operator of the first Finnish offshore wind farm. In addition, this study is of great interest to wind farm operators in general. The thesis is also important to Robin Radar Systems B. V., which is the supplier of the radar system used in this study. Results of this thesis can be used as such in marine environments, or they can be generalized and utilized in various other kinds of environments. The specific objectives of this study are:

1. to develop real-time algorithms for the two key bird species identification
2. to develop real-time algorithms for bird species identification generally while the first objective still holds

3. to develop a system for detecting birds in flight
4. to develop a system for automatically capturing images of birds in flight.

The first two items in the list can be recognized as classification problems. Methods to identify bird species are based on vocalization and morphology. Morphology includes shape, structure, colour, pattern, and size [38, 48, 49]. Because birds will be monitored from a long distance, morphology remains the only feasible method applied in this study. The solution for the third problem is a radar system but it can only detect flying birds, and thus it is not a solution for the first two problems. A feasible method to study the morphology of bird species in the test area is collecting visual images of them. The first problem becomes an image classification problem. This thesis studies deep learning models and convolutional neural network (CNN) particularly as a solution to this image classification problem. The last problem is two-fold: how to aim the camera to the target and how to capture the image with no human involvement. Modern DSLR cameras are capable of taking images without human touch, thus solving the second part of the automatic image capture problem. A solution for the first part of the problem is a motorized video head, which can be remotely steered and controlled by computer software. The radar system provides the position of a flying bird to the steering software.

The long term objective, beyond this thesis, is to develop a deterrent system that operates at species or species group level, i.e., a different deterrent method for gulls and eagles. A species group could be composed of merely gull species, for example, and this group would be treated as a single class.

1.2 Publications and Author's Contribution

In publications [P1 - P5], the author carried out all of the work apart from supervision and review, which were carried out by Juha T. Tanttu.

Automatic image collection [P2] [P4]

Detection of flying birds is solved by a radar system, which provides WGS84 coordinates of a target bird. Automatic image collection requires a system that is able to aim the camera to the target bird when its location in WGS84 coordinates is known.

The motivation of publication [P2] was to propose a system that can automatically collect images of flying birds. The system consists of the separate radar system, a motorized video head, and a SLR camera with a telephoto lens. This paper also combines parameters, provided by the radar, with the image classification. In publication [P4] the final version of the proposed system is addressed with details of the aiming problem. To our knowledge, these are the first published papers on automatic bird identification implemented by aforementioned equipment.

Image classification [P1] [P3] [P5]

In previous studies, bird identification has been based on morphology and vocalization. Vocalization is difficult to record, and even to detect, in offshore environment. In addition, birds can be silent for undefined period of time. Hence, morphology is the only feasible method to identify bird species offshore. Morphology can be examined from images, and thus this makes the problem an image classification problem. The motivation of these publications were to develop a robust image classification algorithm for real-world images. Publications [P1] and [P3] were based on a CNN with a SVM classifier on the top. Balanced dataset were applied to these classifiers. Their classification performance was acceptable, but they could not make use of the classes with the smallest number of data examples. A data augmentation algorithm was also proposed in the publications [P1] and [P3]. The proposed algorithm converts images into several different color temperatures and also rotates them randomly. Several papers have been published on image classification by CNN, but to our knowledge, these are the first published papers using real-time images of wild birds in flight as the input data. Publication [P5] was motivated by applying imbalanced data sets for training classifiers. A hybrid model of hierarchical and cascaded models was developed. This model consists of several classifiers, which are based on the same CNN architecture. The SVM classifier that was used in previous classifiers was omitted, because it did not increase the classification performance, but increased the training time of the classifiers. The hybrid model uses thresholds to determine the acceptable probability for correct classification. These thresholds are based on the statistics of the collected image data sets of bird species in the test area. The classification performance of the hybrid model is better than the previous two models, and it is also able to classify the classes with the smallest number of data examples.

Papers have been published on image classification using hierarchical and cascaded model, respectively, but to our knowledge, no papers have been published on image classification using the hybrid model, which is also boosted by thresholds gained by the statistics of training data.

2 BACKGROUND AND LITERATURE OVERVIEW

At present the impact of wind turbines on birds is assessed using visual observations, which is often unreliable. Also the estimation of flight trajectories from the visual observations is very difficult. A need to have more detailed information about behaviour and actions of different bird species in the turbine area is obvious. The real number of bird strikes is not known in the existing wind farms in Finland. Actually, no reliable way to measure or even estimate the number of strikes offshore is available. So far no integrated system has been developed for measuring the individual bird flight trajectories, and identifying, if possible, the species in question, have not been developed. A research group at the Univ. of Toledo, Ohio, has developed a prototype system integrating radar, infrared, and acoustic information [42]. This system was able to identify a limited set of bird and bat species mainly based on their vocalizations and also estimate the flight trajectories in 3D by fusing the infrared and radar data.

2.1 Bird Collisions and Mortality

Bird collisions are considered to be one of the major risks of wind farms. The aggregate wind turbine's impact on birds consists of disturbance, barrier effects, and habitat loss as well as collision risk. The consequences of bird collisions might have direct effect on the local breeding population depending on the level of mortality [18].

The actual number of birds killed by collisions with wind turbines in a certain area is not available, mainly because of lacking a reliable method to measure it automatically. Nevertheless there are studies providing an estimate varied according to area and species. The estimated numbers lie between 0-68 birds per turbine per year [16, 18, 30, 37]. The number of collisions varies owing to the season as the flux (number of flight movements per hour per km in a given area) alters accordingly. The higher flux results in a greater number of the collisions, which has been formulated as follows: collision rate = collision risk × flux [30].

Most of the studies have been conducted in the onshore environment and therefore they cannot be applied directly to the offshore environment. In case of the offshore turbines, bird populations consist of different species, and therefore bird behaviour is different and, as a result, collision rates probably differ from the land based turbines. At present, only little data are available on actual collisions with offshore turbines. However, some efforts have been made to develop a collision model and perform collision related probability calculations based on a specific tern population [13, 14, 40]. Inevitably, collisions occur offshore as well as onshore but the actual collision rate of the offshore wind turbines is still unknown and the onshore estimations of mortality are only directional.

2.2 Monitoring Collisions

The characteristic feature of bird collisions is that they are infrequent alternating with the season and time of day. Collision probability is higher in migration and breeding seasons. Bad weather (low visibility and high winds) increases the risk of collision [6]. A remote technique for collision monitoring is required. Publications on monitoring collisions manually and systematically in the offshore environment are not known, and collecting corpses is not a real possibility at the sea. As a result, improvement of the methods of measuring collisions offshore is obvious [9, 15].

No automated technology to measure collisions exists at present, and the developed collision risk models are based on land operating wind turbines [2]. The direct and actual recording of bird collisions is essential in order to develop a deterrence system and collect relevant statistics. The tools developed for direct measurements have to be able to deal with strong winds, salt water, and noises from the mill structure that have to be filtered.

A better understanding is needed of the avoidance behaviour used in the collision risk models in dominant weather conditions. The avoidance behaviour is twofold: the micro-avoidance, which concerns birds close to individual turbines, and the macro-avoidance, which concerns avoidance behaviour around the entire wind farm.

Of course, direct measurement of the collisions, if possible, will provide information without the uncertainty associated with collision risk models.

Systems for monitoring bird collisions at offshore wind turbines should be able to count actual collisions and identify the species at least at the species group (genus) level. They should be able to tell the difference between a gull and a waterfowl, for example. Flight activities through the wind farm area occur also at night and in poor weather conditions with low visibility especially during migration periods. Therefore the monitoring system must be able to operate with and without daylight. Since the collision rate varies within a wind farm and with the time of the year [7, 44], the collision data should be collected from all turbines during the year. The conditions at sea are often severe causing the visits to the wind farm to be difficult and expensive. A solution for that is a remote control of the monitoring system. In addition, if the number of collisions is needed to compare to the number of birds flying through the wind farm area, the flight intensity (flow) of birds/bird groups/species through the wind farm area has to be measured and not only the rate of collisions.

2.3 Sensors

Recording a bird collision with a wind turbine in the offshore environment is currently based on visual observations. The techniques used in the onshore environment, such as collecting bird remains, are not feasible in the offshore environment, because no remains are usually found. Therefore, the focus should be on automated technologies that require no manual detection of collisions. However, the need for monitoring the total bird flow through the wind farm area causes the necessity to record the visual observation data as well.

Sensors can be divided in two groups: contact and non-contact sensors. Contact sensors consist of accelerometers and fibre optics sensors. Contact sensors, such as accelerometers and piezoelectric sensors, are sensitive to vibrations and the hardware needed to be mounted on rotor blades is generally not acceptable. Non-contact sensors are commonly acoustic sensors or microphones, of which the most feasible sensor type is the acoustic sensor [50].

The main technologies used to detect collision are radar, acoustic sensors, thermo graphic (infrared) camera, visible light camera, and video camera.

2.3.1 Radars

Radar stands for radio detection and ranging. Electromagnetic waves are emitted (via antenna) usually in pulses. If a layer of medium with different dielectric constant compared to its environment is encountered by the waves, a part of the pulse energy is scattered. Only minor fraction of the scattered radiation is reflected back to the radar and detected by the radar antenna.

There are different ways to classify commercial radars. The radar operating frequency range can be subdivided into frequency bands, with the most frequently used radars in ornithological studies operating in the X-band (3 cm; 8–12.5 GHz), S-band (10 cm; 2–4 GHz) and L-band(23 cm; 1–2 GHz). The peak power output differs according to the strength of the radar signal (usually between 10 kW and 200 kW), which determines the operational range for a given target size. Radars are usually divided into three groups based on their operation purpose: surveillance radar, Doppler radar and tracking radar [13].

Surveillance radars can be used as marine radar, airport surveillance radar or weather surveillance radar. These are characterized by a scanning antenna often shaped as a 'T-bar' or as a parabolic disc (conical or pencil beam). Surveillance radars can be used to map the trajectories of moving targets and the echo trail feature makes each echo visible for a given period of time. Low-powered surveillance radars can detect individual birds (size of ducks) within a range of a few kilometres and flocks of birds within a range of 10 km [13].

Doppler radars have the ability to detect small differences in target position between consecutive pulses of radiation, and generate information on the velocity of the target [13].

Tracking radars are made mainly for military purposes and can only track a single object at a time. They often have a high peak power output, heavy structure and they operate in the X-band. Usually the air space has to be scanned manually before locking the radar on to the target. Automated scanning for targets is also possible, and in this mode the radar locks to the target and follows it [13].

In bird studies, surveillance radar is mostly used for studies at offshore wind farms [13, 29, 30]. A fixed-beam radar directed vertically is used to measure the altitude of the migrating birds, and a surveillance-type radar is used to examine the geographical patterns of movements (the trajectory of a flying bird) [50].

The detection range of flying birds varies with the radar power, format, and even software. Radars are operational without day-light but the detection might be disturbed by moisture that certain weather conditions might generate [13, 69].

The analysis of the data collected by radar requires expertise to filter false echoes from the data. These false echoes are commonly called the clutter in radar technology. Also, the potentially vast amount of data causes another analysis problem. At present, the echoes cannot be separated at species level or not even family level and the number of individuals within a track is not always countable. There are indications that this could be aided by the latest radar technology. The flying speed, wing-beat frequency and object size have been proposed as methods to identify species indirectly [51].

Radar is an excellent tool for monitoring and documenting bird activity, but it is not suitable for automated collision detection, because it is not able to directly monitor and detect collisions. Radar can only detect the presence of a bird in the vicinity of the turbines [50].

2.3.2 Acoustic Sensors

Acoustic sensors (microphones) measure the pressure variations produced by sound waves. Microphones convert the acoustic energy into electrical energy. Acoustic sensors require amplifiers and signal conditioners prior to digitization through an analogue-to-digital converter.

Acoustic sensors seem to be (at present) the most efficient way of detecting bird collisions with the wind turbines. Microphones are also cost-efficient compared to other detection sensors [50]. Field tests have shown that microphones, mounted on the wind turbine, were able to detect the majority of collisions of a 50 g, 7 cm bird [66, 67, 72]. This excludes only small passerine species such as Common chaffinch (*Fringilla coelebs*). False detections, caused by (e.g., mechanical noise and weather) were detected at a rate of 5-10 false triggers per day. The sensitivity of individual systems should be configurable to the existing circumstances, and falsely triggered collisions should be distinguished from the correctly detected collisions [72, 73].

The noise from the rotor blades and other mechanical systems needs to be filtered and the noise will be different for different turbines and under different operating conditions. A high noise level could result in difficulties in detecting small bird collisions [50].

2.3.3 Cameras

There are basically two types of infrared cameras both of which have two different names: active infrared cameras or image intensification cameras and thermal graphic cameras or thermo imaging cameras. The latter type is also called passive infrared cameras. Active infrared cameras detect shorter infrared wavelengths, whereas passive cameras (like thermo graphic cameras) detect thermal longer infrared wavelengths (heat). Active infrared cameras require, in most cases, additional infrared illumination. The heat emitted from an object is detected by thermal graphic cameras, and thus no additional infrared illumination is needed. Active infrared cameras are usually more cost effective with higher resolution than thermal graphic cameras. Visible cameras have higher resolution, and they are less expensive than both of the infrared camera types.

Large birds (over 30 cm in length) can be detected from greater distance with infrared cameras (thermal graphic) than with visible light cameras in conditions of poor visibility [13]. A digital image processing technique based on differencing sequential frames to remove stationary clutter can be used to track moving objects [50].

Video cameras are used for surveillance and monitoring and can offer an excellent visual record of collisions if combined with an automated sensor that detects the collision and starts recording the video [50]. There is obvious limitation; demand for visible light. However, performance can be aided with, (e.g., infrared led lights in poor lighting conditions).

To our knowledge, there are no published papers on digital visible light still cameras applied to collision monitoring.

2.4 WT-bird and DTBird

[72] have developed a method (WT-bird) for detection and registration of bird collisions that is suitable for continuous remote operation onshore. The characteristic sound of a collision is detected by sensors in the blades, which triggers the video registration and sends an alert message to the operator. A prototype has been tested

successfully on a NordexN80/2.5MW turbine at ECN's Wind turbine Test park, Wieringermeer (onshore location) [72]. This implementation is based on monitoring noise, generated from an impact of bird collision with a wind turbine. The collision is detected with microphones, and the noise monitoring is combined with a video camera. The role of the camera is to be able to identify the bird collided with the turbine [72]. Field experiments were carried out to detect the possible bird collisions. These experiments were performed by taking into consideration the small weight of birds compared to the mass of a wind turbine. The experiments consisted of the simulations of bird collisions; small bags of sand with different weights were thrown against the turbine and the tower. Several other turbine generated sounds, different from the bird collisions, were entered to the system as well [73]. The amount of collisions at a single onshore wind turbine was too small for conducting the system calibration during the early field test period. In addition, only one collision was detected in later testing at an offshore location. New camera types of significantly improved image quality were tested, but the image quality was still insufficient to be able to recognize birds during complete darkness. The original objective of this project (a calibrated bird collision monitoring system for offshore) was not generally achieved mainly due to technical problems [71].

At least one commercial system exists: the DTBird developed by Liquen Consultora Ambiental,S.L., Spain [36]. This system is based on video-recording bird flights near wind turbines, and it promises to detect birds automatically and prevent possible collisions in the vicinity of the turbines. However, [41] have evaluated how well the DTBird system is able to detect birds in a wind farm in Norway. They also examined the suitability of DTBird to study near-turbine bird flight behaviour and possible deterrence. They defined the following quantitative criteria: detectability, as measured by the percentage of detected birds by the total number of birds near the turbines, should be over 80%; the number of false positives (video sequences without birds) should be less than 2 per day; the percentage of falsely triggered video sequences should be less than 10 %; the percentage of falsely triggered warnings and dissuasions should be less than 20 %. Their evaluation showed the following results: detectability was over 80 %, the daily number of false positives was below two, the percentage of falsely triggered warnings/dissuasions was circa 50 %, and the percentage of falsely triggered warnings and dissuasions was 40 %. Thus, the DTBird system met the two out of the four evaluation criteria. In addition, the researchers found that the DTBird system enables monitoring of near-turbine flight behaviour, although individual birds usually cannot be identified to the species level, and with the DTBird system collisions may be mitigated [41].

2.5 Bird Species Identification

[55] have studied machine learning (ML) algorithms implemented in marine radars in order to automatically detect and attempt to classify objects. Six ML algorithms have been applied and their performance have been compared. These widely used ML algorithms are: random forests (RF), support vector machine (SVM), artificial neural networks, linear discriminant analysis, quadratic discriminant analysis, and decision trees (DT). All algorithms showed good performance when the problem was to distinguish birds from non-biological objects (area under the receiver operating characteristic (AUC) and accuracy > 0.80 with $p < 0.001$), but the algorithms showed greater variance in their performance when the problem was to classify within bird species of bird species groups (e.g., herons vs. gulls). In their study, RF was the only one that performed with an accuracy > 0.80 for all classification problems, albeit SVM and DT followed closely in their performance. All algorithms correctly classified 86 % or 66 % of the target points when vertical scanning radar (VSR) or horizontally scanning radar (HSR) was used, respectively, and only 2 % or 4 % of the points were misclassified by all algorithms in the respective radar configurations. The results proposed ML algorithms for distinguishing birds from other objects by radar, but classification performance using these algorithms within bird species or bird species groups was poor.

Birdsnap by [5] proposes a solution to the problem of large-scale fine-grained visual categorization, resulting in an on line field guide to 500 North American bird species. Users can upload bird images in the field guide database, and the developed system identifies the images automatically. Researchers introduce one-vs-most classifiers by eliminating highly similar species during training, and they show how spatio-temporal class priors can be used to improve performance. The spatio-temporal class priors are gained from the embedded time and location data that modern cameras include in each image file they produce. Birdsnap uses a set of one-vs-most linear SVMs based on POOFs [4], and it achieved an accuracy of 0.8240 in bird species identification [5].

Time-lapse photography is a technique in which the frame rate of viewing a sequence of images is different than the frame rate of taking the sequence of images. Time-lapse images can make very fast or very slow time-related processes better interpretable to the human eye. Time-lapse images have been used to detect birds around a wind farm by taking images in two seconds interval. [75] have been applied an image-based detection to build a bird monitoring system. This system utilizes a fixed camera and an open-access time-lapse image dataset around a wind farm. The system uses the following algorithms: AdaBoost (Adaptive Boosting), Haar-like feature extraction, and histogram of oriented gradients (HOG). A CNN architecture was also applied to the image classification problem. AdaBoost is a learning algorithm for binary classification, which is developed to improve classification performance by combining multiple weak classifiers into a single strong classifier. These weak classifiers are low performing algorithms (e.g., decision trees with a single split) with error rate slightly under 50 %, i.e., slightly better than a random guess. The idea of AdaBoost is to give more weight to the data points that are poorly classified by the weak learners. The weightings are repeated in each iteration of the algorithm, and finally, by weighted majority voting, the algorithm selects those outputs of the weak classifiers which are combined into a weighted sum that represents the final output of the boosted classifier. As long as the performance of each of the weak classifiers is slightly better than random guessing, the final model can be proven to converge to a strong classifier [19]. Haar-like features are digital image features used in object recognition. In mathematics, the name Haar refers to square-shaped functions which together form a wavelet family. Haar-like is an image feature that utilizes contrasts in images. It extracts the light and the shade of objects by using black-and-white patterns. A Haar-like feature extraction examines rectangular regions by using a detection window to scan an image. It summarizes the pixel intensities in each region and calculates the difference between these sums. The difference is used to segment the image. The position of the rectangles is defined with respect to the detection window, which is used like a bounding box to the target object. In the detection phase of the Haar-like algorithm, the detection window is slid across the input image, and for each segment of the image the Haar-like feature is calculated. Finally, the differences are compared to a learned threshold that separates non-objects from objects. Haar-like features are only weak classifiers [68]. HOG is a feature descriptor used to detect objects in computer vision. A feature descriptor is a representation of an image that simplifies the image by extracting useful information and discarding irrelevant information. A feature descriptor introduces a 2D image as a feature vector. The main idea of HOG is that local object appearance and shape within an image can be described by the distribution of intensity gradients. The image is divided into small connected regions (cells), and a histogram of gradient directions is computed for the pixels in each cell. The descriptor is formed by concatenating the histograms. The HOG descriptor is invariant to geometric and photometric transformations, except for object orientation [12]. [75] found that the best method for detection was Haar-like, and the best method for classification was CNN. The system was tested on two bird functional groups: hawks and crows, and it achieved only moderate performance [75].

2.6 Deterrence

According to a study by [20], everything from fireworks to herding dogs have been tested as a suitable deterrent method for birds in airports. However, they tested red and blue LED lights in their study, and these caused some birds to choose the opposite direction to the lights. A brown-headed cowbird (*Molothrus ater*) was released to fly along a flight path that had been planned in advance. This flight path was equipped with a LED light on one side, and the other side was dark. A single-choice test, in which the bird chooses between a light and darkness rather than between two colors, is ideal for measuring avoidance behaviour. If the bird goes to the dark side, the light used on the other side might be a good candidate for warning birds of danger. The test was repeated with five different wavelengths of light. Birds consistently avoided LED lights of wavelengths 470 nm and 630 nm, which appear blue and red to the human eye. Ultraviolet, green, and white light did not generate any obvious pattern of avoidance or attraction.

Also in airports, introducing a noise net around airfields that emits sound levels equivalent to those of a conversation in a busy restaurant could prevent collisions between birds and aircraft. Researchers set up speakers and amplifiers in three areas around an airfield. Bird abundance was observed over eight weeks, of which the first four weeks without noise, and the second four weeks with the noise turned on. Results showed a significant decrease in the number of birds in the 'sonic net'. This method was particularly effective in deterring starlings (*Sturnus vulgaris*) [64].

3 AUTOMATED BIRD DETECTION AND IDENTIFICATION

In this chapter the hardware and the software, used in the proposed automated bird detection and identification system, are described. The applied methods are also briefly presented. The methods are described in the published papers in more detail. The methods are divided into two categories:

- automatic image collection presented in papers [P2] and [P4].
- image classification presented in papers [P1], [P3], and [P5].

3.1 Hardware

The proposed system consists of several hardware as well as software modules. See Fig 3.1 for an illustration. The radar system for detecting birds is connected to a local area network (LAN). The system has three servers, which are also connected to the LAN: radar server, steering server, and camera control server. A motorized video head and a camera system are connected to the respective server. The work flow is as follows: the radar system detects a target bird and passes its WGS84 coordinates to the video head steering software. The steering software steers the video head into the correct position. Camera control software takes series of images of the target bird and passes the images to classification software, which outputs a prediction of a class (species or species group) of the target bird. The classification software can be operated on a standalone computer such as laptop, or it can be installed as a separate module into the camera control server. For more details about the system as a whole, see publications [P2] and [P4].

A radar system supplied by Robin Radar Systems B.V. is used in this study. In particular, The ROBIN 3D FLEX v1.6.3 model is used, which is actually a combina-

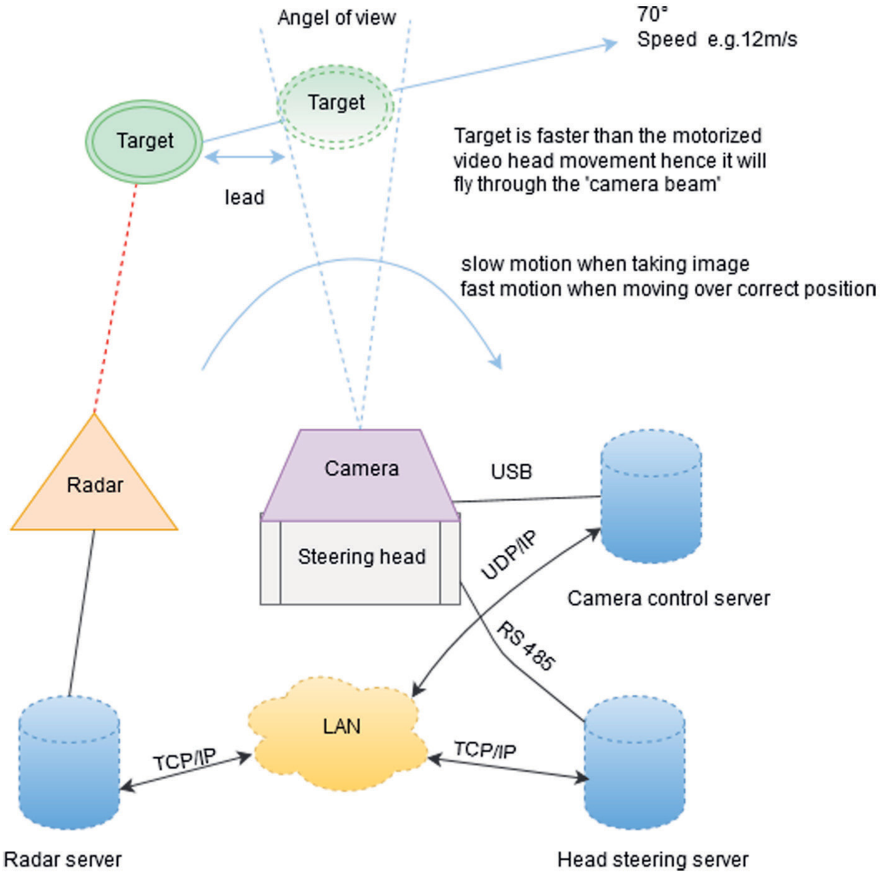

Angel of view
70°
Speed e.g.12m/s
Target
Target
lead
Target is faster than the motorized video head movement hence it will fly through the 'camera beam'
slow motion when taking image
fast motion when moving over correct position
Radar
Camera
Steering head
USB
UDP/IP
Camera control server
RS 485
LAN
TCP/IP
TCP/IP
Radar server
Head steering server

Figure 3.1 The system for automatic image collection.

tion of two radars and a software package for implementation of various algorithms such as tracker algorithms. The PT-1020 Medium Duty video head of the 2B Security Systems is used as the motorized video head. For more details, see publication [P2].

The Canon EOS 7D II camera with 20.2-megapixel sensor and the Canon EF 500/f4 IS lens are used as an image collection system. Correct focusing of the images relies on the autofocus system of the lens and the camera. Automatic exposure is also applied. The operation of the proposed system is not restricted to this combination of the camera and the lens, but a combination of any standard DSLR camera with any standard lens suitable for that camera can be utilized. For more details, see publication [P2].

3.1.1 Automatic Image Collection

The system for automatic image collection is also depicted in Fig 3.1. The automatic image collection is based on an assumption that the given WGS84 coordinates (by the radar system) are accurate enough to enable aiming to a target bird. The WGS84 coordinates are given in decimal degrees with eight decimal places. The motor of the video head has only seven selectable speeds, rendering impossible to track a flying bird at the speed it flies. Thus, the steering software computes a lead point, where the camera should be turned in order to achieve images. Successful image collection is based on constant trajectory of a target bird and the autofocus system of the camera. Here, the constant trajectory means that the flight path of the bird should be invariable enough for only a short period of time.

The radius of the semi-major axis of the Earth at the equator is 6378137.0 m, and the circumference is 40075161.2 m. The equator is divided into 360 degrees of longitude, so that each degree at the equator represents 111319.9 m. This number representing degrees in meters at the equator is multiplied by the cosine of the latitude. This means that the number representing degrees in meters decreases as the latitude increases. Finally, the number is zero when either one of the poles is reached. Longitudes are positive to the east of a prime meridian (i.e., Greenwich, London, a.k.a zero meridian) and negative to the west of it. As the WGS84 reference ellipsoid is applied, one arc minute along a meridian or along the Equator is 1855.3 m [47]. The latitude of the test site is approximately 60°. As the WGS84 coordinates are given with eight decimal places, the precision for the latitude and the longitude is 0.0011112 m and 0.0005556 m, respectively, in the field.

Radar accuracy is measured as range resolution and angular resolution. The range resolution describes how long distance is needed lengthwise between two objects in order them to be detected as two different blips. If the distance between two objects is too short, the two objects will be detected as only one blip. Analogously, the angular resolution describes similar minimum distance between two objects, which are perpendicular to the radar beam [53]. The radar system has actually two radars: a horizontal radar and a vertical radar. The angular resolution of these two radars at a given distance defines a rectangle that can be seen as a 2D resolution cell of the radar system in the given distance. In theory, the detected object can be located anywhere inside of this resolution cell. However, the boundaries of the range resolution and

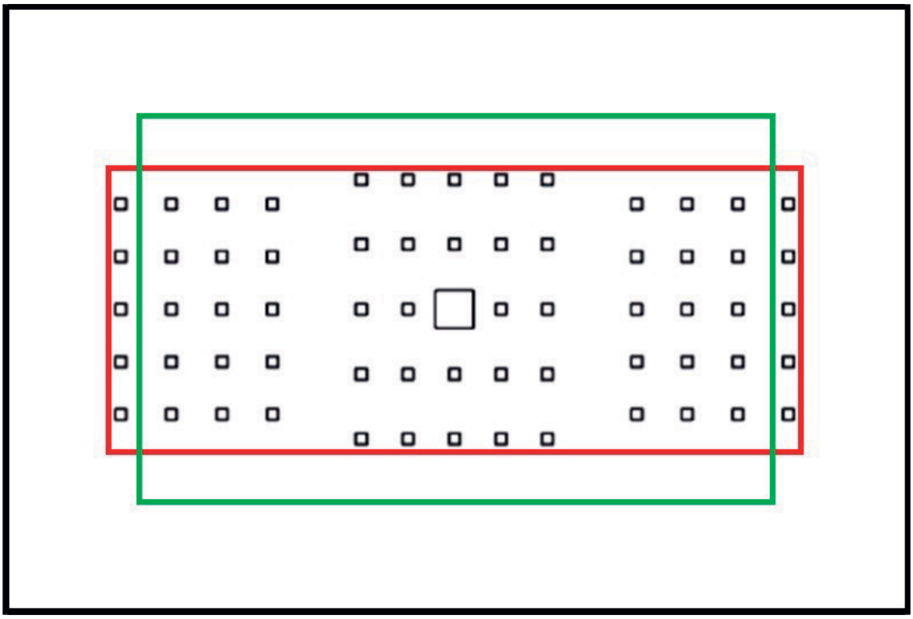


Figure 3.2 Focusing point coverage of the camera frame with a sensor of crop factor 1.6. The focus cell is depicted in red, the angular resolution cell is depicted in green, and the camera frame is depicted in black.

the angular resolution are defined by the 3 dB beam width, i.e., the beam has attenuated to half of its peak value at the boundaries in terms of power [53]. This implies that the probability of object detection is the largest in the center of the resolution rectangle, and it decreases towards the edges.

The frame size at a given distance from the camera can be calculated when the angle of view of the lens is known. The effective frame size also depends on a crop factor of the sensor of a given camera. If a camera with a full frame (FF) sensor is used, the crop factor is 1, otherwise it is expressed by a number greater than one. The reciprocal of the crop factor is used in calculations. However, the rectangular area considered in this doctoral thesis is smaller than the effective frame size because the focusing points of the camera do not cover the whole frame area. The camera frame, its focusing points, and the angular resolution cell at a given distance are illustrated in Fig 3.2. The larger square in the center denotes that the midmost focusing point is currently selected, but all of the focusing points can be selected simultaneously. The rectangle area that covers all the focusing points is called focus cell in this thesis.

The size of the 2D angular resolution cell of the radar system and the size of focus cells at a given distance are presented in Table 3.1. The values in the table for the angular resolution cells are computed as follows:

$$\delta_A = [b_h R \quad b_v R] \tag{3.1}$$

where δ_A is the angular resolution in meters expressed as a vector, b_h is the width of the beam in radians of the horizontal radar, b_v is the width of the beam in radians of the vertical radar, and R is a given distance in meters. The values for the focus cells are computed by the right-angled triangle formed by a given distance and the of view of the lens. The 500 mm lens was used in calculations.

The values for all of the cells are given as 2D, i.e., [*horizontal vertical*]. Focus cell is given for both the FF sensor and for a sensor of crop factor 1.6, respectively. All units in the table are in meters. It can be seen from the table that the horizontal resolution of the radar system is smaller than that of both focus cells, but the vertical

Table 3.1 The sizes of the 2D angular resolution cell and the focus cells at a given distance in meters.

Distance	FF Focus Cell	1.6 Crop Focus Cell	2D Angular Resolution Cell
100	[5.333 1.540]	[3.333 0.962]	[3.141 1.658]
200	[10.666 3.079]	[6.666 1.925]	[6.283 3.316]
300	[15.999 4.619]	[10.000 2.887]	[9.425 4.974]
400	[21.332 6.158]	[13.333 3.849]	[12.566 6.632]
500	[26.665 7.698]	[16.666 4.811]	[15.708 8.290]
600	[31.999 9.238]	[19.999 5.774]	[18.850 9.948]
700	[37.332 10.777]	[23.332 6.736]	[21.991 11.606]
800	[42.665 12.317]	[26.665 7.698]	[25.133 13.265]
900	[47.998 13.857]	[29.999 8.660]	[28.274 14.923]
1000	[53.331 15.396]	[33.332 9.623]	[31.416 16.581]
1100	[58.664 16.936]	[36.665 10.585]	[34.558 18.239]
1200	[63.997 18.475]	[39.998 11.547]	[37.699 19.897]
1300	[69.330 20.015]	[43.331 12.509]	[40.841 21.555]
1400	[74.663 21.555]	[46.665 13.472]	[43.982 23.213]
1500	[79.996 23.094]	[49.998 14.434]	[47.124 24.871]
1600	[85.330 24.634]	[53.331 15.396]	[50.265 26.529]

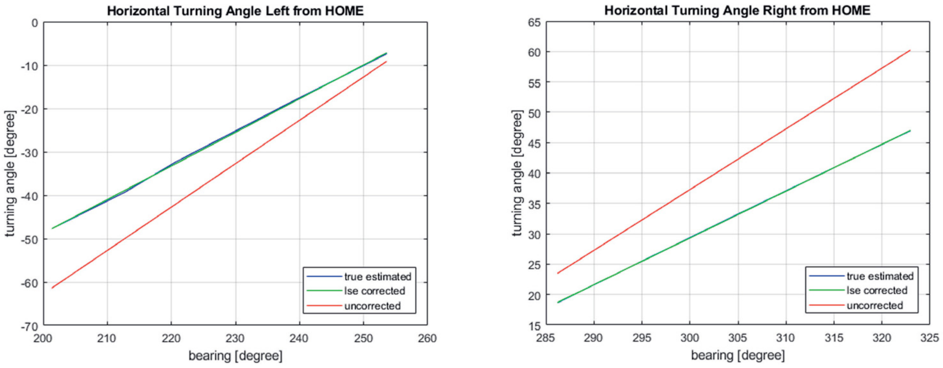
resolution of the radar system is clearly larger than that of the focus cell of the 1.6 crop sensor, and it is also slightly larger than that of the focus cell of the FF sensor. As a result, some of detected objects may be outside of the focus cell if the camera has 1.6 crop factor sensor. The center-weighted probability distribution of the object detection should mitigate this possibility.

3.1.2 Aiming the Motorized Video Head

The video head used in this application has limitations. It cannot be steered by entering the desired horizontal and vertical angles, but it requires the driving time of the motors (separate motors for horizontal and vertical movement). The video head has a fixed home position, which is halfway of the steering range in both directions. The head is installed so that at the home position the camera is horizontally pointing to the west (bearing = 270°), and vertically so that the vertical turning angle at the home position is zero. Tests show that the video head has an increasing error in turning angle towards each steering direction. In addition, this error is significantly larger in horizontal steering than in vertical steering, and it also depends on which direction the head is steered from the home position. As a result, a method for targeting the camera by the head was needed in order to compensate the errors. Locations of the wind turbines in the test area are used as reference locations for error correction, because their positions are fixed and their exact WGS84 coordinates are known. Distances of the wind turbines range from 600 m to 2000 m from the camera location, resulting in relatively large error in meters with only a small error in turning angle.

The least squares method (LSM) was applied to find the angle and offset of regression lines that minimize these errors. This was done separately for horizontal directions left and right from the home position. A constant was used to correct the error in the vertical turning angle, because the error seems to be very small. In addition, the actual vertical turning angle error was obstructed by the erratic flight path (flight path deviates significantly from a straight line) of some bird species, and it was further amplified by the time delay between the timestamp of tracks and the current clock time of the software server. It was more convenient to implement the error correction to the horizontal and vertical turning angles than to the respective steering times, because computations for steering times are based on the turning angles. In horizontal steering, the idea is to find a line that gives a correction to the computed horizontal turning angle when the bearing (a compass direction the head should be pointing at) has been computed first.

The estimate for the true horizontal turning angle for each reference location has been discovered by measuring error in pixels from test images. As the frame size of the camera and the distances are known, error in meters can be computed. These test images are taken automatically by the developed system, and aiming is perfect when the rotor hub of a wind turbine is in the center of these test images. Figure 3.3 shows the estimate for the true horizontal turning angle, the computed horizontal turning angle without correction, and the corrected horizontal turning angle for each reference location, respectively.



(a)

(b)

Figure 3.3 Estimated true, uncorrected and corrected horizontal turning angles for the reference wind turbine locations.

3.1.3 Results of Image Collection

The data structure for detected objects is called a track in the radar system. A track contains the timestamp of a blip concerned, which is the time instant when the blip has been detected by the radar system. Tracks also have the position information of a target; latitude [WGS84], longitude [WGS84], and altitude [m]. Moreover, tracks have the speed [m/s] and the bearing [degrees] of a target. Bearing is a compass point the target is heading to. Successful image collection requires that a target bird has a constant trajectory. Constant trajectory means that the flight path of a target bird should be invariable enough for a certain period of time. The duration of this time period depends on a time delay between the timestamp of a track and the current clock time of the software server. The time delay varies between 2 and 16 seconds. The probability distribution of the delay is shown in Fig. 3.4. From the figure it is apparent that the time interval between 3 s and 4 s has the largest probability, i.e., 30.84 % of the time delays fall in this time interval. More than half (56.13 %) of the time delays fall in the intervals between 2 s and 5 s. When the delay is longer than 5 s, the flight path of a given bird becomes very unpredictable, in terms of aiming, when the prerequisite of constant trajectory stands.

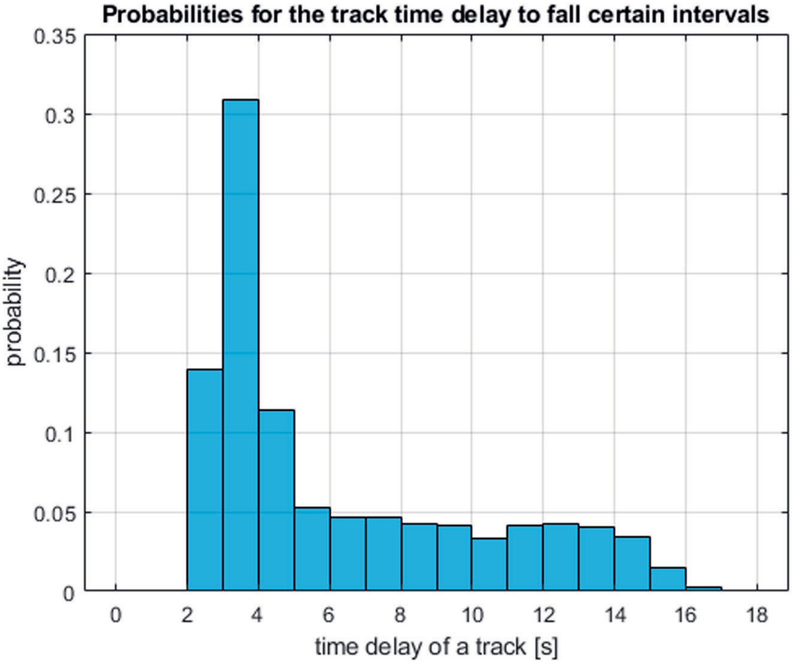


Figure 3.4 The probability distribution of the time delay between the timestamp of a track and the current time.

3.2 Software

All the software needed in this system, excluding the software of the radar system, is developed and implemented by the author. The developed software includes all communication software for various servers (see Fig. 3.1) over TCP/IP and UDP/IP networks, software for steering the video head, software to control the camera, and software for implementing the CNN models. Figure 3.5 shows a diagram of the developed software architecture including the radar system control software for clarity. All commands and data are transmitted via a LAN. The architecture operates as follows: at first, the radar system detects a target bird, and passes the track information, including WGS84 coordinates, to the video head steering software. The steering software controls the video head by computing the vertical and the horizontal turning angles (taking into account the aforementioned error corrections) based on the passed WGS84 coordinates and the altitude of a track. When the head has been steered into the correct position, a release shutter command is transmitted to the camera control software. Then, series of images is taken of a target, and the images with the classify command are transmitted to classification software. The classification software is the implementation of the CNN models, and its results can be displayed on the console of the system and/or they can be transmitted to an external system via the LAN.

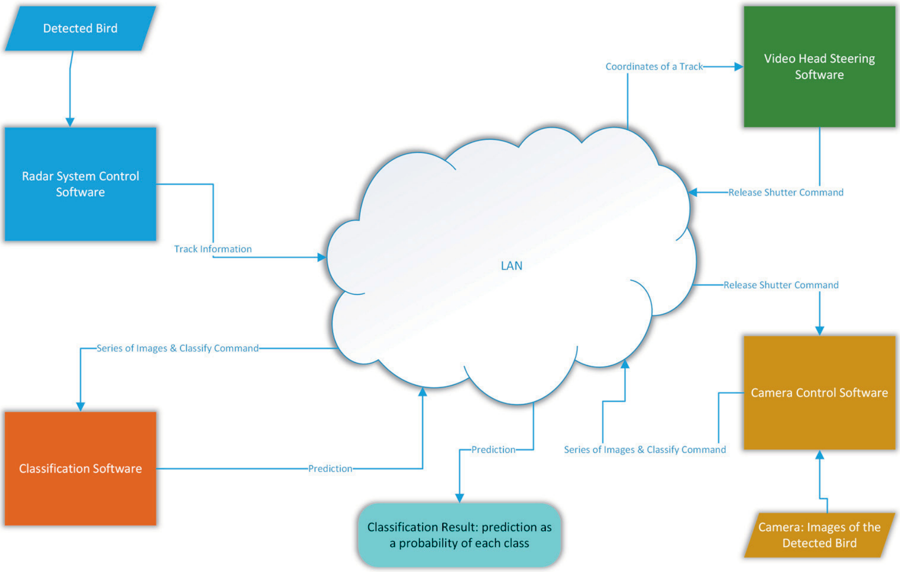

Detected Bird
Radar System Control Software
Track Information
Coordinates of a Track
Video Head Steering Software
Release Shutter Command
LAN
Release Shutter Command
Series of Images & Classify Command
Camera Control Software
Classification Software
Prediction
Prediction
Series of Images & Classify Command
Classification Result: prediction as a probability of each class
Camera: Images of the Detected Bird

Figure 3.5 Diagram of developed software architecture.

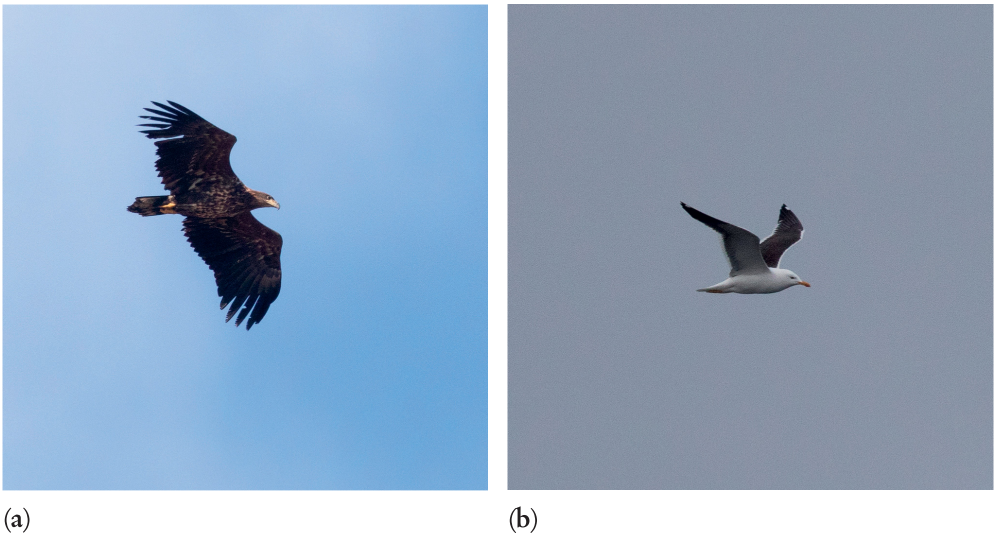


Figure 3.6 Data examples of the white-tailed eagle (3.6a, *Haliaeetus albicilla*) and the lesser black-backed gull (3.6b, *Larus fuscus fuscus*, a.k.a the baltic gull).

3.3 Input Data and Data Augmentation

Data for the classification system are mainly digital images of RGB color model, but information provided by the radar system is also applied. Images were collected manually at the test site at the western coast of Finland. These images were used to train a CNN model for image classification. Figure 3.6 shows examples of images used as data.

Because a large number of examples is required to train the CNN for achieving a sufficient performance as an image classifier, and also because of the difficulties of collecting a sufficient number of images for each class, a data augmentation [28, 70] method has been developed and proposed in publications [P1-P3]. In this method, images are converted into different color temperatures between 2000 K and 15000 K using a step size s. This resembles the natural light at the test site that varies in accordance with cloudiness and humidity [11, 60, 62, 63, 65, 74]. The number of augmented training examples is given:

$$N = [(15000 - 2000)/s + 1]n, \tag{3.2}$$

where N is the number of augmented training examples, and n is the number of

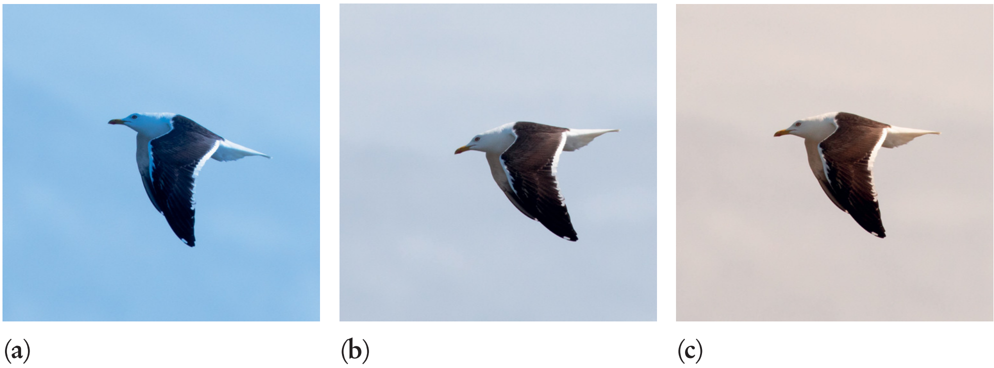


Figure 3.7 Three augmented data examples of a single image of the lesser black-backed gull (*Larus fuscus fuscus*). The color temperature of the images is 3750 K (3.7a), 5750 K (3.7b), and 7750 K (3.7c).

original training examples. When conversion is done, the images are also rotated by a random angle between -20 and 20 degrees drawn from the uniform distribution. Motivation for this is that CNN is invariant to small translations but not rotation of an image [27]. Figure 3.7 shows examples of the output of the data augmentation algorithm.

The radar system provides the following parameters for each detected object: speed, distance, and trajectory. The speed is applicable as it is, and the distance is used to calculate an estimate of the size of an object. The trajectory is a sequence of blips (i.e., successive echoes from an object received by the radar system) of the same object, and all the trajectories are saved into a database. Trajectories are not used in this thesis, because currently the only way to link the species of an object to its trajectory is to identify the species visually by the human eye, and save the result manually into the database.

Images are collected from relatively long distance, thus the number of pixels that cover the object in an image is small, which means that most of the pixels in an image cover only sky. All other pixels, except those that cover the object, are considered as noise, and it is reasonable to crop these pixels as they do not contribute to the classification process. Segmentation is used for cropping the images without loosing any pixels that cover the object. Segmentation is also needed for calculating the size estimate of the object. Fuzzy logic segmentation was applied in publications [P1-P3], but as it showed to be computationally expensive, discrete convolution without a neural network was introduced in publication [P4] as a segmentation method.

3.3.1 Results of Data Augmentation

Figure 3.8 in publication [P3] is reprinted here for convenience, and it depicts the significance of the data augmentation algorithm to classification performance. However, it became clear that beyond some threshold it is useless to augment the original data set any further because of increasing overfitting. The exact value of this threshold as a step size value, s, was not determined. In the figure, the number of the original (without augmentation) data examples was 9312 as a balanced dataset was used.

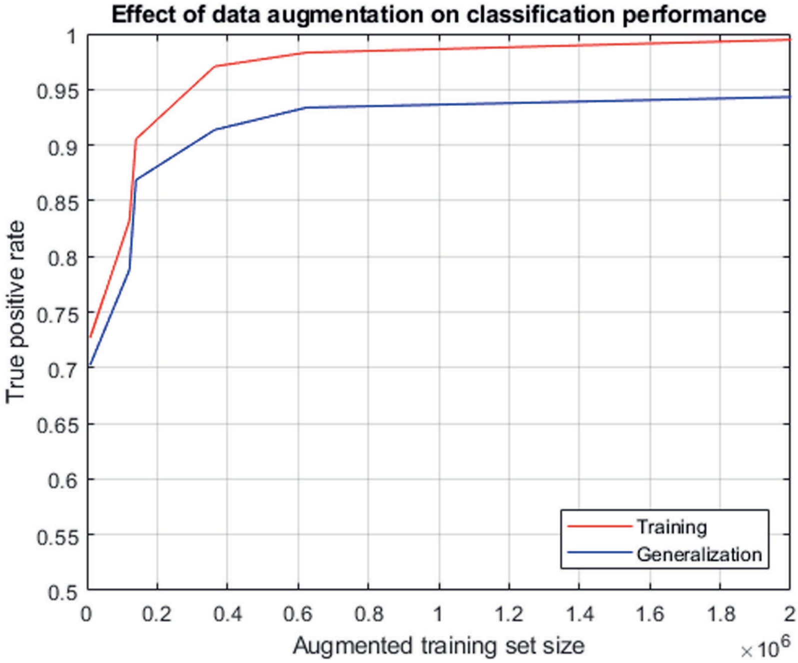


Figure 3.8 The red and blue curves indicate the true positive rate in classification for the training data and the test data, respectively. The details of the classification task and the applied algorithm are given in [P3]. The starting value for both curves is the value when the models were trained on the original data set, i.e., the data set was not augmented.

3.4 Image Classification

Machine learning is a science of making computers learning automatically from a given data and the respective real-world observations, and improve this learning over time autonomously. Machine learning applies models and inference rather than conventional if-else structure. It is one of the main building blocks of artificial intelligence. Machine learning is based on a training data set, which is used to train a mathematical model of these sample data. This approach enables predictions and decisions to be made without them being explicitly programmed to perform the task. Above-mentioned is true only if the dataset is separable in general. Machine learning algorithms are especially used in the applications of computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task [8, 23, 45]. The name machine learning was introduced in 1959 by Arthur Samuel [57]. The fundamentals were presented by Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?" [22]. Tom M. Mitchell provided a formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." [43].

Machine learning has two main types of learning algorithms: supervised learning and unsupervised learning. Supervised learning algorithms build a mathematical model of a training data set, which consists of a set of training examples that are used as inputs for each output. The values of the outputs are known in supervised learning. Supervised learning algorithms are used for classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range [1]. Unsupervised learning algorithm builds a mathematical model of a data set which has no known outputs for the respective inputs. Unsupervised learning algorithms are used to find structures in the data such as groups or clusters [56].

Deep learning is a subset of machine learning methods concerned with algorithms inspired by the structure and function of the brain called artificial neural networks (ANN). The concept "deep learning" refers to the structure of ANN as they typically have many (deep) layers and large number of parameters that enable learning [32, 35]. CNN is one implementation of deep learning. CNN is a specialized kind of neural network for processing data that have a known grid-like topology like image data, which can be thought of as a 2D grid of pixels. CNN is a serial structure that consists of consecutive layers, which convolve the inputs in order to extract information. Convolution layers have kernels (a.k.a. filters) with parameters that are learned during training [21].

In machine learning applications, the input of a convolution is usually a multidimensional array of data, and a kernel that is usually a multidimensional array of parameters, is slid over the input. These multidimensional arrays are referred to as tensors. What is called convolution operation in CNNs is actually cross-correlation, as the kernel is not flipped, however, as the values of the kernel are set during the training procedure, this distinction has no practical meaning. The cross-correlation (convolution henceforth) is formally defined as follows:

$$\mathbf{F}(i,j) = (\mathbf{K} \cdot \mathbf{I})(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{I}(i+m, j+n)\mathbf{K}(m,n), \tag{3.3}$$

where $\boldsymbol{F}$ is the result of the convolution called feature map, $\boldsymbol{K}$ is the kernel, $\boldsymbol{I}$ is the input, i is the row index of the feature map, j is the column index of the feature map, m is the row index of the kernel, and n is the column index of the kernel [21].

The convolution layer applies the convolution operator to the input tensor, and also transforms the input depth to match the number of kernels. The number of kernels is a design parameter, which can be found empirically by monitoring the performance of the model, but usually the physical memory of the used computer sets the upper limit. The depth of the output of a convolution layer is the number of the kernels at this layer. The other parameters of the convolution layer are: the width and height of the input tensor (e.g., an image), the width and height of the kernel, convolved width and height of the tensor (the output of the layer), the number of pixels (neurons) that the kernel moves over at each step called stride, and the number of zeros added to the border of the tensor called padding. The number of convolution layers in a CNN architecture is also a design parameter, which depends on a used dataset and a task to be performed. The function of the convolution layer is to extract features from the tensors (e.g. images), and accordingly, to form feature maps at the respected layer. Convolutions are computed at each convolution layer over the output of the previous layer using a trainable kernel. The feature extraction consists of successive convolution layers. Empirically, the convolution operation typically implements a local edge detector, especially on the first convolution layer of the architecture. Subsequently, the convolution operation extracts features from the previous feature maps, which are the results of the previous convolution operations of the respective layers. Thus, it is much more difficult to depict what the latter convolution layers implement. Multiple convolution kernels are used at each convolution layer to create several different feature maps. Primarily, it is the values of these kernels that CNN learns through a training process [21].

Besides convolution layers, the layered structure has other layers as well, and two different conventions are commonly used to depict the structure. In one of them, each component, such as convolution or pooling, is presented as a layer. In the other, the components that follow the convolution operation and process the data before the next convolution layer, are grouped with the preceding convolution operation to form a single layer, resulting in a lower number of layers. The components of a typical convolution layer of the CNN (according to the latter convention) include the rectified linear unit (ReLU) layer and the pooling layer. Various normalization layers are also commonly used as well as the dropout layer for avoiding overfitting [21].

The ReLU layer computes a piecewise linear activation function, which transfers the input to the output if the input is positive or zero, but outputs zero if the input is negative. The ReLU layer does not change the spatial or depth information of the input data. Thus, the function of the ReLU layer is to apply a threshold operation that removes all negative weights and transforms the positive weights linearly, because this is shown to make the training of a CNN model faster than, e.g., when using the logistic activation function (sigmoid). This is due to its properties such as faster convergence compared to the sigmoid and tanh functions. Other advantages of the ReLU function are that it does not saturate (due to linearity) when the input is large, and it does not have the vanishing gradient problem suffered by other activation functions like sigmoid or tanh. It is also sparsely activated, i.e., it is zero for all negative inputs, which means that many units do not activate at all. Because a sparse network has fewer non-zero elements, and thus fewer computing, it is faster than a dense network [21, 25, 46].

The pooling layer is used to reduce the spatial dimensions, but not depth, of the input data. By pooling layer, the number of parameters of a CNN decreases, computation performance increases, and some translation invariance is gained. If the exact position of an object in an image is important, the pooling layer can be omitted, but then the outcome suffers probably from strong overfitting. Two simple commonly used pooling functions are: average pooling and max-pooling. In this study, the max-pooling is applied. The purpose of the max-pooling layer is to build robustness to small distortions. The operation of the max-pooling layer is similar to that of the convolution layer with respect to the sliding kernel. The kernel of the max-pooling layer slides over the input and it takes the maximum value of the weights of each area that the kernel covers. Therefore, the max-pooling layer samples down the width and height of the input, but not the depth of it [21, 25].

Dropout layers are for reducing overfitting. This is achieved by setting zero to the output of the hidden neurons with the probability of 0.5. The neurons which are dropped out do not contribute to the forward pass and do not participate in backpropagation. In this way, the neural network samples a different architecture for each input, but the weights are still shared [21, 25, 61].

In classification , CNN is appended with one or more fully connected layers. The fully connected layers connect each neuron in the preceding layer to each neuron in the output layer. The operation of the CNN architecture may be seen in a way that the convolution layers extract features (aided by ReLu and pooling layers) and fully connected layers classify the input data. The number of the output neurons of the last fully connected layer is the same as the number of classes. The softmax function can be used in multiclass cases after the last fully connected layer, because it provides the result as a probability of each class [21].

3.4.1 Applied CNN Models

In this study, image classification is primarily based on CNN. Figure 3.9 depicts the classification process. Parameters provided by the radar system have only slight weight to bird species identification. The speed and the size estimate of a bird are the parameters applied. The size estimate is based on the distance of a bird and its size in pixels in the captured image. Images are always captured in series, which increases a possibility that the target bird is in good position in the image in terms of identification. As a result, the applied size estimate is actually the average of the size estimates of images in a series. Nevertheless, it is only a coarse approximation of the size, and its value for the final identification is diminutive. The speed of the target bird is measured by the radar, and it is much more accurate than the size estimate. Collected flight speed data show that, in average, the speed of waterfowl is two times

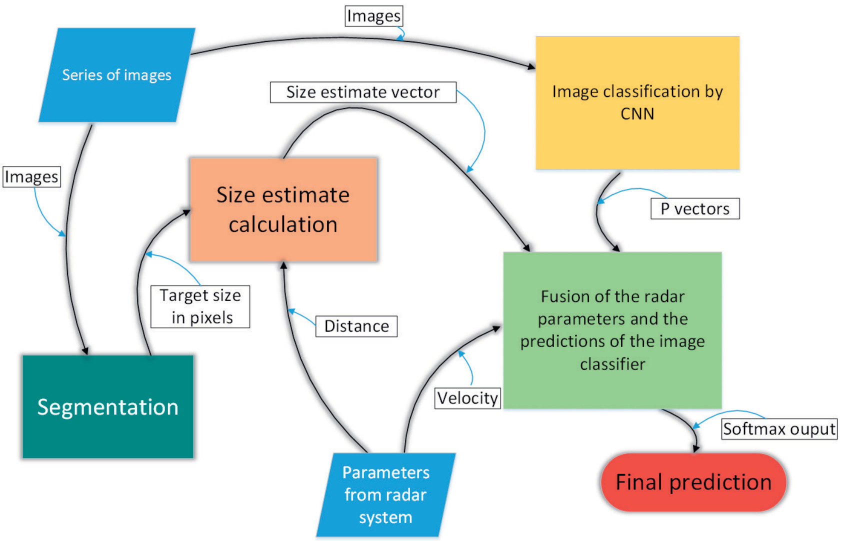

Images
Series of images
Size estimate vector
Image classification by CNN
Images
Size estimate calculation
P vectors
Target size in pixels
Distance
Fusion of the radar parameters and the predictions of the image classifier
Segmentation
Velocity
Softmax ouput
Final prediction
Parameters from radar system

Figure 3.9 Classification process.

larger that of gulls, terns, and the white-tailed eagle.

A CNN model was developed for extracting features from the images and to classify bird species. In this thesis, a single CNN model, which is applied to various classifiers, has 19 layers. The layered architecture is illustrated in Fig 3.10 and Table 3.2. Normalization applied in this architecture is the cross-channel normalization [31].

Values of all the hyperparameters were fixed in all publications, except for pub-

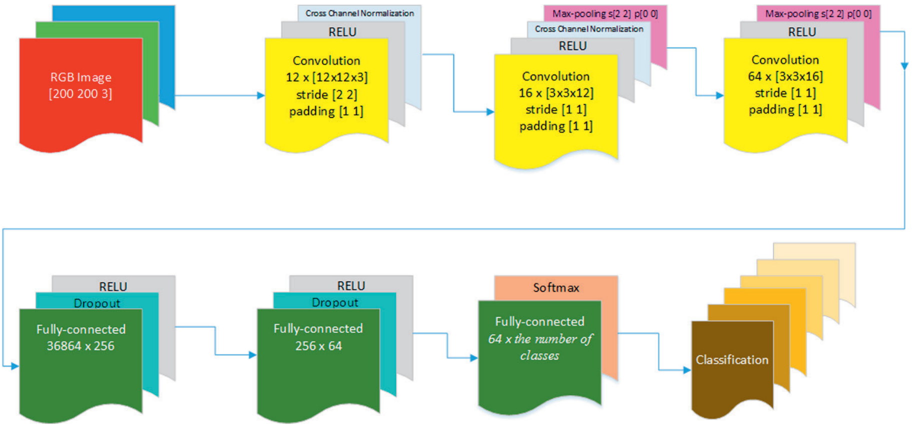


Figure 3.10 Architecture of the basic CNN model.

Table 3.2 The CNN architecture.

Layer	Function	Kernel Size	# Feature Maps
1	Input Layer (RGB image)	-	-
2	Convolution 1	12 × 12 × 3	12
3	ReLU	-	12
4	Normalization	-	12
5	Convolution 2	3 × 3 × 12	16
6	ReLU	-	16
7	Normalization	-	16
8	Max-pooling	2 × 2 × 12	16
9	Convolution 3	3 × 3 × 16	64
10	ReLU	-	64
11	Max-pooling	2 × 2 × 16	64
12	Fully-connected 1	-	-
13	Dropout 1	-	-
14	ReLU	-	-
15	Fully-connected 2	-	-
16	Dropout 2	-	-
17	ReLU	-	-
18	Fully-connected 3	-	-
19	Softmax	-	-

lication [P3], in which case manual tuning for selecting the learning rate parameter was tested. The tests were carried out by using the learning rate decay schedule (LRDS). In the LRDS method the learning rate was dropped by a factor of 0.1 when a given number of epochs was reached. This given number of epochs is the effective value of the learning rate drop period (LRDP). The motivation for using the LRDS method comes from the fact that as the training proceeds with shorter leaps on the loss function surface from some point on, the optimal value for the weights can be found more accurately. If only the short leaps would be applied, the number of epochs should be very large, thus resulting in significant increase of training time. The challenge was to find the points from where on the learning rate should be reduced. The tests showed that the performance of the CNN model can be slightly increased by the LRDS method.

In this study, the first CNN model had also a support vector machine (SVM) classifier (a.k.a. SVM-on-top model) [17, 26]. In this approach, the CNN just extracts the features and the SVM completes the classification. However, the model with the SVM did not perform better than the CNN alone and, as a result of that, the SVM was omitted. The model with the SVM classifier was used in publications [P1-P3]. The original (without augmentation) sizes of the datasets applied to the CNN models are given in Table 3.3. In this table, the size of the entire original dataset is given in the column 'Dataset Size', and the size of the balanced dataset used in in publications [P1-P4] is given in the column 'Balanced Dataset Size'.

In publication [P5], imbalanced dataset [24, 33] was applied in order to use all images in the original dataset. The number of classes was increased up to 11 in the publication [P4] resulting in a smaller number of data examples in the balanced dataset, which can be seen form the table. The number of images in the balanced dataset was not increased compared to the previous balanced datasets in spite of the increased number of images in the respective original dataset. This is due to the distribution of bird species in the test area. In data collection, the rate of increase of the number of data samples (images) of the commonest species was clearly higher than the rate of increase of the number of the scarcest species. This resulted in the relatively low number of data examples in the smallest class, which was only 396, and as undersampling is used as a resampling method, the number of data examples in the balanced dataset remained low.

In the course of this doctoral study, it became clear that it is very difficult, even impossible, to collect enough images of scarcer bird species in the test area in order

Table 3.3 Dataset sizes and the number of classes applied to CNN models in the publications [P1-P5].

Publication	Dataset Size	Number of Classes	Balanced Dataset Size
P1	14783	6	3786
P2	17514	6	6984
P3	20373	8	9312
P4	23552	11	4356
P5	24631	14	-

to apply a balanced dataset to train the classifier. This motivated a use of imbalanced dataset as the training examples to CNN classifiers in the publication [P5]. In this approach, bird species identification is achieved by a hybrid model [59] of hierarchical and cascaded [34, 39, 52, 58] classifiers. Each classifier in this model has the same basic CNN architecture as the previous models, but the statistics of the training dataset was utilized in order to obtain a threshold for the respective classifier. The hybrid model has eight classifiers. The number of classes of these classifiers is typically two, except for the top-level classifier with four classes, and a classifier of the group of other-waterfowl with five classes. These exceptions are rationalized as follows: two classes at the top level, the white-tailed eagle and swans, are well separable in terms of all data, thus their classification straight away at the top level is justified. The performance of the classifier of the group of other-waterfowl is adequate to classify all these five classes simultaneously. Figure 3.11 presents a revised flow chart (compared to that in the publication [P5]) of the image classification process when the hybrid model is applied.

Classification prediction (given as probability) for each test image fed to a given classifier in the hierarchy can be used as a probability distribution for the classifier. This enables determination of thresholds, which are used to improve classification performance. A threshold is defined for each class (a single species or a group of species) in the hierarchy. The use of a threshold requires that the input data of a given classifier are classified into only two classes. If the number of classes is higher, the class of interest is dealt as a positive class, and the other classes are combined to form a negative class. For more details, see [P5].

3.4.2 Results of Image Classification

In this thesis, confusion matrix, ROC curve, and TPR are used as metrics for classification performance. TPR is used instead of accuracy because when dealing with class-imbalanced dataset, accuracy typically does not provide enough information for classification. For a binary classification problem, accuracy is given:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.4}$$

where *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative. Consider, for example, a dataset of two classes: gray-backed gulls (GBG,

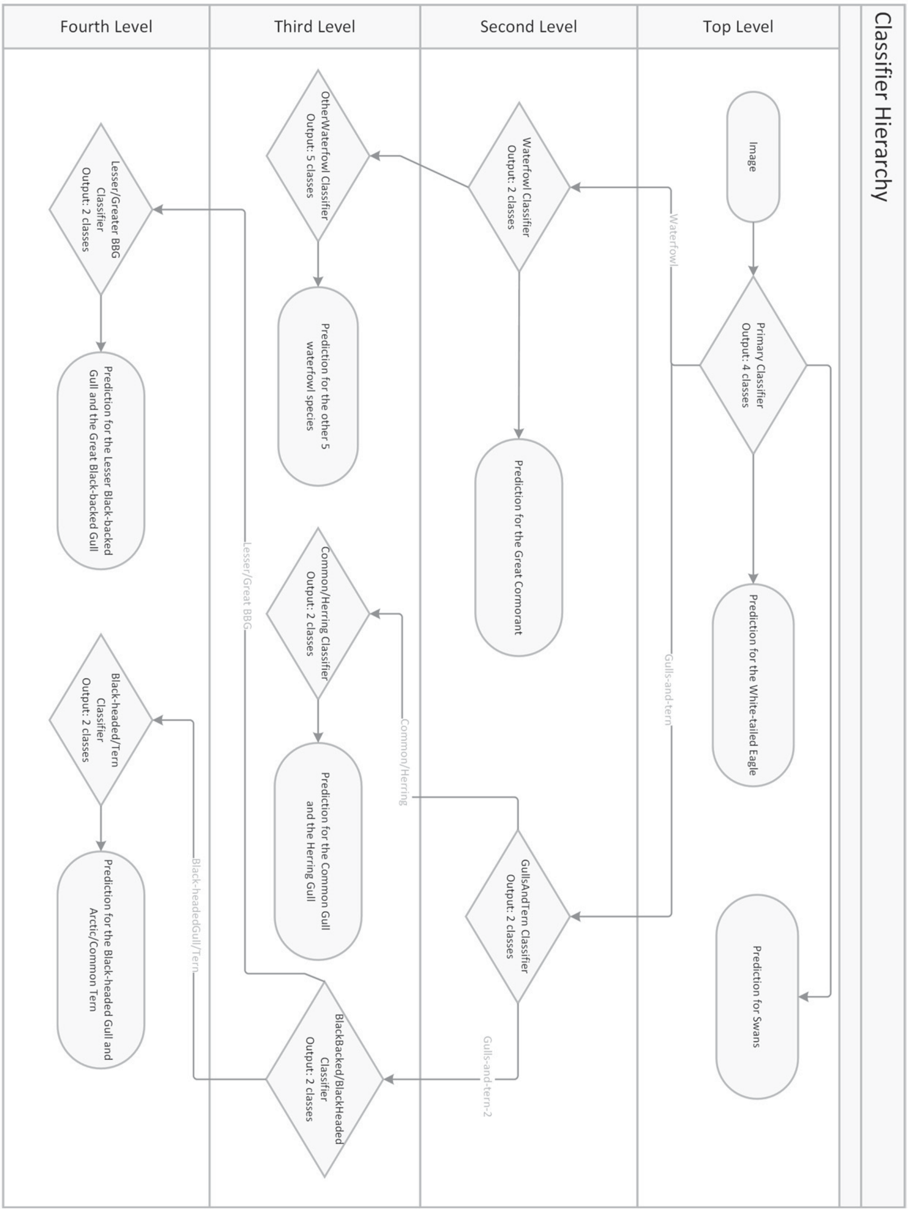

Classifier Hierarchy
Top Level
Second Level
Third Level
Fourth Level
Image
Primary Classifier
Output: 4 classes
Prediction for the White-tailed Eagle
Prediction for Swans
Waterfowl
Gulls-and-tern
Waterfowl Classifier
Output: 2 classes
Prediction for the Great Cormorant
GullsAndTern Classifier
Output: 2 classes
Gulls-and-tern-2
Common/Herring
OtherWaterfowl Classifier
Output: 5 classes
Prediction for the other 5 waterfowl species
Common/Herring Classifier
Output: 2 classes
Prediction for the Common Gull and the Herring Gull
BlackBacked/BlackHeaded Classifier
Output: 2 classes
Lesser/Great BBG
Black-headedGull/Tern
Lesser/Greater BBG Classifier
Output: 2 classes
Prediction for the Lesser Black-backed Gull and the Great Black-backed Gull
Black-headed/Tern Classifier
Output: 2 classes
Prediction for the Black-headed Gull and Arctic/Common Tern

Figure 3.11 Image classification by the hybrid model of hierarchical and cascaded models.

the common gull and the herring gull) and black-backed gulls (BBG, the great black-backed gull and the lesser black-backed gull) with 100 GBGs and 10 BBGs. The BBG class is assigned to the positive class. If a classifier model used to classify this dataset predicts correctly 2 BBGs and 99 GBGs, and the rest of the predictions are incorrect, the outcome is: *TP* = 2, *TN* = 99, *FP* = 1, and *FN* = 8. This gives a good general accuracy of 0.9182. However, the model correctly identifies 99 as GBGs out of 100 GBGs, but of the 10 BBGs, the model correctly identifies only 2 BBGs. By using TPR (a.k.a. recall or sensitivity) as a metric for this model fed by the given data, the value of the metric becomes 0.2, which reveals the weakness of the model as a classifier. For a binary classification problem, TPR is given:

$$TPR = \frac{TP}{TP + FN}, \tag{3.5}$$

and precision is given:

$$Precision = \frac{TP}{TP + FP}. \tag{3.6}$$

It can be seen from the equations above that precision measures the probability of correct detection of positive values, while TPR is a probability of detection, i.e., it measures the proportion of actual positives that are correctly identified as such. Thus, TPR and FPR measure the ability to distinguish between the classes. For a binary classification problem, FPR is given:

$$FPR = \frac{FP}{FP + TN}, \tag{3.7}$$

and they are both used simultaneously to form a probability curve called the ROC curve, which gives a performance measurement for a classification problem at various thresholds settings.

3.4.2.1 The Outcomes

The early results of this study are given in the respective publications [P1-P4]. The early models have relatively high performance as image classifiers, but the number of classes was not high enough for practical use in the test site. These early models used balanced datasets, which limited the number of classes, even though the collected number of data examples (images) increased during the development process. Finally, the number of data examples of the balanced dataset, which was used in publication [P4], showed a downturn when the scarcest classes were included into the data.

The best results were accomplished by using the hybrid model in publication

[P5]. In the study of the hybrid model, three distinct CNN models were implemented and tested. The first model was a single classifier trained on a balanced dataset. The second model was the same single classifier, without thresholds, trained on an imbalanced dataset. The third model was a hybrid of hierarchical and cascaded model, with thresholds, trained on the same imbalanced dataset as the second model. The data augmentation algorithm was applied to training datasets of all models. The first and the second CNN model used the basic CNN architecture presented in Table 3.2. In the first model, under-sampling was used as a resampling method in order to compose the balanced training dataset, i.e., data examples were deleted from the over-represented classes. The second and the third CNN models were tested in parallel as they were both trained on the same imbalanced dataset. The third model consists of eight classifiers trained on various groups of bird species. Each classifier in this model used the same basic CNN architecture. Determination of pseudo-classes became possible as the groups were applied and thresholds were used. This is important because the perfect classification performance, in terms of given data and classes, was not always possible, and CNN always gives a prediction over the classes that it is trained on, thus rendering impossible to predict any other classes. Using thresholds gives a possibility to define pseudo-classes when none of the predicted probabilities are larger than the value of the applied threshold. In these cases, a test image can be classified in a suitable pseudo-class with the label of which is determined by the logic of the implemented software. These pseudo-class labels can be, e.g., unknown bird species, unknown waterfowl species, unknown gull species, black-backed gull, grey backed gull, and so forth.

The classification performances for four groups of bird species and the lesser black-backed gull of these three aforementioned models are given in Table 3.4. The abbreviations in this table are: WTEA = the white-tailed eagle, SWSP = swan species, and LBBG = the lesser-black-backed gull. The group of WTEA consists of only a single species, and swan species were not tried to classify further than onto the group

Table 3.4 TPRs for classifier models.

Classifier	WTEA	SWSP	waterfowl	gulls-and-terns	LBBG
Hybrid	1	1	0.9935	1	0.9231
Imbalanced single	0.9773	0.4000	0.7629	0.7691	0.6923
Balanced single	1	0.8000	0.8621	0.7762	0.8846

Table 3.5 Confusion matrix for all the classes.

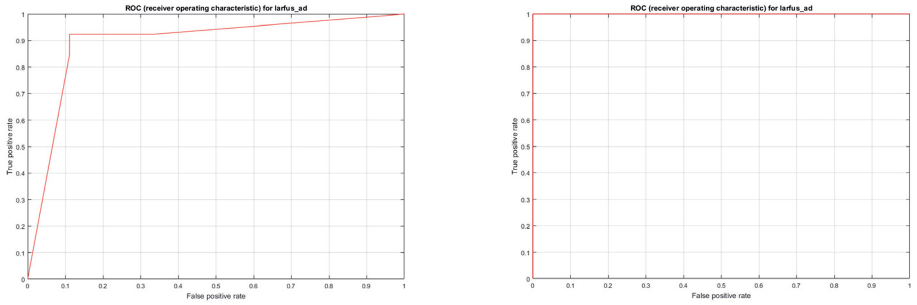
	WTEA	LOSP	GRCO	COEI	COGO	SWSP	VESC	RBME	GBBG	HEGU	LBBG	COGU	BHGU	CATE
WTEA	44	0	0	0	0	0	0	0	0	0	0	0	0	0
LOSP	0	7	0	0	0	0	0	0	0	0	0	0	0	0
GRCO	1	0	98	0	1	0	0	0	0	0	0	0	0	0
COEI	0	0	0	16	0	0	0	0	1	0	0	0	0	0
COGO	0	0	0	0	19	0	1	1	0	0	0	0	0	0
SWSP	0	0	0	0	0	5	0	0	0	0	0	0	0	0
VESC	0	0	0	0	0	0	3	0	1	0	0	0	0	0
RBME	0	0	0	0	0	0	0	4	1	0	0	0	0	0
GBBG	0	0	0	0	0	0	0	0	4	1	4	0	0	0
HEGU	0	0	0	0	0	0	0	0	0	66	0	6	0	0
LBBG	0	0	0	0	0	0	0	0	0	0	26	0	0	0
COGU	0	0	0	0	0	0	0	0	0	1	0	64	0	0
BHGU	0	0	0	0	0	0	0	0	0	0	0	0	32	0
CATE	0	0	0	0	0	0	0	0	0	0	0	1	0	31

level, thus the group also includes only one class. The table clearly shows that the hybrid model performed significantly better than the other two models. The classification performance of the hybrid model for all of the 14 classes is given in Table 3.5. The abbreviations used in this table are presented in Table 3.6.

Two receiver operating characteristic (ROC) curves are presented for the lesser black-backed gull in Fig 3.12. Area under the curve (AUC) is 1 for the rightmost curve (3.12b) and 0.9250 for the leftmost curve (3.12a).

Table 3.6 Class labels for all of the classes.

Species Name (Eng.)	Species Name (Lat.)	Class Label
Loon species	Gavia sp	LNSP
Swan species	Cygnus sp	SWSP
Great Cormorant	Phalacrocorax carbo	GRCO
Common Eider	Somateria mollissima	COEI
Common Goldeneye	Bucephala clangula	COGO
Velvet Scoter	Melanitta fusca	VESC
Red-breasted Merganser	Mergus serrator	RBME
White-tailed Eagle	Haliaeetus albicilla	WTEA
Great Black-backed Gull	Larus marinus	GBBG
Herring Gull	Larus argentatus	HEGU
Lesser Black-backed Gull	Larus fuscatus	LBBG
Common Gull	Larus canus	COGU
Black-headed Gull	Chroicocephalus ridibundus	BHGU
Common/Arctic Tern	Sterna hirundo/paradisaea	CATE



(**a**) (**b**)

Figure 3.12 Two ROC curves for the lesser black-backed gull (*Larus fuscus fuscus*) that demonstrates the advantage of using thresholds. (**a**) ROC curve without a threshold, and (**b**) ROC curve with a threshold of 0.9993 applied to the great black-backed gull (*Larus marinus*) at the forth level of the hierarchy, i.e., an image is classified as GBBG only if the prediction of the classifier is larger than the threshold, otherwise the image is classified as LBBG.

The ROC curves for the lesser black-backed gull are from the fourth level of the hybrid model, which deals with the group of black-backed gulls, i.e., either the lesser black-backed (LBBG) gull or the great black-backed gull (GBBG). The curve 3.12b is gained by setting the GBBG as the positive class and the LBBG as the negative class, regardless that the LBBG is in the interest. By setting a threshold with a high value of 0.9993 to the GBBG, a test image is classified as the GBBG only if the given probability for this class is higher than the threshold, and otherwise the test image is classified as the LBBG. This method is justified because no LBBG should be miss-classified according to the environmental license, and the GBBG is quite scarce bird species in the test area.

4 CONCLUSIONS AND DISCUSSION

The developed bird identification system is evaluated in terms of bird detection and identification (classification) ability. Bird detection was two-fold: initially, the applied radar system detected birds in the sky, and the developed steering software for the motorized video head drove the camera to the correct position. Successful results were shown for the latter part of the detection problem. The former part (the radar system) was beyond the scope of this study. Practical results for automatic bird identification were presented.

The results of [P4] showed that the delay between the timestamp of a track and the current clock time of the software server seems to be the most significant factor for inaccurate targeting of the camera. This is further evident in higher wind speed circumstances. Visual observations showed that flight paths of all bird species in the test area were increasingly erratic in high-wind weather conditions. This emphasizes the effect of the delay in targeting of the camera. However, more research is needed for reducing the time delay between the timestamp of a track and the current clock time of the software server.

The results of [P4] also showed that the precise vertical angle error was difficult to estimate because of the stochastic element of a target bird position, which is caused by the aforementioned delay. Therefore, the constant vertical error compensation was discovered empirically during the targeting of the camera. The probability that target birds will be captured in the focus rectangle cell from any distance is large if the vertical turning angle error is 0.25°or smaller. If this error is greater than 0.25°, target birds will probably be missed.

The autofocus system of the camera and the lens showed to be problematic resulting in an unreliable focus. This seemed to be linked to the far range that the images are taken from. If the autofocus system cannot reach the correct focus, the camera does not release the shutter, and therefore images are not taken. No solution was found from the API to release the shutter even if the correct focus is not reached. This became important because the image classifier showed good generalization ability, even when it was fed by slightly out-of-focus images. As a solution to the autofocus problem, the focusing system was switched off, and manual focusing was used instead. As a result of this, focusing was completed to a target at a chosen range before the system was started resulting in the need of refocusing when the photographing range was changed. The camera and the lens used are both two generations old. The most recent camera and lens generation has better autofocus system, which may be a long term solution to this autofocus problem.

The applied CNN model showed sufficient performance as an image classifier. Mostly, images that were totally out-of-focus or images in which the bird is in unfavourable position to be identified even by the human eye, were misclassified.

Early tests showed that CNN is a feasible method to classify real-world data (images of birds). It became clear during the tests that the SVM-on-top model failed to bring substantial improvement to the classification performance, thus SVM was omitted from the CNN model. The early tests also implied that the single CNN model trained on the balanced dataset, formed by images taken in the test site, would not achieve adequate performance as an image classifier. The result was not better, in terms of classification performance, when the single CNN model was trained on the imbalanced dataset. However, the following tests showed that the hybrid model had significantly better performance as an image classifier than the above classifiers, especially if the thresholds are applied to the classes. The results of [P5] showed that the only problematic class, in terms of the environmental license, is the LBBG. Possible solutions are: misclassification of the GBBG as the LBBG is accepted or the two classes are combined to one class called the black-backed gulls. Another problematic pair of classes (bird species) was the gray-backed-gulls: the herring gull and the common gull. The bird species that this group consists of are also very similar to each other in terms of morphology. This leads to a conclusion (assessed by human eye) that the overlapped area of the classification boundary is clearly wide for both these groups. If the species level identification is important, which is not always the case as in this study, the results imply that significant increase of classification performance can be achieved only by collecting more images of these groups.

The results of [P5] showed that when the basic CNN model was modified with architecture of more than three convolution layers, these models did not perform better than the original CNN model. This implies that the original model, with the architecture of three convolution layers, is capable to extract all relevant features from the training images, and additional convolution layers cannot provide any more information.

The overall conclusion of [P5] is that the parameters provided by the radar system (speed and distance, which enables the size estimate computing) turned to be much less useful than was initially expected. The size estimate was constantly too coarse for giving adequate accuracy, rendering impossible its usage to increase the performance of the image classifier. Speed had potentially more significant contribution to the classification performance, but in practise, the only case it had a crucial role was classification between the white-tailed eagle and the great cormorant, because the image classifier misclassified the great cormorant as the white-tailed eagle in four cases.

All in all, proof of concept was shown in this study. The problem of real-world bird identification can be approached successfully by the proposed system and the applied methods. However, a deeper insight into the pros and cons of the system is gained by examining it at the component level. The radar system is the top level component of the developed system even though it was not a subject of the study. Thus, if the radar system fails to detect target birds entirely, or it fails to pass the accurate position of target birds, the developed system cannot operate. Moreover, the radar system is by far the most expensive component of the system, and at the time when the radar system was chosen, only two manufacturers had a suitable system for this kind of use. The high cost of a suitable radar system might constitute an obstacle for wider use of the developed system. Ability to capture images of target birds is primarily based on the passed coordinates, which was found to be accurate enough. This leaves the aforementioned delay to be the main reason for inaccurate targeting of the camera. According to the manufacturer of the radar system, the only way to significantly reduce the delay is to increase the rotation speed of the radar antenna, which is not possible to implement in the used radar system.

The developed system has limitations concerning its utilization. Obvious limitations are caused by prevailing weather such as poor visibility and high winds (explained above). In practice, operating range is limited distally (with the used lens) roughly at 1100 m, because the smallest bird species do not cover enough pixels in their images. Shorter distance than 100 m was not tested in this study. The classification (bird identification) is limited to only those species that were included in the used training set. Thus, if the system is implemented in other location, images of possible new species to the system must be collected, the thresholds must be updated, and at least some of the classifiers in the hierarchy must be retrained on the updated training sets.

At the time when this study commenced, the main objective was to develop an automatic bird identification system for existing (offshore) wind farms in order to monitor various bird species movement near the turbines. At present, the original objective still holds, but demand for a mobile system that can identify bird species automatically is increasing. The main reason for this is that it is cost effective to study a potential wind farm locations in terms of Environmental Impact Assessment (EIA) before any other action is taken. EIA is not only an obligatory procedure but also one of the most significant component in the process to gain a building licence to a new wind farm. With a little effort, the developed system is modifiable to the desired mobile system.

One of the main research theme in the future is to link the developed system to a deterrent system. This deterrent system should use methods that are feasible in the offshore environment considering also possible limitations set by the maritime administration.

The developed system is principally comparable to the DTBird [36]. This is a commercial system, which is probably the reason that no detailed information of applied methods is available. However, [41] showed that the DTBird is capable to monitor birds in flight, but bird species level identification is not usually possible. The main objective of the DTBird seems to be collision monitoring from video footage, whereas the main objective of this thesis is bird species identification automatically by the developed system. Evidently, both systems are capable to detect birds in flight.

In some respects, systems or applications such as the WT-bird, Birdsnap, and collecting time-lapse images around a wind farm are comparable to the developed system in this study. The objective of the WT-bird [71, 72, 73] is also to detect and register possible collisions between birds and wind turbines, and no automatic bird species identification is implemented. Birdsnap [5] is based on entirely different methods (no CNN), but it also utilizes images of birds and a probability distribution (spatio-temporal class priors) of bird species in a certain location and time of year. The objective of this application is to implement an on line field guide. Compared to the system in this thesis, the images in Birdsnap are manually collected and added to the application. This gives a wide control to the images in terms of image processing, whereas the similar way of control is not usually possible in automatically operating systems. Only the accuracy of 0.8240 is given to describe the performance of Birdsnap as an image classifier. The performance of the system developed in this thesis is usually given as TPR, but if accuracy is used to show the performance, it is 0.8975. Compared to the system that uses time-lapse imaging developed by [75], the system developed in this study is capable to detect birds from the whole wind farm area, which makes the direct comparison of the detection performances unreasonable. [75] also included image classifiers in their system. The image classifiers were based on four methods, and the best performed method was CNN. Their classification results are given in a ROC curve in their study, and when these results are compared to the performance of the hybrid CNN model used in this thesis, it seems that the performance of the hybrid CNN model is significantly better. This is especially obvious when the thresholds are applied to separate the most challenging classes such as the lesser black-backed gull and the great black-backed gull, see Fig. 3.12.

REFERENCES

[1] E. Alpaydin. *Introduction to Machine Learning*. ISBN 978-0-262-01243-0. Cambridge, MA: MIT Press, 2010.

[2] W. Band, M. Madders and D. P. Whitfield. Developing field and analytical methods to assess avian collision risk at wind farms. *Birds and wind farms Risk assessment and mitigation*. Ed. by M. de Lucas, G. F. E. Janss and M. Ferrer. ISBN 978-84-87610-18-9. Spain: Quercus, 2007.

[3] A. T. Baxter and A. P. Robinson. A comparison of scavenging bird deterrence techniques at UK landfill sites. *International Journal of Pest Management* 53 (2007), 347–356. DOI: 10.1080/09670870701421444.

[4] T. Berg and P. N. Belhumeur. POOF: Part-Based One-vs.-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation. CVPR. Portland, OR: IEEE, 2013. DOI: 10.1109/CVPR.2013.128.

[5] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs and P. N. Belhumeur. Birdsnap: Large-Scale Fine-Grained Visual Categorization of Birds. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. San Diego, CA: IEEE, 2014. DOI: 10.1109/CVPR.2014.259.

[6] J. Bernardino, R. Bispo, H. Costa and M. Mascarenhas. Estimating bird and bat fatality at wind farms: A practical overview of estimators, their assumptions and limitations. *New Zealand Journal of Zoology* 40 (2013). DOI: 10.1080/03014223.2012.758155.

[7] K. Bevanger, F. Berntsen, S. Clausen, E. L. Dahl, Ø. Flagstad, A. Follestad, D. Halley, F. Hanssen, L. Johnsen, P. Kvaløy, P. Lund-Hoel, R. May, T. Nygård, H. C. Pedersen, O. Reitan, E. Røskaft, Y. Steinheim, B. Stokke and R. Vang. *Pre-and post-construction studies of conflicts between birds and wind turbines in coastal Norway*. Report 620. Report on findings 2007-2010. Norway: NINA, 2010.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning*. ISBN 978-0-387-31073-2. New York, NY: Springer, 2006.

[9] R. Brabant, N. Vanermen, E. V. M. Stienen and S. Degraer. Towards a cumulative collision riks assessment of local and migrating birds in North Sea offshore wind farms. *Hydrobiologia* 756 (2015), 63–74.

[10] B. Bruderer. The Study of Bird Migration by Radar, part1: The Technical Basis. *Naturwissenschaften* 84 (1997), 1–8.

[11] CIE. *Proceedings, Vienna Session 1963*. Committee Report E-1.4.1. Bureau Central de la CIE, 1964, 209–220.

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. CVPR'05. San Diego, CA: IEEE, 2005, 886–893. DOI: `10.1109/CVPR.2005.177`.

[13] M. Desholm, A. D. Fox, P. D. L. Beasly and J. Kahlert. Remote techniques for counting and estimating the number of bird-wind turbine collisions at sea: a review. *Ibis* 148 (2006), 76–89.

[14] M. Desholm and J. Kahlert. Avian Collision Risk at an Offshore Wind Farm. *Biology Letters* 1 (2005), 296–298. DOI: `10.1098/rsbl.2005.0336`.

[15] A. L. Drewitt and R. H. W. Langston. Assessing the impacts of wind farms on birds. *Ibis* 148 (2006), 29–42.

[16] A. L. Drewitt and R. H. W. Langston. Collision effects of wind-power generators and other obstacles on birds. *Annals of the New York Academy of Sciences* 1134 (2008), 233–266.

[17] K. B. Duan and S. S. Keerthi. Which Is the Best Multiclass SVM Method? An Empirical Study. *Multiple Classifier Systems* (2005), 278–285.

[18] J. Everaert and E. W. M. Stienen. Impacts of wind turbines on birds in Zeebrugge (Belgium). Significant effect on breeding tern colony due to collisions. *Biodiversity and Conservation* 16 (2007), 3345–3359.

[19] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory* 904 (1995), 23–37.

[20] B. Goller, B. F. Blackwell, T. L. DeVault, P. E. Baumhardt and E. Fernández-Juricic. Assessing bird avoidance of high-contrast lights using a choice test approach: implications for reducing human-induced avian mortality. *PeerJ* 6 (2018). DOI: `10.7717/peerj.5404`.

[21] I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016. URL: `www.deeplearningbook.org`.

[22] S. Harnad. The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence. *The Turing Test Sourcebook: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Ed. by R. Epstein and G. Peters. Alphen aan den Rijn, The Netherlands: Kluwer, 2008.

[23] S. Haykin. *Neural networks. A comprehensive foundation*. 2nd ed. ISBN 0-13-908385-5. New York, NY: Prentice Hall/Pearson, 1994.

[24] P. Hensman and D. Masko. *The Impact of Imbalanced Training Data for Convolutional Neural Networks*. Available online. Accessed on 21 January 2020. 2018. URL: `https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf`.

[25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv preprint. arXiv:1207.0580. 2012.

[26] F. J. Huang and Y. LeCun. Large-scale learning with SVM and convolutional nets for generic object categorization. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*. CVPR 2006. New York, NY: IEEE, 2006, 284–291.

[27] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato and Y. LeCun. What is the best multi-stage architecture for object recognition. *IEEE 12th International Conference on Computer Vision, ICCV 2009*. ICCV 2009. Kyoto, Japan: IEEE, 2009, 2146–2153.

[28] S. Jia, P. Wang, P. Jia and S. Hu. Research on data augmentation for image classification based on convolution neural networks. *2017 Chinese Automation Congress*. CAC. Jinan, 2017.

[29] J. Kalhert, I. K. Petersen, A. D. Fox, M. Desholm and I. Clausager. *Investigations of Birds During Construction and Operation of Nysted Offshore Wind Farm at Rødsand*. NERI Report. Annual status report 2003. Rønde, Denmark: National Environmental Research Institute, 2004.

[30] K. L. Krijgsveld, K. Akershoek, F. Schenk, F. Dijk and S. Dirksen. Collision risk of birds with modern large wind turbines. *Ardea* 97 (2009), 357–366.

[31] A. Krizhevsky, I. Sutskever and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*. NIPS. Lake Tahoe, Nevada: Curran Associates Inc., 2012, 1097–1105.

[32] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*. Vol. 86. New York, NY: IEEE, 1998, 2278–2324. DOI: `10.1109/5.726791`.

[33] F. Li, S. Li, C. Zhu, X. Lan and H. Chang. Class-imbalance aware CNN extension for high resolution aerial image based vehicle localization and categorization. *2nd International Conference on Image, Vision and Computing*. ICIVC. Chengdu, China, 2017.

[34] H. Li, Z. Lin, X. Shen, J. Brandt and G. Hua. A Convolutional Neural Network Cascade for Face Detection. *IEEE Conference on Computer Vision and Pattern Recognition*. CVPR. Boston, MA, 2015.

[35] M. Li, T. Zhang, Y. Chen and A. J. Smola. Efficient Mini-batch Training for Stochastic Optimization. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge*. New York, NY, 2014, 661–670.

[36] S. Liquen Consultoría Ambiental. *DTBird*. Tech. rep. Spain, 2010. URL: `http://www.dtbird.com/`.

[37] M. de Lucas, G. F. E. Janss, D. P. Whitfield and M. Ferrer. Collision fatality of raptors in wind farms does not depend on raptor abundance. *Journal of Applied Ecology* 45 (2008), 1695–1703.

[38] S. Madge and H. Burn. *Wildfowl, an identification guide to the ducks, geese and swans of the world*. ISBN 0-7470-2201-1. London: Helm, 1988.

[39] R. Mao, Q. Lin and J. Allebach. Robust Convolutional Neural Network Cascade for Facial Landmark Localization Exploiting Training Data Augmentation. *Imaging and Multimedia Analytics in a Web and Mobile World 2018*. 2018, 374.1–374.5.

[40] A. T. Marques and et al. Understanding bird collisions at wind farms: An updated review on the causes and possible mitigation strategies. *Biological Conservation* 179 (2014), 40–52.

[41] R. May, Hamre, R. Vang and T. Nygård. *Evaluation of the DTBird video-system at the Smøla wind-power plan*. Report. Accessed on 25.11.2019. Trondheim, Norway: Norwegian Institute for Nature Research - NINA, 2012. URL: `https://www.nina.no/archive/nina/PppBasePdf/rapport/2012/910.pdf`.

[42] G. Mirzaei. Data Fusion in Multi-Sensory Environment of Infrared, Radar, and Acoustics Based Monitoring System. PhD thesis. University of Toledo, Ohio, 2013.

[43] T. Mitchell. *Machine Learning*. ISBN 978-0-07-042807-2. New York, NY: McGraw Hill, 1997.

[44] A. R. Muńoz-Gallego, M. de Lucas, E. Casado and M. Ferrer. Raptor mortality in wind farms of southern Spain: mitigation measures on a major migration bottleneck area. *Oral presentation at Conference on Wind energy and Wildlife impacts*. Trondheim, Norway, May 2011.

[45] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. ISBN-13 978-0262018029. Cambridge, MA: The MIT Press, 2012.

[46] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. *Proc. 27th International Conference on Machine Learning*. ICML'10. 2010, 807–814.

[47] NGA. *World Geodetic System 1984*. Report. Accessed on 14.02.2019. National Geospatial-Intelligence Agency, 2004. URL: `http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html`.

[48] K. M. Olsen and H. Larsson. *Terns of Europe and North America*. ISBN 0-7136-4056-1. London: Helm, 1995.

[49] K. M. Olsen and H. Larsson. *Gulls of Europe, Asia and North America*. ISBN 0-7136-4377-3. London: Helm, 2003.

[50] A. Pandey, J. Hermenc and R. Harness. *Development of a Cost-Effective System to Monitor Wind Turbines for Bird and Bat Collisions -Phase I: Sensor System Feasibility Study*. Report. CEC-500-2007-004. California Energy Commission, PIER Energy-Related Environmental Research, 2006.

[51] C. J. Pennycuick. Speeds and wing beat frequencies of migrating birds compared with calculated benchmarks. *The Journal of Experimental Biology* 204 (2001), 3283–3294.

[52] R. Rachmadi, K. Uchimura, G. Koutagi and Y. Komokata. Japan road sign classification using cascade convolutional neural network. *ITS (Intelligent Transport System) World Gongress*. CVPR. Tokyo, 2016, 1–12.

[53] M. A. Richards. *Fundamentals of Radar Signal Processing*. ISBN 0-07-144474-2. New York, NY: The McGraw-Hill companies, 2005.

[54] *Robin Report Viewer UserManual*. EN-2.1.6.1. Robin Radar Systems B. V. 2016.

[55] I. M. D. Rosa, A. T. Marques, G. Palminha, H. Costa, M. Mascarenhas, C. Fonseca and J. Bernardino. Classification success of six machine learning algorithms in radar ornithology. *Ibis* 158 (2016), 28–42.

[56] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Third. ISBN 978-0-13-604259-4. Upper Saddle River, NJ: Prentice Hall, 2010.

[57] A. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3 (1959), 210–229. DOI: 10.1147/rd.33.0210.

[58] Y. San, X. Wang and X. Tang. Deep convolutional network cascade for facial point detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*. Portland, OR: IEEE, 2013, 3476–3483. DOI: 10.1109/CVPR.2013.446.

[59] C. Silla and A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22 (2011), 31–72.

[60] N. I. Speranskaya. Determination of spectrum color co-ordinates for twenty-seven normal observers. *Optics and Spectroscopy* 7 (1959), 424–428.

[61] N. Srivastave, G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.

[62] W. S. Stiles and J. M. Burch. NPL colour-matching investigation: Final report. *Optica Acta* 6 (1959), 1–26.

[63] A. Stockman and L. T. Sharpe. Spectral sensitivities of the middle- and long-wavelength sensitive cones derived from measurements in observers of known genotype. *Vision Research* 40 (2000), 1711–1737.

[64] J. P. Swaddle, D. L. Moseley, M. K. Hinders and E. P. Smith. A sonic net excludes birds from an airfield: implications for reducing bird strike and crop losses. *Ecological Applications* (2016). DOI: 10.1890/15-0829.

[65] Vendian. *Blackbody color datafile*. Report. Vendian.org, 2016. URL: http://www.vendian.org/mncharity/dir3/blackbody/.

[66] J. P. Verhoef, P. J. Eecen, R. J. Nijdam, H. Korterink and H. H. Scholtens. *WT-Bird: a Low Cost Solution for Detecting Bird Collisions*. Report. Report ECN-C-04-046. Energy research Center of the Netherlands, 2004.

[67] J. P. Verhoef, C. A. Westra, H. Korterink and A. Curvers. *WT-bird: A novel bird impact detection system*. Report. Report ECN-CX-03-091. Energy research Center of the Netherlands, 2003.

[68] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. CVPR 2001. Kauai, HI: IEEE, 2001, 511–518. DOI: 10.1109/CVPR.2001.990517.

[69] R. Walls, C. Pendlebury, R. Budgey, K. Brookes and P. Thompson. *Revised best practice guidance for the use of remote techniques for ornithological monitoring at offshore wind farms*. Report. Report commissioned by COWRIE Ltd. COWRIE REMTECH-08-08, London: COWRIE Ltd, 2009.

[70] J. Wang and L. Perez. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *Stanford University Report* (2017). URL: http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf.

[71] E. J. Wiggelinkhuizen and H. J. den Boon. *Monitoring of bird collisions in wind farm under offshore-like conditions using WT-BIRD system, final report*. Report. Accessed on 11.02.2019. Energy research Center of the Netherlands, 2013. URL: http://www.we-atsea.org/wp-content/uploads/2013/01/RL2-3-2004-029-Monitoring-of-bird-collisions.pdf.

[72] E. J. Wiggelinkhuizen, L. W. M. M. Rademakers, S. A. M. Barhorst and H. J. den Boon. *Bird collision monitoring system for multi-megawatt wind turbines WT-Bird: Prototype development and testing*. Report. Report ECN-E-06-027. Energy research Center of the Netherlands, 2006a.

[73] E. J. Wiggelinkhuizen, L. W. M. M. Rademakers, S. A. M. Barhorst, H. J. den Boon, S. Dirksen and H. Schekkerman. *WT-Bird: Bird collision recording for offshore wind farms*. Report. Report ECN-RX-06-060. Energy research Center of the Netherlands, 2006b.

[74] G. Wyszecki and W. S. Stiles. *Color Science: concepts and methods, quantitative data and formulae*. 2nd ed. ISBN-13 978-0471399186. New York, NY: Wiley, 1982.

[75] R. Yoshihashi, R. Kawakami, M. Iida and T. Naemura. Evaluation of Bird Detection using Time-lapse Images around a Wind Farm. *Wind Energy* 20(12) (2017), 1983–1995. DOI: 10.1002/we.2135.

PUBLICATIONS

PUBLICATION

I

Automatic bird identification for offshore Wind farms: a case study for deep learning

J. Niemi and J. T. Tanttu

*ELMAR-2017, 59th IEEE International Symposium ELMAR-2017*2017

DOI: `10.23919/ELMAR.2017.8124482`

Automatic Bird Identification for Offshore Wind Farms: A Case Study for Deep Learning

Juha Niemi, Juha T. Tanttu,
Signal Processing Laboratory/Tampere University of Technology
P.O.B 300, 28101 Pori, Finland
juha.niemi@tut.fi

***Abstract*—An automatic bird identification system is required for offshore wind farms in Finland. Indubitably, a radar is the obvious choice to detect birds but actual identification requires external information such as digital images. The final bird species identification is based on a fusion of radar data and image data. We applied deep learning method for image classification and we developed a data expansion technique for the training data. We present classification results for the image classifier based on small convolutional neural network.**

***Keywords*—Classification; Deep Learning; Convolutional Neural Networks; Machine Learning; Data Expansion; Wind Farms**

I. Introduction

Several offshore wind farms are under construction on the Finnish west coast. The official environmental specifications define that bird species behaviour at the vicinity of wind turbines must be monitored. This concerns especially two species: White-tailed Eagle, Haliaeetus albicilla and Lesser Black-backed Gull, Larus fuscus fuscus. The only way to fulfil the demand cost efficiently is to automate monitoring, and that requires automatic bird species identification at such a level that the aforementioned bird species are separable from all other species in the study area. The prototype system for automated bird identification is developed and placed at a test location on Finnish west coast.

A radar is feasible choice for the detection of birds since the identification need is restricted to the flying birds only. If merely a radar is used the identification capability is limited to a few size classes. Radar suppliers have observed that the main cause for the limited identification capability is that variation between the same object in a different section in a radar beam (and in a different position as well) is larger than variation between two different objects.

A feasible way to bring more information into identification algorithm is to apply digital camera images. In this approach the radar is responsible for the detection. The radar locks on a target bird and provides the coordinates to the camera steering software. The camera steering system tracks the flying bird and the camera takes series of images. The data of digital images provides information about coloration and shape of the target. The system includes our software for controlling the camera and steering a motorized video head. The data provided by the radar consists of the velocity and the distance to the target. The final identification is the result of a fusion between information provided by the radar and information extracted from digital images.

II. Data set

A. Collecting data

The data set consists of manually taken images of wild birds in flight. We took all the images at the test location during one year in all seasons. The wind turbine swept area is a suitable altitude level constraint for taking the images. No normalization is applied to the images even though the sky at the background can be concolorous or multicoloured. The location of the prototype system compared to the location of the pilot wind turbine acts as constrain to the photographing distance. This results in the distance of 300m - 1000m. We tried to photograph all the birds that were flying in the suitable distance and in the correct altitude level. The number of training examples per class set at low level in terms of deep learning requirements, and therefore data expansion is required. The number of training examples per class varies roughly from 100 to 6000. To assign the same number to each class the minimum is determinative. We have used 6 as the number of classes (which includes both key species) due to difficulties in obtaining large numbers of images for all possible classes.

B. Expanding the training data

Data expansion is accomplished by applying the following algorithm: for all the images in each training set, convert the image to different color temperatures between $2000K^\circ$ and $15000K^\circ$ with step size s, where s ϵ {50, 75, 100, 150, 200, 250, 300, 1000} [14–16]. This makes the training set significantly larger, e.g. if s is 200, the smallest number of training example per class of 631 becomes $66{*}631{+}1 = 41647$. Rotate images by random angle between -30°and 30°drawn from the uniform distribution after completion of the conversion.

III. The system

We took series of images of a single target bird and each image is processed according to the schematic diagram of the system in the Fig. 1.

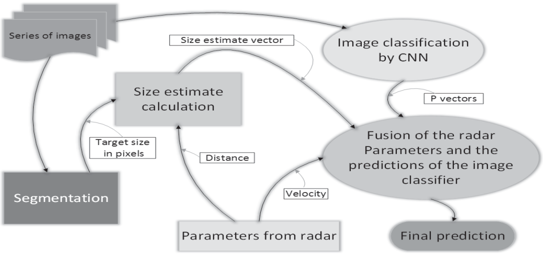


Figure 1: Schematic diagram of the system.

A. Segmentation

Segmentation is needed for the target size (in pixels) calculation. We studied methods from simple threshold to fuzzy logic for solving the problem at hand i.e. a dark figure against bright background and vice versa as well. At the extremity, the background and the target can share several colors in RGB space. We achieved the best result, in terms of signal-to-noise ratio, by applying fuzzy logic segmentation [1]. We applied Mamdani's fuzzy inference method in the proposed system [2].

B. Size estimate

The size estimate is calculated for the target bird in each image in a series as follows. The frame size ([*x y*], *in pixels*) of the camera and the angle of view (α) of the lens are known. The distance (*d*) to the target bird is provided by radar. The numbers of maximum *pixels*, horizontally (σ_z) and vertically (σ_v), of the target bird are calculated from the segmented image, respectively. The angle of view, *b*, at the distance, *d*, is calculated over a right-angled triangle (see Fig. 2). The horizontal number of *pixels per meter* is:

$$\rho_h = \frac{x}{b_h}, \tag{1}$$

and the vertical number of *pixels per meter* is:

$$\rho_v = \frac{x}{b_v}. \tag{2}$$

The size estimate for the bird in a single image in *square meters* as an area of rectangle is:

$$e = \frac{\sigma_h}{\rho_h} * \frac{\sigma_v}{\rho_v}. \tag{3}$$

The size estimate is presented as a vector in which the vector elements are placed according to class order i.e. *class 1, class 2 … class n*. The composition of the vector is following: calculate the average of the size estimates of the image series, check from the size-look-up table for which classes the average size fits, turn those elements to one and set the others to zero, yielding

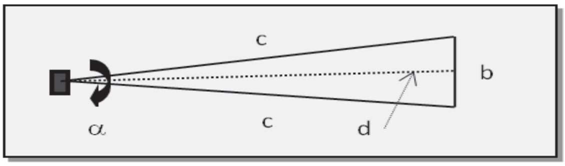


Figure 2: Diagram referred to the size estimate calculation.

$$\boldsymbol{E} = [e1, e2, \dots en], \tag{4}$$

with elements:

$$e_j = \begin{cases} 1, & \text{if } e \text{ fits class } j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

IV. Methods

A. Convolutional neural network

We apply convolutional neural network (CNN) for feature extraction. The network architecture is presented in the Fig. 3. The input image is normalized and zero-centered before entering it to the network. CNN with Mini-batch training (i.e. a hybrid between batch and on-line training) and supervised mode as well as stochastic gradient descent with momentum is applied [8], [10–12]. The L2 Regularization (i.e. weight decay) method for reducing over-fitting is also applied [3], [11], [12]. Due to limited capability to collect training data the network size in terms of free parameters is kept small, thus resulting in total of 92 feature maps which are extracted by convolution layers of sizes 12x12x3x12, 3x3x12x16 and 3x3x16x64, respectively. Total number of weights is as rounded to 9.47*10^6, which is over 14 times smaller than e.g. VGGNet [13] in which the rounded number is 138*10^6. In VGGNet the number of classes is 1000 but if we set it to 1000 in our net it would only raise the number of weights to 9.53*10^6.

Each convolution layer is followed by a Rectified Linear Units (ReLU) nonlinearity layer [9], which simply applies a threshold operation to all the components of its input:

$$f(x) = \begin{cases} 0, & x < 0, \\ x, & x \geq 0. \end{cases} \tag{6}$$

This non-saturating non-linearity in deep CNN makes the training several times faster with the hyperbolic tangent sigmoid transfer function (i.e. the transfer function we applied) [5], [9]. Cross Channel Normalization layers follow the first and the second ReLU layers. These layers aid the generalization as their function may be seen as brightness normalization [5].

The purpose of max-pooling layer is to build robustness to small distortions. This is achievable by filtering over local neighbourhoods as follows: divide the input into rectangular pooling regions, and compute the maximum of each region, thus performing down sampling and reducing the over-fitting as well [6].

There are three fully-connected layers at the end of the network for making final non-linear combinations of features, and prediction by the last fully-connected layer followed by softmax activation which produces a distribution over the class labels with cross entropy loss function [12].

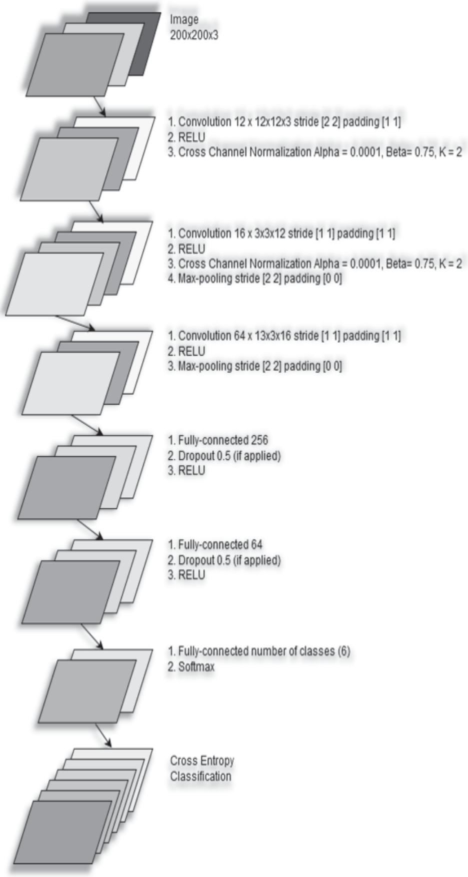


Image
200x200x3
1. Convolution 12 x 12x12x3 stride [2 2] padding [1 1]
2. RELU
3. Cross Channel Normalization Alpha = 0.0001, Beta= 0.75, K = 2
1. Convolution 16 x 3x3x12 stride [1 1] padding [1 1]
2. RELU
3. Cross Channel Normalization Alpha = 0.0001, Beta= 0.75, K = 2
4. Max-pooling stride [2 2] padding [0 0]
1. Convolution 64 x 13x3x16 stride [1 1] padding [1 1]
2. RELU
3. Max-pooling stride [2 2] padding [0 0]
1. Fully-connected 256
2. Dropout 0.5 (if applied)
3. RELU
1. Fully-connected 64
2. Dropout 0.5 (if applied)
3. RELU
1. Fully-connected number of classes (6)
2. Softmax
Cross Entropy
Classification

Figure 3: The architecture of the convolutional neural network results in (200-12+2*1)/2+1=96 neurons in each feature map of the first convolution layer thus making the number of neurons for the layer to be 96*96*12=110592. The number of neurons for the next layers is 147456, 36864, 147456, 36864, 256, 64 and 6 (i.e. the number of classes), respectively. Note that there is no max-pooling layer between the first and the second convolution layers.

B. Hyper-parameter optimization

We used manual tuning (also known as expert tuning) as hyper-parameter optimization method for number of epochs and learning rate decay schedule (LRDS). We kept all the other hyper-parameters at fixed values. We draw initial weights for all layers from the Gaussian distribution with mean 0 and standard deviation 0.01. Initial biases are set to zero. Initial learning rate is set to 0.01 and we dropped it by a factor 0.1 after different number of epochs (i.e. learning rate decay schedule). L2 value is set to 0.0005 and mini-batch size is set to 128.

C. Dropout

We applied the dropout technique for improving the performance of our CNN i.e. we trained multiple models with and without the dropout to study the contribution of this method to the classification performance by reducing over-fitting [4], [5].

D. Classification

A two-step learning method is applied i.e. a CNN is trained with the first N-1 layers viewed as feature vectors and these feature vectors are used to train a Support Vector Machine (SVM) classifier [7]. The SVM makes use of one-versus-all binary learners, in which for each binary learner, one class is positive and the rest are negative. The total number of the binary learners is the same as the number of classes. A linear classification model is applied with stochastic gradient descent (mini-batch size = 10) and the Hinge loss function with regularization term 1/*n*, where *n* is a number of training examples [17], [18]. Because of the joint performance of the CNN and the SVM classifier is mainly dependent of the features extracted by the CNN we set all hyper-parameters to their conventional values in the SVM classifier.

Final classification is obtained by a fusion between parameters provided by the radar and predictions from the image classifier. Prediction per image, $\boldsymbol{P}_i$, is combined into a vector:

$$\boldsymbol{P}_i = [c1, c2, \dots cn], i = 1, \dots n, \tag{7}$$

where c_j is a probability for *class j* and *n* is the number of images in each series. Velocity of the target bird is also a parameter provided by the radar and it is presented as a vector. It is composed in similar way than the vector ***E*** in (4) i.e. check from the velocity-look-up table for which classes the provided velocity, v_p, fits and turn those elements to one and the others to zero.

$$\boldsymbol{V} = [v1, v2, \dots vn], \tag{8}$$

with elements:

$$v_j = \begin{cases} 1, & \text{if } v_p \text{ fits class } j, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The fusion vector is:

$$\Phi = \sum_{i=1}^{n} \boldsymbol{P}_i .* \boldsymbol{V} .* \boldsymbol{E}, i = 1, \dots n \tag{10}$$

where ".*" denotes element wise multiplication and *n* is the number of images in each series. The score, S_f, for final prediction is:

$$[Sf, j] = \mathbf{argmax}_j(\Phi), \tag{11}$$

where *j* is the index of the predicted class.

V. Results

We trained several models from original image set by empirically changing hyper-parameter values. The average classification performance for the models based on the original set, in terms of true positive rate (TPR), is *0.58* tested during training. We trained also several expanded models based on different step sizes of the color conversion resulting in training set sizes between $0.53*10^5$ - $9.88*10^5$. Classification performance measured as TPR of these models set between *0.86-0.99*, which shows clear improvement as expanded training set size increases and especially compared to the models of the original set. We also tested generalization by feeding the models with images they have never seen. According to this test the system achieves its state-of-the-art performance of *0.91* (as TPR) with the expanded training set of size $4.96*10^5$ (i.e. the color conversion step size *100*), number of epochs *12*, LRDS *7* and the dropout applied. This performance level was also achieved but not exceeded by larger expanded training set. The effect of data expansion on classification performance during training and by generalization test is presented in Fig. 4. The receiver operating characteristic (ROC) for 7 models based on the original set and for 7 models based on the best expanded set is presented in Fig. 5 in which the White-tailed Eagle is plotted against all the other species (i.e. the 5 other classes) in both sets of models.

VI. Conclusions

We assembled the non-deep (i.e. in terms of weight layers) convolutional neural network for image classification, and demonstrated that the model is suitable for real world application, especially, when the number of training data is limited. We presented and demonstrated the data expansion technique that improves significantly the performance of a classifier based on small convolutional neural network. We also showed that the data expansion is crucial for the classification performance. We demonstrated that a small convolutional neural network applied with our data expansion for the training data achieved the desirable state-of-the-art performance as an image classifier. We also showed that our model generalize well to images never seen before and it is applicable for real world problem.

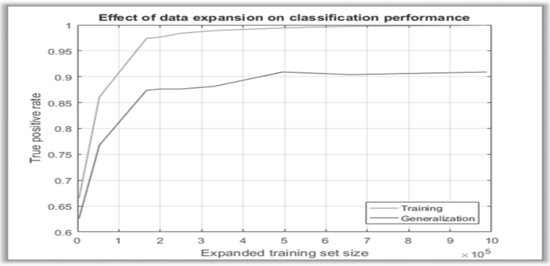


Figure 4: Effect of data expansion starting from the original training set. The red curve is for testing during training and the blue curve is according to the generalization test.

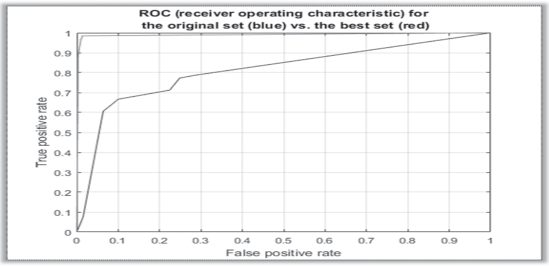


Figure 5: ROC curve for the 7 original models and the best 7 expanded models according to the generalization test. Area under the curve (AUC) is *0.84* and *0.99*, respectively.

Obviously, the parameters supplied by the radar provide additional and relevant a-priori knowledge to the system and can turn a misclassified (by images) class into the correct one.

References

[1] The MathWorks, Inc, "Fuzzy Logic Toolbox documentation," https://se.mathworks.com/help/fuzzy/fuzzy.pdf

[2] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

[3] Haykin, S., "Neural networks. A comprehensive foundation," 2. ed, New York: Prentice Hall/Pearson, 1994, p. 470.

[4] Srivastave, N., Hinton, G. E., Krizhevsky, A. Sutskever, I. Salakhutdinov R., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research. Vol. 15, pp. 1929-1958, 2014.

[5] Krizhevsky, A., Sutskever, I., Hinton, G. E., "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, 2012.

[6] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. "What is the best multi-stage architecture for object recognition?," In International Conference on Computer Vision, pp. 2146–2153, IEEE, 2009.

[7] Huang, J. F., LeCun, Y., "Large-scale learning with svm and convolutional nets for generic object categorization," InProc. Computer Vision and Pattern Recognition Conference(CVPR'06), IEEE Press, 2006.

[8] LeCun Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 86(11), pp. 2278–2324, 1998.

[9] Nair, V., Hinton, G.E., "Rectified linear units improve restricted boltzmann machines, " In Proc. 27th International Conference on Machine Learning, 2010.

[10] Li, M., Zhang, T., Chen, Y., Smola, A. J. "Efficient Mini-batch Training for Stochastic Optimization," KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge, New York, USA, 2014.

[11] Murphy, K. P., "Machine Learning: A Probabilistic Perspective," The MIT Press, Cambridge, Massachusetts, 2012.

[12] Bishop, C. M., "Pattern Recognition and Machine Learning, " 1. ed., New York: Springer, 2006, p. 144, pp. 206-209, p. 240.

[13] Simonyan K., Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv technical report, 2014.

[14] Wyszecki, G., Stiles, W. S., "Color Science: Concepts and Methods, Quantitative Data and Formulae," 2. ed., Wiley-Interscience Publication, 1982, pp. 130-170.

[15] Stiles, W. S., Burch, J. M., "NPL colour-matching investigation: Final report," Optica Acta, 6, pp. 1-26, 1959.

[16] Charity M., "What color is a blackbody? – some pixel rgb values," http://www.vendian.org/mncharity/dir3/blackbody/.

[17] Moore, R. C.; DeNero, J., "L1 and L2 regularization for multiclass hinge loss models," Proc. Symp. on Machine Learning in Speech and Language Processing, 2011.

[18] Duan, K. B.; Keerthi, S. S., "Which Is the Best Multiclass SVM Method? An Empirical Study," Multiple Classifier Systems LNCS. 3541, pp. 278-285, 2005.

PUBLICATION

II

Automatic Bird Identification for Offshore Wind Farms, In Wind Energy and Wildlife Impacts

J. Niemi and J. T. Tanttu

Wind Energy and Wildlife Impacts. Ed. by R. Bispo, J. Bernardino, H. Coelho and
J. L. Costa. 2019, 135–151. ISBN 978-3-030-05519-6

Automatic Bird Identification for Offshore Wind Farms

Juha Niemi ✉ and Juha T. Tanttu ✉

Tampere University of Technology, Signal Processing Laboratory,
Pori, Finland

Abstract. There is a need for automatic bird identification system at offshore wind farms in Finland. The developed system should be able to operate from onshore, which is cost-effective in terms of installations and maintenance. Indubitably, a radar is the obvious choice to detect flying birds but external information is required for actual identification. A conceivable method is to exploit visual camera images. In the proposed system the radar detects birds and provides the coordinates to camera steering system. The camera steering system tracks the flying birds, thus enabling capturing a series of images. Classification is based on the images and it is implemented by a small convolutional neural network trained with a deep learning algorithm. We also propose a data augmentation method in which images are rotated and converted in accordance with the desired color temperatures. The final identification is based on a fusion of data provided by the radar and image data. We present the results of the number of correctly identified species based on manually taken images.

Keywords: Image classification, Deep learning, Convolutional neural networks, Machine learning, Data augmentation

1 Introduction

The first offshore wind farm is under construction on the Finnish west coast. The minimal demand of the environmental license defines that bird species in the turbine areas are monitored and possible collisions prevented concerning especially two species: the White-tailed Eagle, *Haliaeetus albicilla* and the Lesser Black-backed Gull, *Larus fuscus fuscus*. At present, the radar system controls directly the pilot wind turbine and shut it down when any bird flies into a perimeter of 300 m to the wind turbine, which is the minimum distance in terms to have sufficient time to shut down the turbine. The number of fast restarting of the turbines should stay as low as possible due to wearing of the mechanics and therefore this operation should be used as a last resort. A solution to this is a suitable deterrent method but it is difficult to find only one applicable deterrence for all bird species in the wind farm area, and one extra problem is that the breeding birds may quicly become accustomed to e.g. sounds as a deterrent method [1]. At first stage, an automatically operating bird species identification system is needed in order to be able to develop such deterrent system. The final objective is to develop a deterrent system that operates on species-group level (i.e. a different deterrent method for e.g. gulls and eagles). At this point our main research question is how to identify bird species in flight automatically in real-time?

It makes sense, in order to implement this system cost-effectively, to build only one control system in such a location from where it is possible to monitor birds in the vicinity of all wind turbines of the wind farm. To achieve this goal it has been decided to place the control system onshore as it is more cost-effective compared to a system installed offshore. The distance from the chosen location to the monitored birds (i.e. the vicinities of the future wind turbines) ranges between 500 m and 1500 m. We initially considered a radar for both detecting the bird and identifying its species. However, it is known that the identification capacity is limited, rendering impossible to classify bird species any further merely by this radar system. Obviously, external information is required and a conceivable method is to exploit visual camera images. A digital single-lens reflex (DSLR) camera with telephoto lens is applied due to the long photographing distance.

2 The System

The proposed system consists of several hardware as well as software modules. See the Fig. 1 for an illustration. First of all there is the radar which is connected to a local area network (LAN) and thus it is able to communicate with the servers in which the various programs are running. The most important role of the radar is to detect flying birds but it also provides some parameters for bird identification (i.e. classification). The parameters are: the distance in 3D of a target (m), velocity of a target (m/s) and trajectory of a target (WGS84 coordinates). The distance of a target is used to estimate the size in meters. Velocity of a target is used for the final classification, which we call the fusion. The system also includes the aforementioned camera with the lens and a motorized video head. The video head is operated by Pelco-D control protocol [2] and the control software for it is developed by us. The camera is controlled by the application programmable interface (API) of the camera manufacturer and the software for controlling the camera is also developed by us. The system has three servers: the radar server, the video head steering server and the camera control server. Software for the radar server is supplied by the manufacturer of the radar but the software for the other two servers is result of our development work.

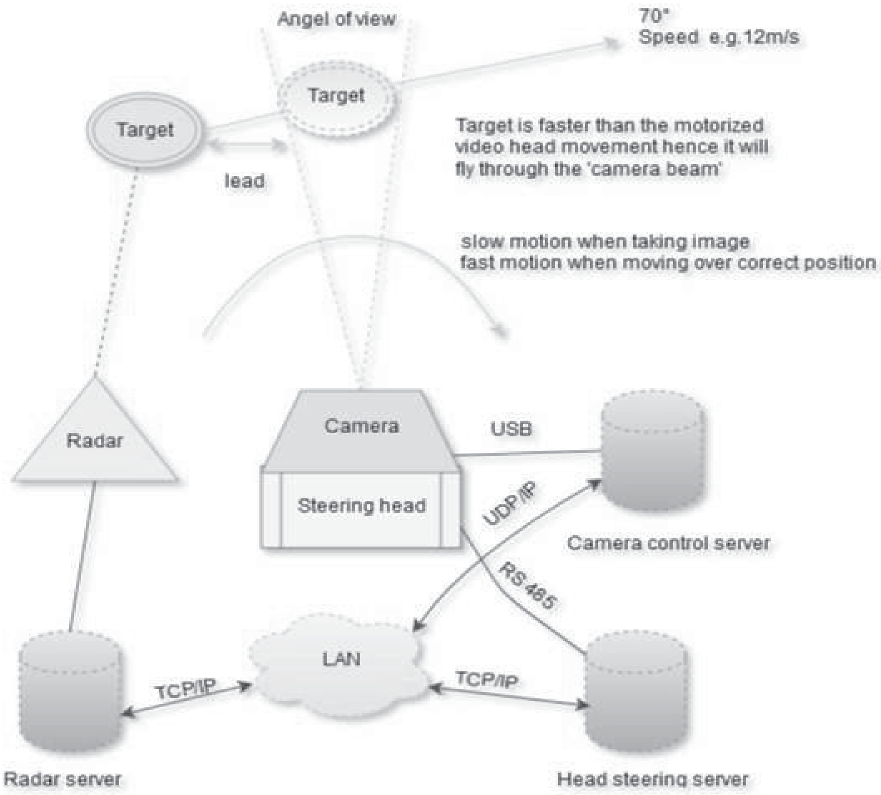

Angel of view
70°
Speed e.g.12m/s
Target
Target
lead
Target is faster than the motorized video head movement hence it will fly through the 'camera beam'
slow motion when taking image
fast motion when moving over correct position
Radar
Camera
Steering head
USB
UDP/IP
Camera control server
RS 485
LAN
TCP/IP
TCP/IP
Radar server
Head steering server

Fig. 1. The hardware of the system and the principle of catching flying bird into the frame area of the camera.

2.1 Hardware Level

The figure 1 illustrates the hardware components of the system. The operation on hardware level commences when the radar detects a bird and sends information of the blip including the parameters to the video head steering server. The video head steering server reads the coordinates from the data sent by the radar. The system reacts only if the data has the altitude information of the object.

We have used the PT-1020 Medium Duty video head of the 2B Security Systems. This head has five different speeds of panning: 12, 18, 24, 32 and 48 as well as two of tilting: 12 and 18, in degrees/s. The maximum speed is 1499 m/s (i.e. 48 degree/s) at the distance of 1350 m, where the pilot turbine is located, and the minimum speed is 287 in m/s (i.e. 12 in degree/s) . The maximum ground speed of a flying bird in the test area is estimated to be 110 km/h = 30.5 m/s [3] added with the maximum average wind of 30 m/s [4] resulting in 60.5 m/s. The minimum ground speed of a flying bird in the test area is roughly 6 m/s, which is based on the radar. Soaring gulls and terns are not included in the minimum speed measure. The calculations show that the maximum speed of the head is sufficient but the minimum speed is too fast for direct tracking.

The too-fast-minimum-speed problem (i.e. the minimum speed of steering the video head is not slow enough in order to track a flying bird at desired distance) is solved as follows: the video head steering system calculates the most probable trajectory of the target bird and adds a lead to the calculated position so that when the head will be driven to that final position it would be ahead of the target bird. The calculations are based on the flight speed (measured by the radar) of a target and the right-angled triangle and they ignore any possible sudden deviations off the flight path. The curvature of the horizon is also ignored due to the relatively short distances in question. When the final position is reached the head stops. This solves the minimum speed problem and also maximizes the probability that the target bird is within the frame of the camera at distance in question, see the Fig. 1. After the head is driven to the final position the camera takes series of images and sends them to the classification software that runs on the same server.

Radar. We use a radar system supplied by Robin Radar Systems B.V. because they provide an avian radar system that is able to detect birds. They also have tracker algorithms for tracking a detected object over time i.e. between the blips. The model we use is the ROBIN 3D FLEX v1.6.3 and it is actually a combination of two radars and a software package for implementation of various algorithms such as the tracker algorithms. The two radars are: a horizontal scanning S-band radar, and a 3D tracking frequency-modulated continuous wave (FMCW) vertical X-band radar. The 3D tracking FMCW radar supports 2-axis scanning mode for 3D coverage and tracking mode with either manual or automatic track selection. The ROBIN 3D FLEX v1.6.3 radar system is capable to provide parameters, such as velocity and bearing of the detected object, for our system. The S-band radar enables a long range detection up to 10 km of flying birds and it provides the longitude and the latitude of a target bird. The X-band radar enables higher resolution i.e. it can detect smaller objects such as small birds up to distance of 5 km. The X-band radar also provides the altitude of the target [5,6,7].

The S-band radar can operate the whole 360 degrees but it is adjusted to operate 180 degrees in the test site since the objects are always in the sector of south via west to north. The X-band radar operates roughly in 20 degree sector that can be configured to lie in a constant position or it can be configured to multiplex between two separate positions. The two possible operation modes of the X-band radar are presented in the Fig. 2. and the Fig. 3., respectively. The ability to multiplex is important because it enables to adjust the minimum distance to the wind turbine of the approaching bird and therefore it gives sufficient time to shut down the wind turbine. The cross section of the two radar beams is the key feature since the intersection of this is the area from where all the three coordinates (i.e. latitude, longitude and altitude) are available [5,6,7]. The proposed system can not work without information about the altitude of the object.

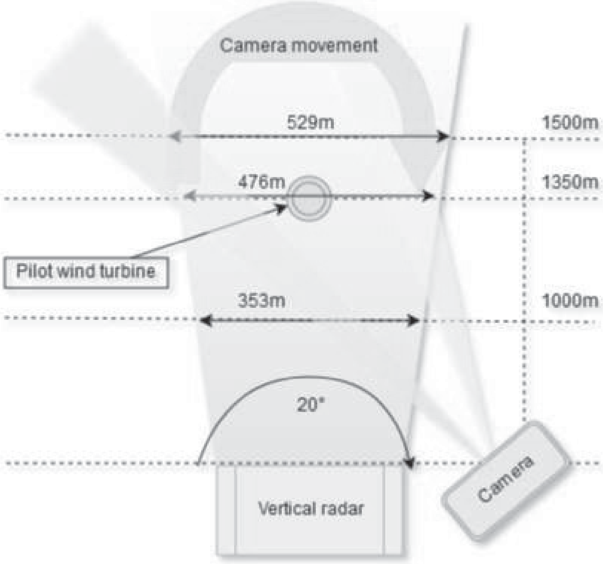


Fig. 2. The vertical X-band radar operation at a constant position.

Camera System. The resolution of the camera sensor measured by the total number of pixels and the focal length of the lens are important qualities because of the long distance to birds of which images are to be taken. We use a term

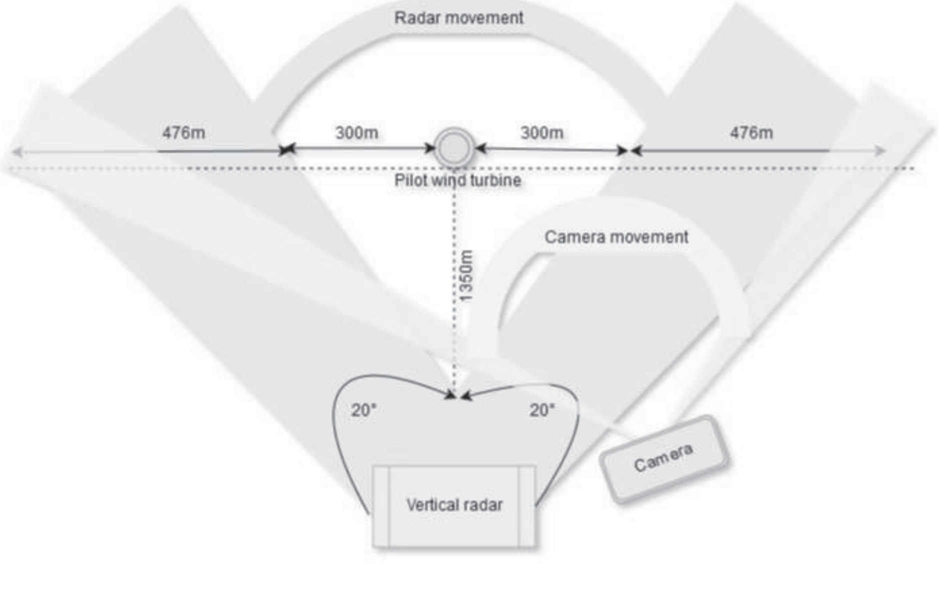


Fig. 3. The vertical X-band radar operation when configured to multiplex between two separate positions.

called the effective number of pixels (ENP) defined by the number of pixels representing a bird. The remaining number of pixels are considered noise. Image classification is achieved only with ENP as birds will be very small (i.e. they consist of only a small number of pixels) in the images. ENP depends on the focal length of the lens and can be increased by choosing a long (i.e. in terms of focal length) telephoto lens. Because of these facts we have chosen to use the Canon EOS 7D II camera with 20.2-megapixels sensor and the Canon EF 500/f4 IS lens. Correct focusing of the images relies on the autofocus system of the lens and the camera. Automatic exposure is also applied. The operation of the proposed system is not restricted to this combination of the camera and the lens but a combination of any standard DSLR camera with any standard lens suitable for that camera can be utilized.

2.2 Software Level

The ability of the system to identify bird species depends on the image classification and the size and velocity estimates of the target bird. Figure 4 illustrates the work flow for processing and classifying images. The classification process begins with a series of images taken of a single bird (i.e. a bird individual or a tight flock of birds flying past). The purpose of taking several images is simply to increase the probability of a correct identification. The images in a single series are fed to the classifier separately.

Segmentation is computed in parallel to image classification in order to obtain an estimate of the target bird size in pixels. We tested several methods from simple threshold to fuzzy logic for solving the problem at hand, i.e., a dark

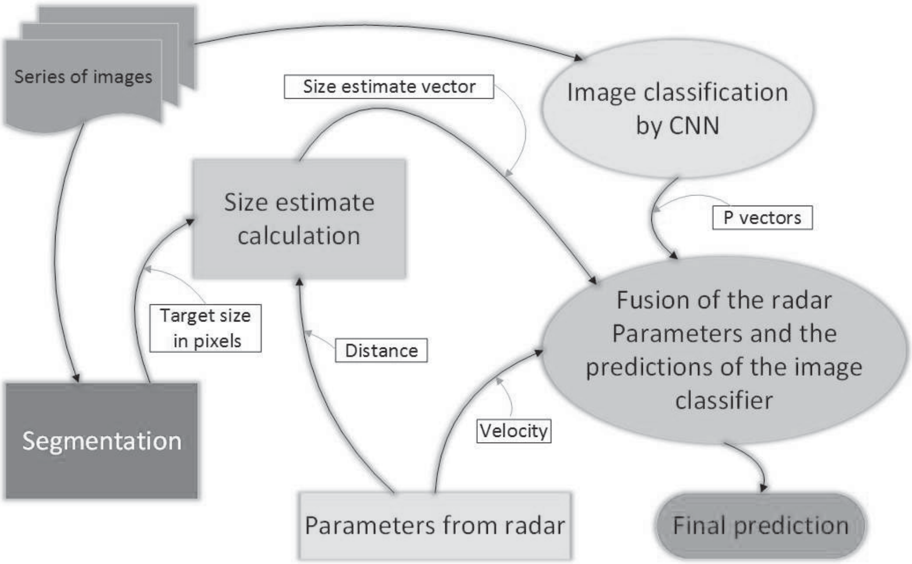


Fig. 4. The work flow for processing and classifying images.

figure against bright background and *vice versa*. The background and the target can share several colors in the RGB color space. We achieved the best results, in terms of signal-to-noise ratio, by applying fuzzy logic segmentation [8]. In particular, we applied Mamdani's fuzzy inference method [9].

Once the target bird size in pixels is obtained the target bird size in meters can be estimated. The image classifier results are given in $\boldsymbol{P}$-vectors, which elements are probabilities of belonging to each class. There is one $\boldsymbol{P}$-vector for each image. The $\boldsymbol{P}$-vectors for a single series of images are combined to the final $\boldsymbol{P}$-vector by adding element-wisely. A size estimate and velocity of a target bird are also represented as vectors. Finally, the fusion is simply an element-wise multiplication of the size vector, the velocity vector and the combined $\boldsymbol{P}$-vector.

2.3 Data

Input data for the identification system consist of digital images and parameters from the radar. The parameters from the radar are real numbers such as velocity of a flying bird in m/s and bearing (i.e. a heading: the horizontal angle between the direction of an object and that of true north) in degrees.

Deep learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning networks are typically large, i.e. they have many layers and large number of parameters. A convolutional neural network (CNN) is one implementation of deep learning [10]. A CNN is a specialized kind of neural network for processing data that has a known grid-like topology like image data, which can be thought of as a 2D grid of pixels. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. Typically CNNs need large datasets for the training phase to achieve sufficient classification performance [11].

All images for training the CNN have been taken manually at the test location and in various weather conditions. The location is the same where the camera will be installed for taking images automatically. There are also constraints concerning the area where the images have to be taken. Here the area refers to the air space in the vicinity of the pilot wind turbine. We have used the swept area (i.e. 130 m) as a suitable altitude level constraint for taking the images. At this stage, the images are only need to be taken in the vicinity of 1350 m in lengthwise direction, which is the distance to the pilot wind turbine. No normalization is applied to the images even though the sky at the background can be concolorous or multicoloured.

The number of training examples for each image class (i.e. bird species) varies from 1164 to 5614. This is not enough (i.e. the demand is typically from tens of thousands to hundreds of thousands) for training of the deep learning algorithm [10], and therefore data augmentation (a.k.a. data expansion) is required. Data augmentation refers to any method that makes the original data set larger. These methods may include noise addition or various transformations such as rotation of images [12]. The number of images of each class should be the same as a CNN is applied and therefore the lowest number of images of the classes is used. We have restricted the number of classes (which includes both key species) to 6 due to difficulties in achieving large numbers of images for all possible classes in a relatively short period of time. The 6 initial classes for training the CNN are the Common Goldeneye (*Bucephala clangula*), the White-tailed Eagle (*Haliaeetus albicilla*), the Herring Gull (*Larus argentatus*), the Lesser Black-backed Gull (*Larus fuscus fuscus*), the Great Cormorant (*Phalacrocorax carbo*) and Common/Arctic Tern (*Sterna hirundo/paradisaea*). These species are chosen simply because we were able to collect the largest numbers of images of them. We are able to increase the number of classes (i.e. species) as the number of images is increasing in those classes (i.e. other than the aforementioned 6) that have only minor number of images at present. The final classes will also include the Common Scoter (*Melanitta nigra*), the Velvet Scoter (*Melanitta fusca*), the Common Eider (*Somateria mollissima*), the Common Gull(*Larus canus*) and the Black-headed Gull (*Larus ridibundus*).

Data Augmentation. Our system is operating in marine environment and therefore prevailing weather has significant influence to the tonality of the images taken at the test site. It is intuitively obvious that the lighting will vary with time, and thus the toning of the images will be changing according to lighting.

Color (in K, Kelvin degrees) temperature is a property of a light source. It is the temperature of the ideal black-body radiator that radiates light of the same color as the corresponding light source. In this context black-body radiation is the thermal electromagnetic radiation emitted by a black body. A black-body is an opaque and non-reflective body. It has a specific spectrum and intensity that depends only on the temperature of the black-body, and it is assumed to be uniform and constant. In our case the light source is the sun that closely approximates a black-body radiator. Even though the color of the sun may appear different depending on its position, the changing of color is mainly due to the scattering of light and it is not because of the changes in the black-body radiation [13,14,15,16]. Color matching functions (CMFs) provide the absolute energy values of three primary colors which appear the same as each spectrum color. We applied the International Commission on Illumination (*Commission internationale de l'clairage*, CIE) 10-deg color matching functions in our data augmentation algorithm [17].

The augmentation is done according to the curves in the Fig. 5. [18] by converting an image into different color temperatures between *2000* K and 15000 K with step size s, where $s \epsilon \{25, 50, 75, 100, 150, 200, 250, 300, 1000\}$. This makes the training set significantly larger, e.g., if s is 200, the class with the smallest number of training examples of 1164 becomes $67 \cdot 1164 = 77988$, where $67 = (15000 - 2000)/200 + 2$ (i.e the difference between 15000 and 2000 divided by the step size plus the two extremes). When conversion is done, the images are also rotated by a random angle between -30 degree and 30 degree drawn from the uniform distribution. Motivation for this is that a CNN is invariant to small translations but not rotation of an image [19].

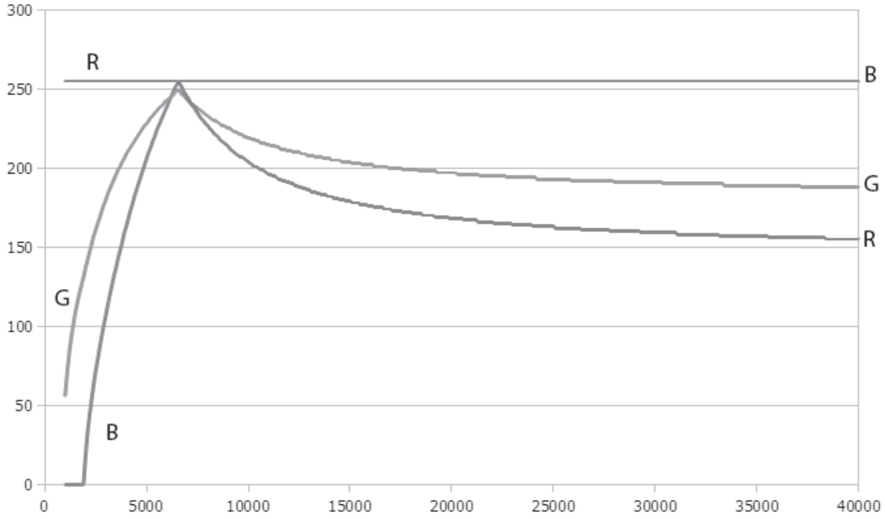


Fig. 5. Color temperature and corresponding RGB values presented according to CIE 1964 10-degree color matching function.

3 Classification

3.1 Target Size Estimate and Velocity

In this subsection we describe how the size of the target bird is estimated. The frame size ([width x height y], in *pixels*) of the camera and the angle of view (α) of the lens are known. The distance (d) to the target bird is provided by the radar. The maximum number of horizontal (σ_h) and vertical (σ_v) *pixels* of the target bird are calculated from the segmented image, respectively. The angle of view, b, at the distance, d, is calculated over a right-angled triangle, see the Fig. 6. The horizontal number of *pixels/meter* is given by

$$\rho_h = \frac{x}{b_h}, \tag{1}$$

and the vertical number of *pixels/meter* by

$$\rho_v = \frac{y}{b_v}, \tag{2}$$

where, b_h and b_v denote the horizontal and the vertical angles of view, respectively. The estimate for the size of the bird in a single image in *square meters* as an area of rectangle is:

$$e = \frac{\sigma_h}{\rho_h} * \frac{\sigma_v}{\rho_v}. \tag{3}$$

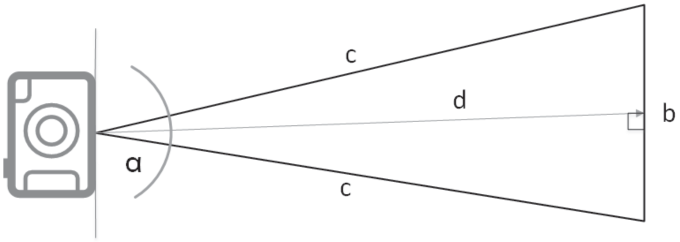


Fig. 6. Diagram of the size estimate calculation.

The size estimate is presented as a vector with elements placed according to the class order (ordered by the probabilities), i.e., *class 1, class 2, class nc*, where *nc* denotes the number of classes. The composition of the vector is following: calculate the average of the size estimates of the image series, check from the size-look-up table for which classes the average size, e, fits, turn those elements to one and set the others to zero, yielding

$$\boldsymbol{E} = [e_1, e_2, ..., e_{\mathrm{nc}}], \tag{4}$$

with elements:

$$e_j = \begin{cases} 1, & \text{if } e \text{ fits class } j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The velocity of the target bird is a parameter provided by the radar also presented as a vector. It is composed in similar way than the $\boldsymbol{E}$-vector in (4) i.e. check from the velocity-look-up table for which classes the provided velocity, v, fits and turn those elements to one and the others to zero.

$$\boldsymbol{V} = [v_1, v_2, ..., v_{\mathrm{nc}}], \tag{6}$$

with elements:

$$v_j = \begin{cases} 1, & \text{if } v \text{ fits class } j, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

3.2 Image Classification

Figure 7 illustrates image classification. We applied a CNN for feature extraction and built an architecture of three convolution layers of which each are followed by rectifier layers. The first two layers are followed by cross channel normalization layers and the second and third layer is followed by max-pooling layer. Finally, three consecutive fully-connected layers are at the end of the CNN network. The dropout technique is applied to the first and second fully-connected layers.

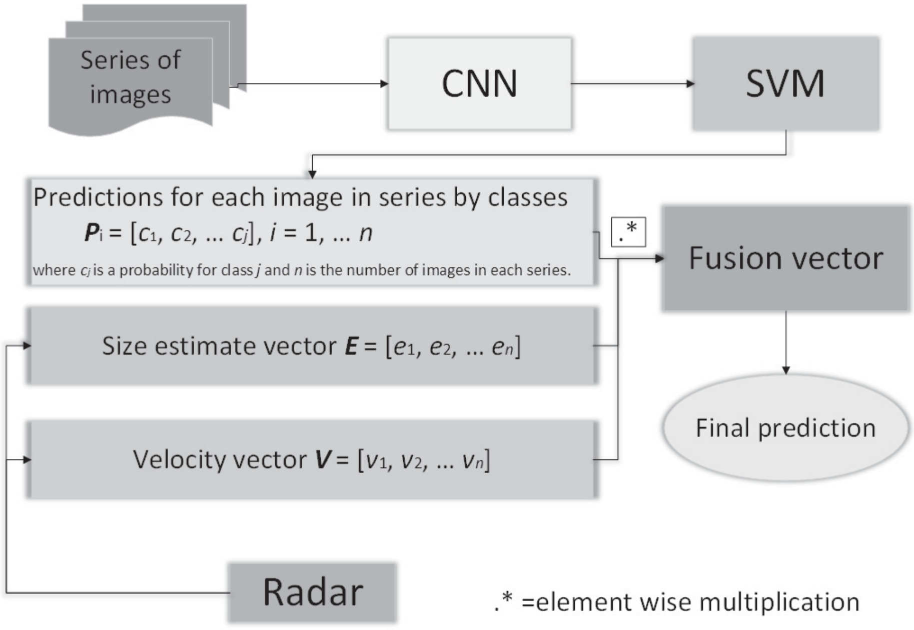

Series of images
CNN
SVM
Predictions for each image in series by classes
$P_i = [c_1, c_2, \ldots c_j], i = 1, \ldots n$
where c_j is a probability for class j and n is the number of images in each series.
.*
Fusion vector
Size estimate vector $E = [e_1, e_2, \ldots e_n]$
Velocity vector $V = [v_1, v_2, \ldots v_n]$
Final prediction
Radar
.* =element wise multiplication

Fig. 7. The classification process.

This architecture may be seen as a small CNN because it is actually non-deep compared with conventional deep learning models. [20]

We applied a two-step learning method i.e. the CNN is trained with the first N-1 layers (i.e. N is the number of the layers) viewed as feature vectors and a combined result is based on these feature vectors that is used to train a Support Vector Machine (SVM) classifier [21]. The SVM classifier is thus the actual classifier [20].

The result of the image classifier is presented as a vector, thus predictions for each image in a single series of images are combined into the vector, $\boldsymbol{P}_i$:

$$\boldsymbol{P}_i = [c_1, c_2, ..., c_{\mathrm{nc}}], i = 1, ..., n, \tag{8}$$

where c_j is a probability of belonging to *class j*, *nc* is the number of classes and n is the number of images in each series.

3.3 Fusion

The final classification is achieved by a fusion between the parameters provided by the radar and the predictions from the image classifier. The combined $\boldsymbol{P}$-vector for a series of images is:

$$\boldsymbol{P} = \sum_{i=1}^{n} \boldsymbol{P}_i, \tag{9}$$

where n is the number of images in each series and the fusion vector, Φ, is:

$$\Phi = \boldsymbol{P} .* \boldsymbol{V} .* \boldsymbol{E} \tag{10}$$

where ".*" denotes element wise multiplication. The score, S, for final prediction is:

$$S = \mathbf{max}_j(\Phi), \tag{11}$$

$$j = \mathbf{arg\ max}_j(\Phi), \tag{12}$$

where j is the index of the predicted class.

4 Results

The data augmentation algorithm proved to be slow with our computer resources when the step size, s, is less than 100. For this reason we initially trained several classifier models on the original image set (i.e. no data augmentation) and also several models on various augmented data sets with s as selected from {25, 50, 75, 100, 150, 200, 250, 300, 1000}, respectively. The original data set size was 6 ·631 = 3786 at this phase. The effect of the data augmentation on classification performance has been examined according to this data set and it is presented in the Fig. 8.

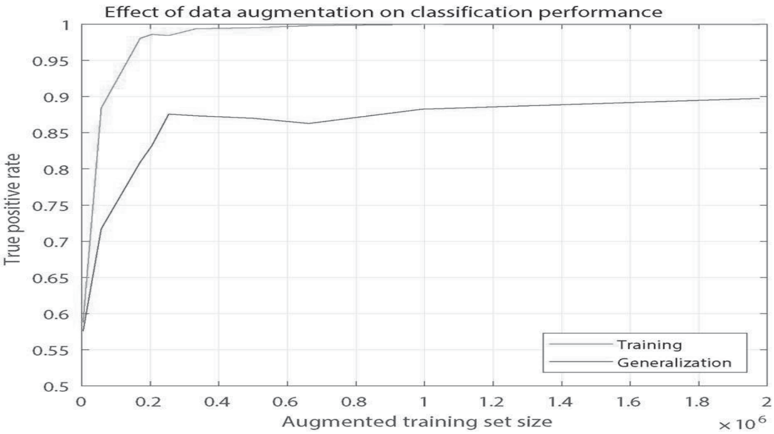


Fig. 8. The upper curve is for testing in training and the lower curve is according to the generalization test. Several models were trained for each step size, s, in the data augmentation. The average TPR values of the models with the same step size are applied in the figure. The starting value for both curves is the average value of the models trained on the first original data set.

Based on observations of the first training process it was obvious that to create augmented data sets with all previously applied values of the step size, s, is time consuming and not reasonable. It was also intuitively clear that the number of epochs should decrease when the number of training examples increases and the learning rate drop period should increase as the number of training examples increases in order to avoid overfitting. Hence we trained the following classifier models with the step side, $s = 50$, when the data augmentation was applied. The classification performance for the latest (i.e. trained on the original data set size at $6984 = 6 \cdot 1164$) models in terms of true positive rate (TPR) is as an average 0.7684 for the original data set and varies between 0.9317 and 0.9999 for the augmented data sets, which shows clear improvement as augmented training set size increases and especially compared to the models of the original data set.

Generalization was also tested based on the latest models by feeding them images they have never seen. According to this test the system achieves its state-of-the-art performance of 0.9583 (as TPR) with the augmented data set of the size $1.83 * 10^6$ (i.e. the color conversion step size 50), number of epochs 8, learning rate drop period (LRDP) 3 with 0.1 as the factor and 0.01 as the initial learning rate, and the dropout applied. See [20] for earlier results.

The results for the original data set (i.e. the data set size $6984 = 6 \cdot 1164$) are presented as a confusion matrix in the Table 1. The results for the best augmented set are presented as a confusion matrix in the Table 2. Cross-validation is applied and the split into a training set and a validation set was 70% and 30%, respectively, in the two previous data sets. The results for the generalization (the best performed model based on the augmented sets is applied) are presented as a confusion matrix in the Table 3. There were 100 randomly picked images from each 6 classes, which the tested model has never seen before. Thus the test set size in the generalization test was $6 \cdot 100 = 600$. The abbreviated class names, which are names of the bird species, in the figures are: *CG = Common Goldeneye, WTE = White-tailed Eagle, HG = Herring Gull, LBBG = Lesser Black-backed Gull, GC = Great Cormorant, C/AT = Common/Arctic Tern.* The most right-handed column, *TP/class*, is for correctly predicted images for the class (i.e. species).

Table 1. Classification performance of the original data set size at $6984 = 6 \cdot 1164$ presented as a confusion matrix.

	CG	WTE	HG	LBBG	GC	C/AT	TP/class
CG	245	4	40	28	17	15	70.2%
WTE	0	305	6	6	20	12	87.4%
HG	13	6	249	39	6	36	71.3%
LBBG	14	1	37	252	3	32	72.2%
GC	21	24	5	12	277	10	79.4%
C/AT	6	6	35	16	5	281	80.5%

Table 2. Classification performance of the best augmented data set presented as a confusion matrix.

	CG	WTE	HG	LBBG	GC	C/AT	TP/class
CG	91489	0	0	0	1	0	99.999%
WTE	0	91485	1	0	3	1	99.995%
HG	0	0	91450	4	1	35	99.956%
LBBG	0	0	1	91474	6	9	99.983%
GC	3	12	9	1	91457	8	99.964%
C/AT	0	0	6	3	2	91479	99.988%

Table 3. Classification performance tested on 100 unseen images from each class as a confusion matrix.

	CG	WTE	HG	LBBG	GC	C/AT	TP/class
CG	97	0	0	0	2	1	97%
WTE	0	99	0	1	0	0	99%
HG	0	1	98	0	0	1	98%
LBBG	0	1	5	87	1	6	87%
GC	0	2	0	1	97	0	97%
C/AT	0	0	2	0	1	97	97%

5 Discussion and Conclusions

A lot of research is done to monitor birds in wind farms, and to find suitable methods for collision detection, e.g. the WT-Bird of the Energy Research Centre of the Netherlands [22,23]. The principle of the WT-Bird system is that a bird collision is detected by the sound of the impact (triggering) and that the species can be recognised from video images [24,25]. However, it has known problems with false alarms in high wind circumstances concerning larger bird species and it has no automated species identification algorithm [26]. We are also aware of two commercial systems: the DTBird of Liquen Consultora Ambiental,S.L., Spain and the MUSE of DHI, USA [27,28]. They both promise to detect birds automatically and prevent possible collisions in the vicinity of wind turbines but no detailed technical data are available. It seems according to their websites that their systems work from an individual wind turbine and they have short distance camera systems for providing images of birds. Only the MUSE seems to have automated identification of species but the methods are unknown to us. All of the three systems use video footage for detecting possible collisions and it seems that they have a video camera and recording system for each wind turbine.

The primary objective of the system here presented is to use only one camera location for monitoring and species identification but a video camera (and an infrared camera) is a noteworthy possibility for collision detection. However, we are seeking solution that operates from a single location, which is onshore. We are currently working on the collision detection problem but no collisions have been observed until now while the pilot wind turbine has been manually monitored for *18* months. It seems that collisions are quite rare in the research area and this makes the field testing of the possible collision detection methods challenging.

We proposed a novel system for automatic bird identification as a real world application. However, the system has restrictions such as the background of the images is the sky (we have not tested images taken otherwise) and images can not be taken in pitch-dark or in poor visibility conditions. Infrared cameras may contribute to the collision detection but their contribution to classification is poor because all color information is lost. The proposed system is still under construction and installation phase, so we have not yet been able to test the complete system.

We built and tested several image classifiers based on the small CNN trained with the deep learning algorithm. The best performed classifier proved to be discoverable by changing the step size in the data augmentation and the hyperparameter values (number of epochs and the learning rate drop period) and with or without the dropout technique applied. Our model may be seen as a non-deep or small neural network since it has only three convolution layers.

We demonstrated that the small CNN applied with our data augmentation algorithm for the training data achieved the acceptable state-of-the-art performance of *0.9583* as an image classifier. We showed that our data augmentation method is suitable for image classification problem and it significantly increases the performance of the classifier.

The measured performance of the image classifier has been obtained without the support of the parameters supplied by the radar. The parameters provide additional and relevant a-priori knowledge to the system and they can turn a misclassified (by images) class into the correct one. We also collect more data continuously at the test site and thus the number of training examples increases resulting in a better performance of the current classes, and we are able to increase the number of classes as well.

Acknowledgements

The authors wish to thank Suomen Hyötytuuli for the financial support and Robin Radar Systems for the technical support with the applied radar system.

References

1. Baxter, A. T., Robinson, A. P.: A comparison of scavenging bird deterrence techniques at UK landfill sites. International Journal of Pest Management, vol. 53, pp. 347-356. Taylor & Francis Online (2007). doi:10.1080/09670870701421444

2. pelco-D protocol. Bruxy REGNET. http://bruxy.regnet.cz/programming/rs485/pelco-d.pdf
3. All about the Peregrine falcon. U.S. Fish and Wildlife Service (1999). https://web.archive.org/web/20080416195055/http://www.fws.gov/endangered/recovery/peregrine/QandA.html#fast
4. Statistics. Finnish Meteorological Institute. http://ilmatieteenlaitos.fi/tuulitilastot
5. Robin radar models. Robin Radar Systems B.V. https://www.robinradar.com/
6. Richards, M.A.: Fundamentals of Radar Signal Processing. The McGraw-Hill companies, New York (2005). ISBN 0-07-144474-2
7. Bruderer, B: The Study of Bird Migration by Radar, part1: The Technical Basis. Naturwissenschaften 84, pp. 1-8. Springer-Verlag, Heidelberg (1997)
8. Fuzzy Logic Toolbox documentation. The MathWorks Inc. https://se.mathworks.com/help/fuzzy/fuzzy.pdf
9. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, vol. 7, No. 1, pp. 1-13. Elsevier (1975)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). www.deeplearningbook.org
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. pp. 330-371. MIT Press (2016). www.deeplearningbook.org
12. Wang, J., Perez, L.: The Effectiveness of Data Augmentation in Image Classification using Deep Learning. Stanford University, Stanford (2017). http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf
13. Speranskaya, N.I.: Determination of spectrum color co-ordinates for twenty-seven normal observers. Optics and Spectroscopy, vol. 7, pp. 424-428. Springer (1959)
14. Stiles, W.S., Burch, J.M.: NPL colour-matching investigation: Final report. Optica Acta, vol. 6, pp. 1-26. Taylor & Francis (1959)
15. Wyszecki, G., Stiles, W.S.: Color Science: concepts and methods, quantitative data and formulae. 2nd ed., Wiley, New York (1982)
16. Stockman, A., Sharpe, L.T.: Spectral sensitivities of the middle- and long-wavelength sensitive cones derived from measurements in observers of known genotype. Vision Research, vol. 40, pp. 1711-1737. Elsevier (2000)
17. CIE Proceedings, Vienna Session 1963. Committee Report E-1.4.1, vol. B, pp. 209-220. Bureau Central de la CIE, Paris (1964)
18. Blackbody color datafile. Vendian.org. http://www.vendian.org/mncharity/dir3/blackbody/UnstableURLs/bbr_color.html
19. K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun.: What is the best multi-stage architecture for object recognition. In: International Conference on Computer Vision, pp. 2146-2153. IEEE, Kyoto, Japan (2009)
20. Niemi, J., Tanttu, J.T.: Automatic Bird Identification for Offshore Wind Farms: A Case Study for Deep Learning. In: Proceedings of ELMAR-2017, 59$^{\mathrm{th}}$ IEEE International Symposium ELMAR-2017, Croatian Society Electronics in Marine, Croatia (2017). ISBN: 978-953-184-230-3
21. Huang, J.F., LeCun, Y.: Large-scale learning with svm and convolutional nets for generic object categorization. In: Computer Vision and Pattern Recognition Conference(CVPR06), pp. 284-291. IEEE, New York, NY (2006)
22. Desholm, M., Kahlert, J.: Avian Collision Risk at an Offshore Wind Farm. Biology Letters, vol. 1, pp. 296-298, the Royal Society Publishing (2005). doi:10.1098/rsbl.2005.0336

23. Marques, A.T, et al.: Understanding bird collisions at wind farms: An updated review on the causes and possible mitigation strategies. Biological Conservation, vol. 179, pp. 40-52. Elsevier (2014)
24. Verhoef, J.P., Westra, C.A., Korterink, H., Curvers, A.: WT-Bird A Novel Bird Impact Detection System. ECN research Centre of the Netherlands (2002), https://www.ecn.nl/docs/library/report/2002/rx02055.pdf
25. Wiggelinkhuizen, E.J., Barhorst, S.A.M., Rademakers, L.W.M.M., den Boon, H.J.: Bird Collision Monitoring System for Multi-Megawatt Wind Turbines, WT-Bird: Prototype development and testing. ECN research Centre of the Netherlands (2006), https://www.ecn.nl/publications/PdfFetch.aspx?nr=ECN-E--06-027
26. Wiggelinkhuizen, E.J., den Boon, H.J.: Monitoring of Bird Collisions in Wind Farm under Offshore-like Conditions using WT-BIRD System: Final report. ECN research Centre of the Netherlands (2009), https://www.ecn.nl/docs/library/report/2009/e09033.pdf
27. DTBird. Liquen Consultora Ambiental,S.L. http://www.dtbird.com/
28. MUSE. DHI. https://www.dhigroup.com/global/news/2017/02/automated-bird-monitoring-system-lands-on-pioneer-us-wind-farm

PUBLICATION

III

Deep Learning Case Study for Automatic Bird Identification
J. Niemi and J. T. Tanttu

Applied Sciences 8.11 (2018)
DOI: 10.3390/app8112089

applied sciences

MDPI

Article

Deep Learning Case Study for Automatic Bird Identification

Juha Niemi [1,*,†,‡] **and Juha T. Tanttu** [2,†]

[1] Signal Processing Laboratory, Tampere University of Technology, 28101 Pori, Finland
[2] Mathematics Laboratory, Tampere University of Technology, 28101 Pori, Finland; juha.tanttu@tut.fi
[*] Correspondence: juha.niemi@tut.fi; Tel.: +358-40-172-0209
[†] This paper is an extended version of our paper published in 2017 International Symposium ELMAR.
[‡] Current address: Tampere University of Technology, Signal Processing Laboratory, P.O. Box 300, 28101 Pori, Finland.

Received: 27 September 2018; Accepted: 23 October 2018; Published: 29 October 2018

Abstract: An automatic bird identification system is required for offshore wind farms in Finland. Indubitably, a radar is the obvious choice to detect flying birds, but external information is required for actual identification. We applied visual camera images as external data. The proposed system for automatic bird identification consists of a radar, a motorized video head and a single-lens reflex camera with a telephoto lens. A convolutional neural network trained with a deep learning algorithm is applied to the image classification. We also propose a data augmentation method in which images are rotated and converted in accordance with the desired color temperatures. The final identification is based on a fusion of parameters provided by the radar and the predictions of the image classifier. The sensitivity of this proposed system, on a dataset containing 9312 manually taken original images resulting in 2.44×10^6 augmented data set, is 0.9463 as an image classifier. The area under receiver operating characteristic curve for two key bird species is 0.9993 (the White-tailed Eagle) and 0.9496 (The Lesser Black-backed Gull), respectively. We proposed a novel system for automatic bird identification as a real world application. We demonstrated that our data augmentation method is suitable for image classification problem and it significantly increases the performance of the classifier.

Keywords: machine learning; deep learning; convolutional neural networks; classification; data augmentation; intelligent surveillance systems

1. Introduction

Several offshore wind farms are under construction on the Finnish west coast. The official environmental specifications define that bird species behaviour at the vicinity of wind turbines must be monitored. This concerns especially two species: the White-tailed Eagle (*Haliaeetus albicilla*) and the Lesser Black-backed Gull (*Larus fuscus fuscus*), which are explicitly mentioned in the environment license. The only way to fulfil this demand cost efficiently is to automate monitoring, and that requires automatic bird species identification at such a level that the aforementioned bird species are separable from all other species in the study area. The problem is how to identify bird species in flight automatically in real-time? The prototype system for automated bird identification is developed and placed at a test location on Finnish west coast. This system is still under construction.

The ultimate objective of bird monitoring in wind farms is to find suitable methods for collision detection [1,2], and especially to find possible deterrent methods [3]. The WT-Bird of the Energy Research Centre of the Netherlands is the first (i.e., known to us) published research of this subject. The principle of the WT-Bird system is that a bird collision could be detected by the sound of the impact and that the bird species can be recognised by non-real time method from video footage [4,5].

However, it has known problems with false alarms in high wind circumstances concerning larger bird species and it has no automated species identification algorithm [6].

Radar is a feasible choice for the detection of birds since the identification need is restricted to the flying birds only. If merely a radar is used, the identification capability is limited to a few size classes according to radar suppliers. Obviously, external information is required and a conceivable method is to exploit visual camera images, thus a digital single-lens reflex (DSLR) camera with a telephoto lens is applied. This paper shows that convolutional neural network (CNN) with deep learning algorithm trained on real-world images is capable to achieve sufficient state-of-the-art performance as an image classifier. At present, all the images are manually taken at the test location. The images will be acquired automatically by the final system.

2. Hardware

2.1. Radar System

We have used a radar system supplied by Robin Radar Systems B.V. (The Haag, Netherlands) because they provide an avian radar system that is able to detect birds. They also have tracker algorithms for tracking a detected object over time i.e., between the blips. The model we use is the ROBIN 3D FLEX v1.6.3 and it is actually a combination of two radars and a software package for implementation of various algorithms such as the tracking algorithms [7].

2.2. Video Head Control

We have used the PT-1020 motorized video head supplied by 2B Security Systems (Copenhagen, Denmark) [8]. The video head is operated by Pelco-D control protocol [9] and the control software for it is developed by us with C in Linux Ubuntu 16.04 platform. The video head steering is based on height, latitude and longitude coordinates (WGS84) provided by the radar. No coordinate conversion from one system to another is needed because all calculations are performed in WGS84 system. However, the geographical coordinates are converted to the rectangular coordinates in accordance with the Finnish Geodetic Institute [10].

2.3. Camera Control

We have manually collected the images at the test site with a Canon 7D mark II camera (Tokyo, Japan) and a Canon 500/f4 IS telephoto lens (Tokyo, Japan). The software for controlling the camera is developed with C# in Microsoft Visual Studio 14.0 because this is the only environment supported by the Canon API at present. The Canon API library of EDSDKLib-1.1.2 is applied. The code is developed in accordance with the instructions and functions of the API. The Canon API library is available for application on the Internet [11].

3. Data Processing

3.1. Input Data

Input data for the identification system consist of digital images and parameters from the radar. The parameters from the radar are real numbers such as velocity of a flying bird in m/s and bearing (i.e., a heading: the horizontal angle between the direction of an object and that of true north) in degrees. All images for training the CNN are of wild birds in flight and they have been taken manually at the test location. There are also constraints concerning the area where the images have to be taken. Here, the area refers to the air space in the vicinity of the pilot wind turbine. We have used the wind turbine swept area (the diameter of the swept area is 130 m) as a suitable altitude level constraint for taking the images, because birds flying below or above the swept area are not in danger. At this stage, the images are only taken in the vicinity of 1350 m in lengthwise direction, which is the distance to the pilot wind turbine. There are 1164 images for each class and the number of classes is 8, thus the original training set size is 8×1164 = 9312. We applied data augmentation as it is well-known method to increase performance of an image classifier. In addition, the original (i.e., not augmented) data set includes plenty of data examples of images with various portion of cloudiness as the background and also with clear sky as the background.

The number of images of each class should be the same as a CNN is applied [12] and therefore the lowest number of images of the classes is used. The number of classes (which includes both key species) is 8 at this phase. The eight classes for training the CNN are the Common Goldeneye (*Bucephala clangula*), the White-tailed Eagle (*Haliaeetus albicilla*), the Herring Gull (*Larus argentatus*), the Common Gull (*Larus canus*, the Lesser Black-backed Gull (*Larus fuscus fuscus*), the Black-headed Gull (*Larus ridibundus*, the Great Cormorant (*Phalacrocorax carbo*) and Common/Arctic Tern (*Sterna hirundo/paradisaea*).

3.2. Data Augmentation

Our system is operating in natural environment and therefore prevailing weather has significant influence on the tonality of the images taken at the test site. Obviously, the lighting will be different in a different time of a day and a different time of a year, and thus the toning of the images will be changing according to lighting. Color temperature is a property of a light source. It is the temperature of the ideal black-body radiator that radiates light of the same color as the corresponding light source. In this context black-body radiation is the thermal electromagnetic radiation emitted by a black body. A black-body is an opaque and non-reflective body. It has a specific spectrum and intensity that depends only on the temperature of the black-body, and it is assumed to be uniform and constant. In our case, the light source is the sun that closely approximates a black-body radiator. Even though the color of the sun may appear different depending on its position, the changing of color is mainly due to the scattering of light and it is not because of the changes in the black-body radiation [13–16]. Color matching functions (CMFs) provide the absolute energy values of three primary colors which appear the same as each spectrum color. We applied the International Commission on Illumination (*Commission internationale de l'éclairage*, CIE) 10-deg color matching functions in our data augmentation algorithm [17].

The data augmentation is done according to the curves in Figure 1. Ref. [18] by converting an image into different color temperatures between 2000 K° and 15,000 K° with step size s, where $s \epsilon \{50, 75, 100, 150, 200, 250, 300, 1000\}$. This makes the training set significantly larger, e.g., if s is 50, a class containing 1164 training examples becomes a class of 261 × 1164 = 303,804 examples + the original image. The augmented data set size as a result of various value of s is given in Table 1 for the original data set of size 8 × 1164 = 9312. After color conversion, the images are rotated by a random angle between $-20°$ and $20°$ drawn from the uniform distribution. This value has been altered from 30 to 20 since our first publication because it was empirically noticed that the target birds had never a position angled this steep. Motivation for image rotation is CNN's property of being invariant to small translations but not rotation of an image [19].

Table 1. Number of images for augmented data set with various step, s, values.

Step, s	Number of Images for One Class	Number of Images for 8 Classes
1100	15,132	121,056
700	23,280	186,240
350	45,396	363,168
200	77,988	623,904
100	153,648	1,229,184
50	304,968	2,439,744

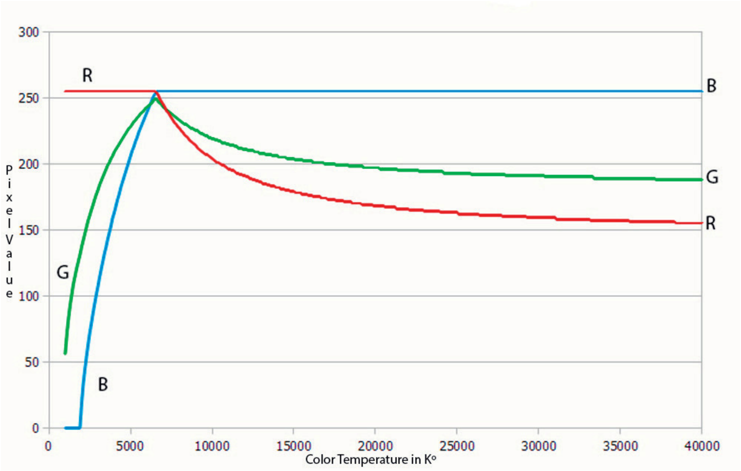


Figure 1. Color temperature and corresponding red, blue and green (RGB) values presented according to Commission Internationale de l'Eclairage (CIE) 1964 10-degree color matching function.

Examples of one original image and two images as an output of the augmentation algorithm with this original image as an input and s = 200 are presented in Figure 2. The color temperature of the original image is 7600 K° and the two augmented images 5600 K° and 9600 K°, respectively.

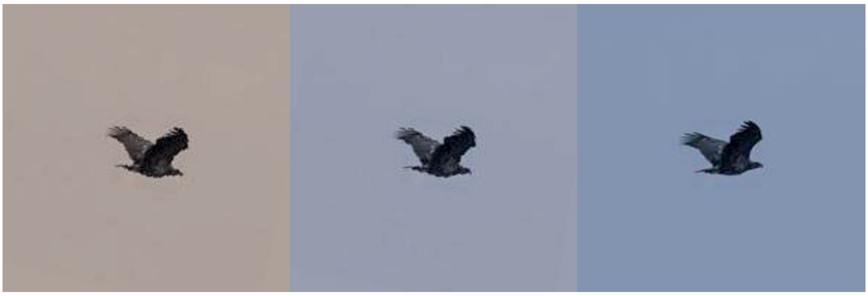


Figure 2. Data example of the White-tailed Eagle. The image on the left is an augmented image with the color temperature 5600 K°. The original image is in the middle with color temperature 7600 K°. The image on the right is an augmented image with the color temperature 9600 K°.

4. The Proposed System

The most important role of the radar is to detect flying birds, but it also provides parameters for bird identification (i.e., classification) [20,21]. The parameters provided by the radar system are: the distance in 3D of a target (m), the velocity of a target (m/s) and the trajectory of a target. The distance of a detected bird is used to estimate the size of the bird in meters. Velocity of a target bird is used for the final classification. The system also includes the aforementioned camera with the telephoto lens and a motorized video head. The camera is controlled by the application programmable interface (API) of the camera manufacturer. The system has three servers: the radar server, the video head steering server and the camera control server. Software for the radar server is supplied by the manufacturer of the radar but the software for the other two servers is result of our development work.

We took series of images of a single target bird and each image is processed according to the schematic diagram of the system in Figure 3.

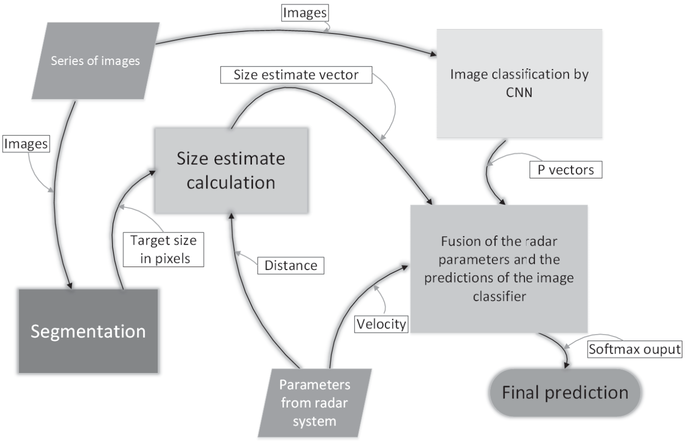
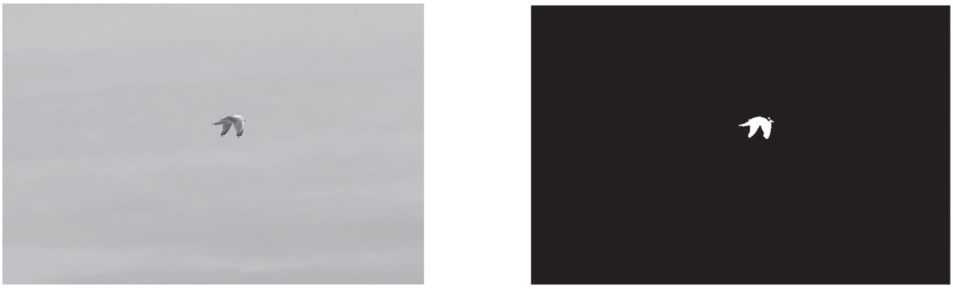

Images
Series of images
Size estimate vector
Image classification by CNN
Images
Size estimate calculation
P vectors
Target size in pixels
Distance
Fusion of the radar parameters and the predictions of the image classifier
Segmentation
Velocity
Softmax ouput
Parameters from radar system
Final prediction

Figure 3. Schematic diagram of the system.

Segmentation is computed in parallel to image classification in order to obtain an estimate of the target bird size in pixels, i.e., despite that segmentation is computed simultaneously when the classification process is started, it is not part of the actual classification, but the result of the segmentation is used for assigning a value to the size estimate parameter. When the estimate in pixels is known, the target bird size estimate in meters can be calculated. We studied methods from simple threshold to fuzzy logic for solving the problem at hand i.e., a dark figure against bright background and vice versa as well. At the extremity, the background and the target can share several colors in the RGB color space. We achieved the best results by applying fuzzy logic segmentation compared to the threshold segmentation and the edge detection segmentation [22,23]. In particular, we applied Mamdani's fuzzy inference method [24]. Figure 4a,b show an example of segmentation.



(**a**) (**b**)

Figure 4. Example of binary image acquired by the segmentation process. (**a**) an original image of the Herring Gull; (**b**) respective binary image as a result of segmentation of the original image.

5. Classification

The classification process is presented in Figure 5. Series of images of a single target (i.e., as a sequence of temporally consecutive frames of the same bird) are fed to the CNN that is applied to feature extraction. The two-step learning method is applied, i.e., the CNN is trained with the first N-1 layers viewed as feature maps and these maps are used to train a Support Vector Machine (SVM) classifier [25]. The SVM classifier makes use of one-versus-all binary learners, in which, for each binary learner, one class is positive and the rest are negative. The total number of the binary learners is the same as the number of classes. A linear classification model is applied. Stochastic gradient descent with 10 as the mini-batch size, and the Hinge loss function with regularization term $1/n$, where n is a number of training examples [26,27] are also applied. The output of the SVM is presented as ***P***-vectors as follows:

$$\boldsymbol{P}_i = [c_1, c_2, \ldots, c_{\text{nc}}], i = 1, \ldots, n, \tag{1}$$

where c_j is a probability of belonging to *class j*, *nc* is the number of classes and n is the number of images in each series, thus there will be one ***P***-vector for each image in any given image series.

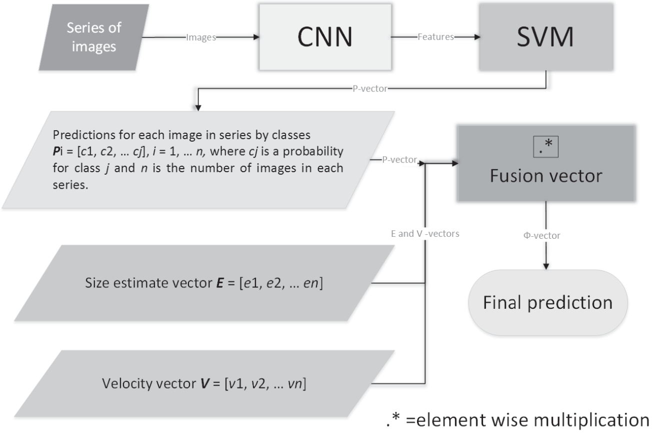

Series of images
Images
CNN
Features
SVM
P-vector
Predictions for each image in series by classes
Pi = [c1, c2, … cj], i = 1, … n, where cj is a probability for class j and n is the number of images in each series.
P-vector
.*
Fusion vector
E and V -vectors
Φ-vector
Size estimate vector **E** = [e1, e2, … en]
Final prediction
Velocity vector **V** = [v1, v2, … vn]
.* =element wise multiplication

Figure 5. The classification process.

There are also two parameters based on information provided by the radar system. The size of the target bird is estimated as follows. The frame size ([width x height y], in *pixels*) of the camera and the angle of view (α) of the lens are known. The distance (d) to the target bird is provided by the radar. The maximum number of horizontal (σ_h) and vertical (σ_v) *pixels* of the target bird are calculated from the segmented image, respectively. The angle of view, b, at the distance, d, is calculated over a right-angled triangle (see Figure 6). The horizontal number of *pixels/meter* is given by

$$\rho_h = \frac{x}{b_h}, \tag{2}$$

and the vertical number of *pixels/meter* by

$$\rho_v = \frac{y}{b_v}, \tag{3}$$

where, b_h and b_v denote the horizontal and the vertical angles of view, respectively. The estimate for the size of the bird in a single image in *square meters* as an area of rectangle is:

$$e = \frac{\sigma_h}{\rho_h} * \frac{\sigma_v}{\rho_v}. \tag{4}$$

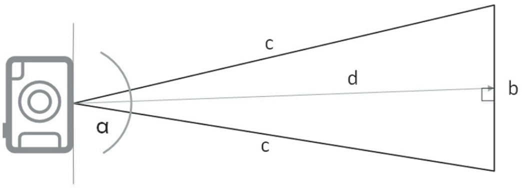


Figure 6. Diagram of the size estimate calculation.

The size estimate is presented as a vector with elements placed according to the class order (the classes are ordered alphabetically by their names), i.e., *class 1, class 2, . . . class nc*, where *nc* denotes the number of classes. The composition of the vector is following: calculate the average of the size estimates of the image series, check from the size-look-up table all the classes that contain the average size, *e*, turn those elements to one and set the others to zero, yielding

$$\textit{Size Estimate}, \boldsymbol{E} = [e_1, e_2, ..., e_{\text{nc}}], \tag{5}$$

with elements:

$$e_j = \begin{cases} 1, & \text{if } e \text{ fits class } j, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The velocity of the target bird is composed in similar way as the ***E***-vector in Size Estimate (5), i.e., check from the velocity-look-up table all the classes that contain the provided velocity, *v*, turn those elements to one and the others to zero.

$$\textit{Velocity}, \boldsymbol{V} = [v_1, v_2, ..., v_{\text{nc}}], \tag{7}$$

with elements:

$$v_j = \begin{cases} 1, & \text{if } v \text{ fits class } j, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The final classification is achieved by a fusion between the parameters provided by the radar and the predictions from the image classifier. The combined ***P***-vector for a series of images is:

$$\textit{Combined P -vector}, \boldsymbol{P} = \sum_{i=1}^{n} \boldsymbol{P}_i, \tag{9}$$

where *n* is the number of images in each series and the fusion vector, Φ, is:

$$\textit{Fusion vector}, \Phi = \boldsymbol{P}. * \boldsymbol{V}. * \boldsymbol{E}, \tag{10}$$

where ".∗" denotes element wise multiplication. The score, *S*, for final prediction is:

$$\textit{Prediction}, S = \mathbf{max}_j(\Phi), \tag{11}$$

$$j = \mathbf{arg\ max}_j(\Phi), \tag{12}$$

where *j* is the index of the predicted class.

5.1. Convolutional Neural Network

The CNN network architecture is presented in Figure 7. The architecture of the CNN results in (200 − 12 + 2 × 1)/2 + 1 = 96 for one side of the feature map and as of the result of square feature maps there are 96 × 96 = 9216 neurons in each feature map of the first convolution layer. Note that there is no max-pooling layer between the first and the second convolution layers. Motivation for this is that we wanted all of the finest edges to be included in resulting feature maps.

The input image is normalized and zero-centered before feeding it to the network. CNN with Mini-batch training and supervised mode as well as stochastic gradient descent with momentum is applied [28–31]. The L2 Regularization (i.e., weight decay) method for reducing over-fitting is also applied [30–32]. Due to limited capacity of computer resources the network size in terms of free parameters is kept small, thus resulting in total of 92 feature maps which are extracted by convolution layers with kernel sizes $[12 \times 12 \times 3] \times 12$, $[3 \times 3 \times 12] \times 16$ and $[3 \times 3 \times 16] \times 64$, respectively. Total number of weights is about 9.47×10^6.

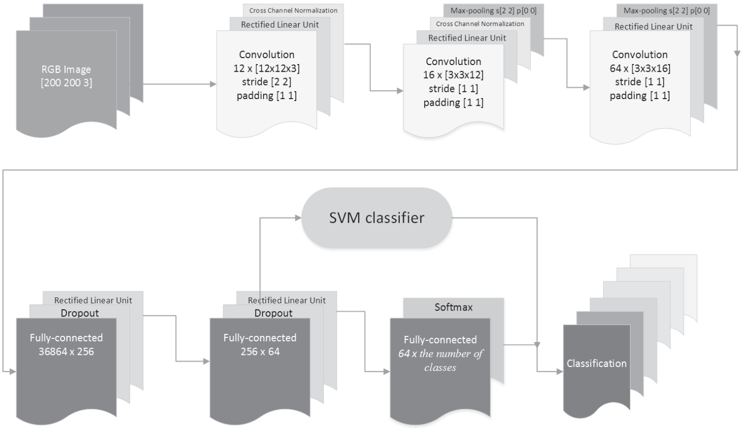


Figure 7. The architecture of the convolutional neural network. The letters, s, and, p, in the max-pooling layers denote stride and padding, respectively. In convolution layers, the first two numbers in the square brackets indicate the width and hight of the respective convolution kernel and the third number is the depth. The number before the brackets is the number of feature maps in respective convolution layer.

Each convolution layer is followed by a Rectified Linear Units (ReLU) nonlinearity layer [33], which simply applies a threshold operation,

$$f(x) = \begin{cases} 0 & x < 0, \\ x & x \geq 0, \end{cases} \tag{13}$$

to all the components of its input. This non-saturating nonlinearity in deep CNN makes the training several times faster when applied together with the hyperbolic tangent sigmoid transfer function [33,34]. Cross Channel Normalization layers follow the first and the second ReLU layers. These layers aid the generalization as their function may be seen as brightness normalization [34].

The purpose of max-pooling layer is to build robustness to small distortions. This is achievable by filtering over local neighbourhoods as follows: divide the input into rectangular pooling regions, and compute the maximum of each region, thus performing downsampling and reducing the overfitting as well [35].

There are three fully-connected layers at the end of the network for making final nonlinear combinations of features, and prediction by the last fully-connected layer followed by softmax activation which produces a distribution over the class labels with cross entropy loss function [31].

5.2. Hyperparameter Selection

The split into a training set and a validation set was 70% and 30%, respectively. The initial weights for all layers were drawn from the Gaussian distribution with mean 0 and standard deviation 0.01. Initial biases were set to zero. The L2 value was set to 0.0005 and mini-batch size was set to 128. The values of all the previously mentioned hyperparameters were fixed and we used manual tuning only for choosing the combination of the number of epochs and the learning rate drop period (LRDP). Two models with different values of the two parameters were trained on the original data set (i.e., no data augmentation applied). One model was trained on the augmented data set with s = 1100 and s = 350, respectively. Several models with various values of the two parameters were trained on the augmented data set with s = 200 and s = 50, respectively. The results of training these models are presented in Table 2, in which performance is presented as true positive rate (TPR, i.e., sensitivity). The initial values of the two parameters applied to training on each data set are selected empirically. As a result of running these tests, the best model in terms of performance is the model trained on the augmented data set with s = 50 (i.e., 2,439,744 training examples), the number of epochs = 8, and the LRDP = 3.

Table 2. Convolutional Neural Network (CNN) performance (with the Support Vector Machine (SVM) as an actual classifier) as a result of various number of epochs and Learning Rate Drop Period (LRDP).

Number of Training Examples	Number of Epochs	LRDP	TPR Training	TPR Generalization
9312	30	30	0.7175	0.6995
9312	60	60	0.7362	0.7052
121,056	25	10	0.8687	0.8662
363,168	18	7	0.9137	0.9187
623,904	12	12	0.9788	0.9253
623,904	16	16	0.9839	0.9254
623,904	24	24	0.9835	0.9170
623,904	16	5	0.9830	0.9270
623,904	16	6	0.9831	0.9337
623,904	16	9	0.9834	0.9249
623,904	16	13	0.9837	0.9154
2,439,744	3	3	0.9960	0.9246
2,439,744	5	5	0.9971	0.9313
2,439,744	8	8	0.9984	0.9363
2,439,744	12	12	0.9984	0.9296
2,439,744	5	3	0.9965	0.9250
2,439,744	8	3	0.9983	0.9463
2,439,744	10	3	0.9984	0.9448
2,439,744	12	3	0.9983	0.9425

Initial learning rate was set to 0.01 and when the same value was applied to the number of epochs and the LRDP the learning rate was kept constantly at its initial value. The learning rate decay schedule (LRDS) was applied when the values of the number of epochs and the LRDP were different of each other. In the LRDS method, the learning rate is dropped by a factor of 0.1 (i.e., the updated learning rate will be the current learning rate × 0.1) when a given number of epochs is reached. This given number of epochs is the effective value of the LRDP. Motivation for using the LRDS method is as training proceeds with shorter leaps on the loss function surface from some point on, the optimal value for the weights (i.e., in terms of performance as a classifier) can be found more accurately. If only the short leaps would be applied, the number of epochs should be very large, thus resulting in significant increase of training time. The challenge is to find the points from where on the learning rate should be reduced. We approached this problem in two ways. We fixed the LRDP value and altered the number of epochs. Initially, the problem was to find a suitable starting value for the LRDP. It was intuitively clear that the LRDP value should increase as the number of epochs increases. A small value of the LRDP combined with a high value of the number of epochs would lead to substantial underfitting. We also fixed the number of epochs and altered the LRDP value instead. The same initial value problem concerns this approach as well. However, the size of the respective data set should give some guidance for choosing the initial values. Moreover, as the number of training examples increases, the number of epochs should decrease in order to avoid overfitting.

We applied the dropout technique for improving the performance of our CNN [34,36]. We trained models with fixed hyperparameter values with and without the dropout technique. If overfitting occurs, the results in terms of classification performance should be better as the dropout technique is applied compared to those models for which it is not applied. These tests indicate that some overfitting occurs when the models were trained on the augmented data sets but not necessarily on the original data sets. The dropout was implemented after the first and the second fully-connected layers by randomly setting the output neurons to zero with a probability of 0.5.

6. Results

The following results are based on manually taken images at the test site. The images have been taken at the same position where the camera will be installed. We trained two models on the original data set and several models on four different augmented data sets, in which *s* was 50, 200, 350 and 1100, respectively (see Table 2). The models with *s* ϵ {350,1100} were trained only for testing the data augmentation algorithm. The effect of the data augmentation algorithm on classification performance is presented in Figure 8.

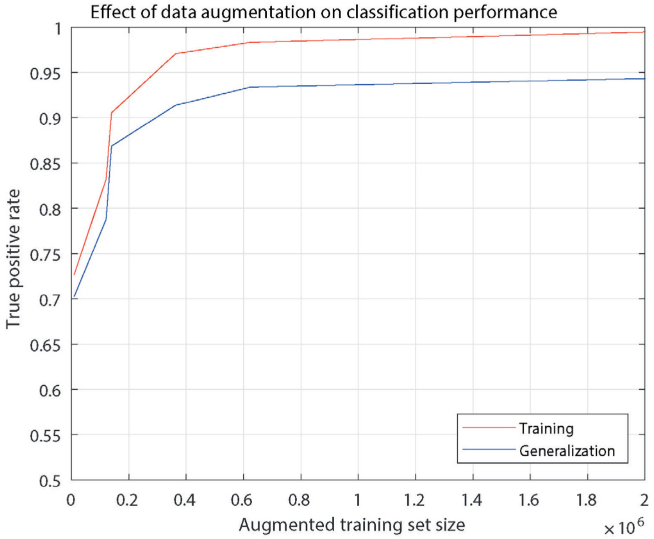
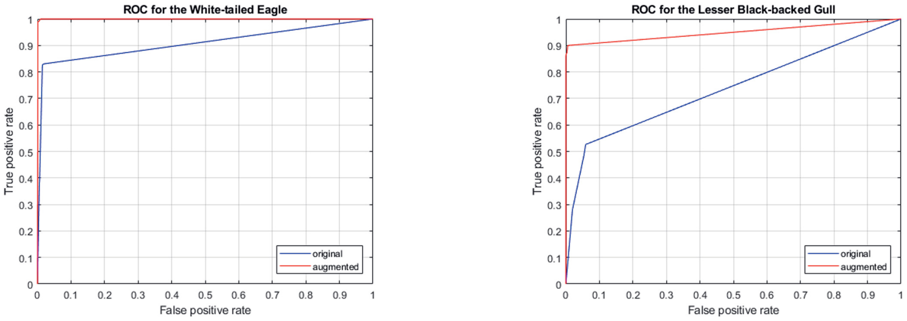


Figure 8. The red curve is for validation during training and the blue curve is according to the generalization test. The actual True Positive Rate (TPR) values are used with *s* ϵ {350,1100}, and the average TPR value is used of the models with *s* ϵ {50,200}, respectively. The starting value for both curves is the average value of the two models trained on the original data set.

The best performance (in TPR) of the two models trained on the original data set is 0.7362. Performance for the models trained on the augmented data sets varies between 0.8687 and 0.9984, which shows clear improvement as the augmented training set size increases and especially compared to the models trained on the original set. Training with and without the dropout technique implied that overfitting will occur to some extent as the data augmentation is applied and the dropout technique decreases this overfitting. The results were different for the original data sets, in which case overfitting was insignificant. These results are logical due to the fact that the enhancement in performance obtained by the data augmentation is extracted from the original images, and thus it inevitably increases redundancy. The results for the original data sets imply that the number of training examples was simply not large enough.

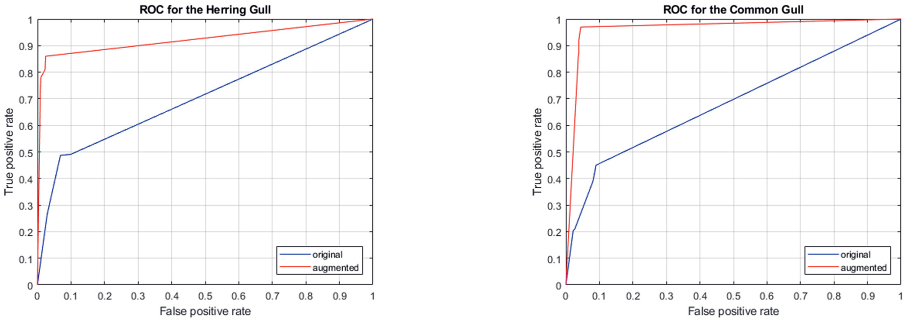
We tested generalization of the models on 100 unseen images for each class, i.e., the data set for testing the models was 8·100,100 = 600 images that the models have never seen before. According to these tests the system achieves its state-of-the-art performance of 0.9463 with the augmented data set of the size 2.44 × 10^6 (i.e., the color conversion step size, s = 50), number of epochs 8, LRDP 3, and the dropout applied.

The receiver operating characteristic (ROC) curves and the area under the curve (AUC) for the 8 classes (i.e., bird species) are presented in Figures 9–12. The TPR values of the generalization tests are applied in these figures. The red curve is for the augmented data set and the blue curve is for the original data set.



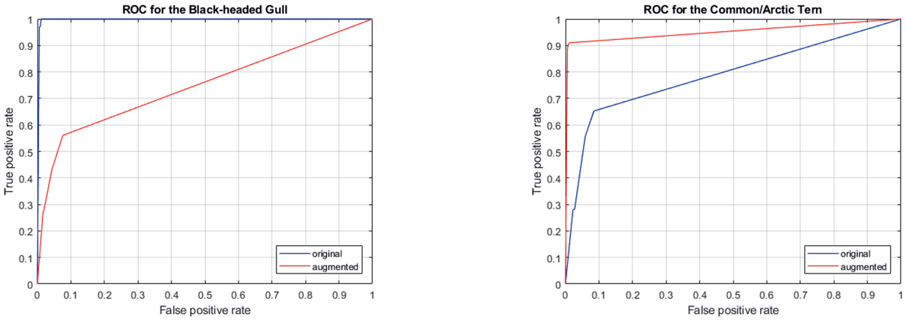
(**a**) (**b**)

Figure 9. ROC curves for the White-tailed Eagle and the Lesser Black-backed Gull. (**a**) AUC for the original data set and for the augmented data set is 0.9137 and 0.9993, respectively; (**b**) AUC for the original data set and for the augmented data set is 0.7460 and 0.9496, respectively.



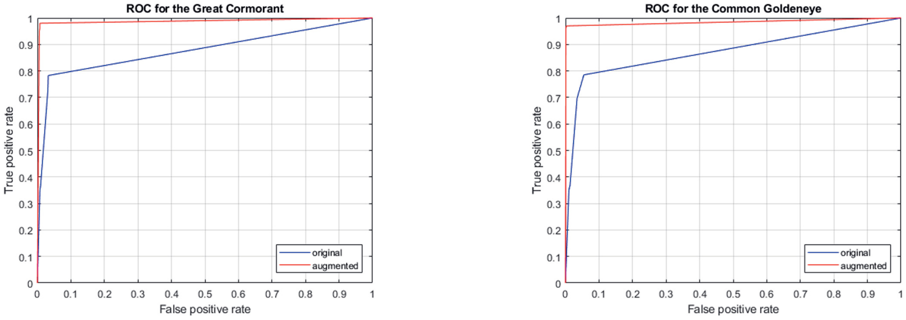
(**a**) (**b**)

Figure 10. ROC curves for the Herring Gull and the Common Gull. (**a**) AUC for the original data set and for the augmented data set is 0.6926 and 0.9128, respectively; (**b**) AUC for the original data set and for the augmented data set is 0.6967 and 0.9644, respectively.



(**a**) (**b**)

Figure 11. ROC curves for the Black-headed Gull and the Common/Artic Tern. (**a**) AUC for the original data set and for the augmented data set is 0.7583 and 0.9972, respectively; (**b**) AUC for the original data set and for the augmented data set is 0.8111 and 0.9508, respectively.



(**a**) (**b**)

Figure 12. ROC curves for the Great Cormorant and the Common Goldeneye. (**a**) AUC for the original data set and for the augmented data set is 0.8853 and 0.9870, respectively. (**b**) AUC for the original data set and for the augmented data set is 0.0.8807 and 0.9829, respectively.

7. Discussion

We assembled the non-deep (i.e., in terms of the number of the convolution layers, 3) CNN for image classification, and demonstrated that the model is suitable for real-world application, especially, when the number of training data is limited. We presented and demonstrated that our data augmentation method improves significantly the performance of the classifier, and the desirable state-of-the-art performance as an image classifier can be achieved by applying it. Thus, we showed that the data augmentation is crucial for the classification performance. We also showed that our model generalizes well to images never seen before and hence it is applicable for real-world problem. The number of images in the original data set have been increased since our first publication resulting in the better state-of-the-art performance of 0.9463 compared to the first result of 0.9100. It is noteworthy that this better result is achieved despite of the increased number of the classes, i.e., 8 compared to 6 [37].

The measured performance of the image classifier has been obtained without using the parameters supplied by the radar. It is obvious that those parameters (i.e., the E- and V-vectors) provide additional and relevant a-priori knowledge to the system and they can turn a misclassified (by images) class into the correct one. Data collection will be continued at the test site resulting in a larger original data set, and thus hopefully better performance of the classifier. The number of classes will increase as more images of scarcer species are collected.

We are currently working on the collision detection problem, but no collisions have been observed until now while the pilot wind turbine has been manually monitored for 30 months. It seems that collisions are quite rare in the research area and this makes the field testing of the possible collision detection methods challenging. More research is required of possible deterrent methods, especially on species or species group level.

We proposed a novel system for automatic bird identification as a real world application. However, the system has restrictions such as images can not be taken in pitch-dark or in poor visibility conditions. Infrared cameras may contribute to the collision detection, but their contribution to classification is poor because all color information is lost. The proposed system is still in the installation phase, so we have not yet been able to test the complete system.

Author Contributions: Conceptualization, J.N.; Data Curation, J.N.; Formal Analysis, J.N.; Funding Acquisition, J.T.T.; Investigation, J.N.; Methodology, J.N.; Project Administration, J.N.; Software, J.N.; Supervision, J.T.T.; Validation, J.T.T.; Visualization, J.N.; Writing—Original Draft, J.N.; Writing—Review and Editing, J.T.T.

Funding: This research received no external funding.

Acknowledgments: The authors wish to thank Suomen Hyötytuuli for the financial support of purchasing necessary equipment and Robin Radar Systems for the technical support with the applied radar system.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application programmable interface
AUC	Area under the curve
CIE	Commission internationale de l'éclairage
CMFs	Color matching functions
CNN	Convolutional neural network
DSLR	digital single-lens reflex camera
LDPR	Learning rate drop period
LRDS	Learning rate decay schedule
ReLU	Rectified linear units
ROC	Receiver operating characteristic
SVM	Support vector machine
TPR	True positive range

References

1. Desholm, M.; Kahlert, J. Avian Collision Risk at an Offshore Wind Farm. *Biol. Lett.* **2008**, *1*, 296–298. [CrossRef] [PubMed]
2. Marques, A.T.; Rodrigues, S.; Costa, H.; Pereira, M.J.R.; Fonseca, C.; Mascarenhas, M.; Bernardino, J. Understanding bird collisions at wind farms: An updated review on the causes and possible mitigation strategies. *Biol. Conserv.* **2014**, *179*, 40–52. [CrossRef]
3. Baxter, A.T.; Robinson, A.P. A comparison of scavenging bird deterrence techniques at UK landfill sites. *Int. J. Pest Manag.* **2007**, *53*, 347–356. [CrossRef]
4. Verhoef, J.P.; Westra, C.A.; Korterink, H.; Curvers, A. WT-Bird A Novel Bird Impact Detection System. Available online: www.ecn.nl/docs/library/report/2002/rx02055.pdf (accessed on 27 September 2018).
5. Wiggelinkhuizen, E.J.; Barhorst, S.A.M.; Rademakers L.W.M.M.; den Boon, H.J. Bird Collision Monitoring System for Multi-Megawatt Wind Turbines, WT-Bird: Prototype Development and Testing. Available online: www.ecn.nl/publications/PdfFetch.aspx?nr=ECN-E--06-027 (accessed on 27 September 2018).
6. Wiggelinkhuizen, E.J.; den Boon, H.J. Monitoring of Bird Collisions in Wind Farm under Offshore-like Conditions Using WT-BIRD System: Final Report. Available online: www.ecn.nl/docs/library/report/2009/e09033.pdf (accessed on 27 September 2018).
7. Robin Radar Models. Available online: https://www.robinradar.com/ (accessed on 27 September 2018).

8. PT1020 Video Head. Available online: http://www.2bsecurity.com/product/pt-1020-medium-sized-pan-tilt/ (accessed on 27 September 2018).
9. Bruxy REGNET for Pelco-D Protocol. Available online: http://bruxy.regnet.cz/programming/rs485/pelco-d.pdf (accessed on 27 September 2018).
10. Häkli, P.; Puupponen, J.; Koivula, H. Suomen Geodeettiset Koordinaatistot Ja Niiden VäLiset Muunnokset. *Natl. Land Surv. Finl.* **2009**. Available online: https://www.maanmittauslaitos.fi/sites/maanmittauslaitos.fi/files/fgi/GLtiedote30_korjausliite.pdf (accessed on 27 September 2018).
11. Canon's European Developer Programmes. Available online: https://www.developers.canon-europa.com/developer/bsdp/bsdp_pub.nsf (accessed on 27 September 2018).
12. Hensman, P.; Masko, D. The Impact of Imbalanced Training Data for Convolutional Neural Networks. Available online: https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf (accessed on 27 September 2018).
13. Speranskaya, N.I. Determination of spectrum color co-ordinates for twenty-seven normal observers. *Opt. Spectrosc.* **1959**, *7*, 424–428.
14. Stiles, W.S.; Burch, J.M. NPL colour-matching investigation: Final report. *Opt. Acta* **1959**, *6*, 1–26. [CrossRef]
15. Wyszecki, G.; Stiles, W.S. *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed.; John Wiley & Sons Inc.: New York, NY, USA, 1982; ISBN 978-0471021063.
16. Stockman, A.; Sharpe, L.T. Spectral sensitivities of the middle- and long-wavelength sensitive cones derived from measurements in observers of known genotype. *Vis. Res.* **2000**, *40*, 1711–1737. [CrossRef]
17. CIE. *CIE Proceedings, Vienna Session*; Committee Report E-1.4.1; CIE: Paris, France, 1963; pp. 209–220.
18. Blackbody Color Datafile. Available online: www.vendian.org/mncharity/dir3/blackbody/UnstableURLs/bbr_color.html (accessed on 27 September 2018).
19. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: www.deeplearningbook.org (accessed on 27 September 2018).
20. Richards, M.A. *Fundamentals of Radar Signal Processing*; The McGraw-Hill Companies: New York, NY, USA, 2005; ISBN 0-07-144474-2
21. Bruderer, B. The Study of Bird Migration by Radar, part1: The Technical Basis. *Naturwissenschaften* **1997**, *84*, 1–8. [CrossRef]
22. The MathWorks, Inc. Fuzzy Logic Toolbox Documentation. Available online: https://se.mathworks.com/help/fuzzy/fuzzy.pdf. (accessed on 27 September 2018).
23. Yuheng, S.; Hao, J. Image Segmentation Algorithms Overview. Available online: https://arxiv.org/ftp/arxiv/papers/1707/1707.02051.pdf (accessed on 27 September 2018).
24. Mamdani, E.H.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Mach. Stud.* **1975**, *7*, 1–13. [CrossRef]
25. Huang, J.F.; LeCun, Y. Large-Scale Learning with Svm and Convolutional Nets for Generic Object Categorization. Available online: http://yann.lecun.com/exdb/publis/pdf/huang-lecun-06.pdf (accessed on 27 September 2018).
26. Moore, R.C.; DeNero, J. L1 and L2 regularization for multiclass hinge loss models. In Proceedings of the Symposium on Machine Learning in Speech and Language Processing, Bellevue, WA, USA, 27 June 2011.
27. Duan, K.B.; Keerthi, S.S. Which Is the Best Multiclass SVM Method? An Empirical Study. *Mult. Classif. Syst. LNCS* **2005**, *3541*, 278–285.
28. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
29. Li, M.; Zhang, T.; Chen, Y.; Smola, A.J. Efficient Mini-batch Training for Stochastic Optimization. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge, New York, NY, USA, 24–27 August 2014; pp. 661–670, ISBN 978-1-4503-2956-9.
30. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012; ISBN 978-0-262-01802-9.
31. Bishop, C.M. *Pattern Recognition and Machine Learning*; Jordan, M., Kleinberg, J., Schölkopf, B., Eds.; Springer: New York, NY, USA, 2006; ISBN 0-387-31073-8.
32. Haykin, S. *Neural Networks: A Comprehensive Foundation*, 2nd ed.; Prentice Hall/Pearson: New York, NY, USA, 1994; p. 470, ISBN 0-13-908385-5.

33. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*. [CrossRef]
35. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.A.; LeCun, Y. What is the best multi-stage architecture for object recognition. In Proceedings of the International Conference on Computer Vision, Kyoto, Japan, 29 Septemer–2 October 2009; pp. 2146–2153.
36. Srivastave, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
37. Niemi, J.; Tanttu, J.T. Automatic Bird Identification for Offshore Wind Farms: A Case Study for Deep Learning. In Proceedings of the 59th IEEE International Symposium ELMAR-2017, Zadar, Croatia, 18–20 September 2017. [CrossRef]

PUBLICATION

IV

Deep Learning Based Automatic Bird Identification System for Offshore Wind Farms

J. Niemi and J. T. Tanttu

Wind Energy 23.6 (2020)
DOI: 10.1002/we.2492

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

JOURNAL ARTICLE

Deep Learning Based Automatic Bird Identification System for Offshore Wind Farms

Juha Niemi* | **Juha T. Tanttu**

[1]Faculty of Information and Communication Technology, Tampere University, Finland

Correspondence
*Juha Niemi. Email: juha.k.niemi@tuni.fi

Present Address
Tampere University

Summary

Practical deterrent methods are needed to prevent collisions between birds and wind turbine blades for offshore wind farms. It is improbable that a single deterrent method would work for all bird species in a given area. An automatic bird identification system is required in order to develop bird species-level deterrent methods. This system is the first and necessary part of the entirety that is eventually able to automatically monitor bird movements, identify bird species, and launch deterrent measures. A prototype system has been built on Finnish west coast.

In the proposed system, a separate radar system detects birds and provides WGS84 coordinates to a steering system of a camera. The steering system consists of a motorized video head and our software to control it. The steering system tracks flying birds in order to capture series of images by a digital single-lens reflex camera. Classification is based on these images, and it is implemented by convolutional neural network trained with a deep learning algorithm. We applied to the images our data augmentation method, in which images are rotated and converted into different color temperatures. The results indicate that the proposed system has good performance to identify bird species in the test area. Aiming accuracy for the video head was 88.91 %. Image classification performance as true positive rate was 0.8688.

KEYWORDS:
machine learning, deep learning, convolutional neural networks, image classification, intelligent surveillance systems, wind farms

1 | INTRODUCTION

The first Finnish offshore wind farm has been constructed on the west coast. The authorities are concerned about the possible bird mortality caused by the constructed wind turbines of the height of 155 m. This has resulted in explicit statements in the environmental license, which obligate the operator of the wind farm to monitor bird movements in the area, and to mitigate, or prevent if possible, collisions between birds and the wind turbines. The authorities have announced two key bird species for particular monitoring in the area: the white-tailed eagle (*Haliaeetus albicilla*) and the lesser black-backed gull (*Larus fuscatus fuscatus*). This demand requires an automatic bird identification system to be developed prior to any measures can be launched to monitor, and to deter, the birds in the area. The alternative would be manual observation by humans which is expensive and inaccurate. Currently, birds are monitored by a radar system, which shuts down wind turbines when any bird flies into a perimeter of 300 m to a wind turbine. This is the minimum distance in terms to have sufficient time to shut down the turbine. However, the number of fast restarts of the turbines should stay as low as possible due to wearing of the mechanics, and therefore this operation should be used as a last resort. A solution to this is a suitable deterrent method, but it is plausible that several different methods are required, because of variety of bird species in the area. An extra problem is that the breeding birds may quickly become accustomed to, e.g., sounds as a deterrent method[1]. The long term objective is to develop a deterrent system that operates either on species-level or species-group level. This means that a different deterrent method is applied to e.g., eagles and waterfowl.

Identification of bird species is mainly based on morphology and vocalization, of which vocalization is not a feasible method in the offshore environment because of long ranges and background noise. Images taken of birds in the test area (the wind farm) are used as a feasible method to study morphology. This turns the identification problem to an image classification problem. Thus, the problem of automatic bird identification in real-time is two-fold: how to successfully aim the camera to a target bird in order to collect images and how to classify (identify) the images?

We propose a system that consists of a separate radar system, a digital single-lens reflex camera with a telephoto lens, and a motorized video head. The radar system is a solution to detect birds in the sky. The applied radar system passes WGS84 coordinates of the detected object to steering software of the video head. Successful aiming requires that the horizontal and the vertical angular resolution (thought as a rectangle) of the radar system is equal or smaller than a rectangle formed by the focusing points of the camera. Focusing is based merely on the autofocus system of the camera.

A convolutional neural network (CNN) trained with a deep learning algorithm is widely used solution to the image classification problem[2,3]. CNNs typically need lot of training data. In real-world applications, it is usually difficult and time-consuming to collect large number of images for each class. Data augmentation is a solution to this problem. We have developed our own data augmentation method, in which the images are first converted into various color temperatures and then rotated by a randomly chosen angle.

The best results have been achieved by a CNN model with three convolution layers, and applying our data augmentation to the training data. Image classification performance of the model was 0.8688 as true positive rate (TPR). TPR is given as the number of true positives divided by the number of all positives in the test data. Aiming accuracy for the video head was 88.91 %. These results imply that the system is capable to take images of almost 90 % of the detected (by the radar system) birds and identify 97.72 % of birds in these images. However, high winds decreases the aiming accuracy and poor visibility disturbs the image collection, resulting in a lower classification performance in these circumstances.

2 | RELATED WORK

A research group at the Univ. of Toledo, Ohio, has developed a prototype system integrating radar, infrared, and acoustic information[4]. This system was able to identify a limited set of bird and bat species mainly based on their vocalizations and also estimate the flight trajectories in 3D by fusing the infrared and radar data.

Wiggelinkhuizen et al. have developed a method (WT-bird) for detection and registration of bird collisions that is suitable for continuous remote operation onshore. The characteristic sound of a collision is detected by sensors in the blades, which triggers the video registration and sends an alert message to the operator. This implementation is based on monitoring noise, generated from an impact of bird collision with a wind turbine. The collision is detected with microphones, and the noise monitoring is combined with a video camera. The original objective of this project was to develope a calibrated bird collision monitoring system for offshore, which was not generally achieved mainly due to technical problems[5,6,7].

Rosa et al. have studied machine learning (ML) algorithms implemented in marine radars in order to automatically detect and attempt to classify objects. Six ML algorithms have been applied and their performance have been compared. These widely used ML algorithms are: random forests (RF), support vector machine (SVM), artificial neural networks, linear discriminant analysis, quadratic discriminant analysis, and decision trees (DT). All algorithms showed good performance when the problem was to distinguish birds from non-biological objects, but the algorithms showed poor performance when the problem was to classify within bird species of bird species groups (e.g., herons vs. gulls)[8].

We are also aware of one commercial system: the DTBird developed by Liquen Consultora Ambiental,S.L., Spain[9]. This system is based on video-recording bird flights near wind turbines, and it promises to detect birds automatically and prevent possible collisions in the vicinity of the turbines. However, Roel May et al. have evaluated how well the DTBird system is able to detect birds in a wind farm in Norway. They also examined the suitability of DTBird to study near-turbine bird flight behaviour and possible deterrence[10]. Their evaluation showed the following results: detectability was over 80 %, the daily number of false positives was below two, the percentage of falsely triggered warnings/dissuasions was circa 50 %, and the percentage of falsely triggered warnings and dissuasions was 40 %. Thus, the DTBird system met two of their four evaluation criteria. In addition, the researchers found that the DTBird system enables monitoring of near-turbine flight behaviour, although individual birds usually cannot be identified to the species level[10].

Birdsnap by Thomas Berg et al. proposes a solution to the problem of large-scale fine-grained visual categorization, resulting in an on line field guide to 500 North American bird species[11]. Users can upload bird images into the field guide database, and the developed system identifies the images automatically. Researchers introduce one-vs-most classifiers by eliminating highly similar species during training, and they show how spatio-temporal class priors can be used to improve performance. The spatio-temporal class priors are gained from the embedded time and location data that modern cameras include in each image file they produce. Birdsnap uses a set of one-vs-most linear SVMs based on POOFs[12], and it achieved an accuracy of 0.8240 in bird species identification[11].

Yoshihashi et al. have applied time-lapse images to detect birds around a wind farm[13]. Time-lapse photography is a technique in which the frame rate of viewing a sequence of images is different than the frame rate of taking the sequence of images. Time-lapse images can make very fast or very slow time-related processes better interpretable to the human eye. The system developed by Yoshihashi et al. utilizes a fixed camera and the following algorithms: AdaBoost (Adaptive Boosting), Haar-like feature extraction, and histogram of oriented gradients (HOG). A CNN architecture was also applied to the image classification problem. AdaBoost is a learning algorithm for binary classification, which is developed to improve classification performance by combining multiple weak classifiers into a single strong classifier. The idea of AdaBoost is to give more weight to the data points that are poorly classified by the weak learners[14]. Haar-like features are digital image features used in object recognition. In mathematics, the name Haar refers to square-shaped functions which together form a wavelet family. Haar-like is an image feature that utilizes contrasts in images[15]. HOG is a feature descriptor used to detect objects in computer vision. A feature descriptor is a representation of an image that simplifies the image by extracting useful information and discarding irrelevant information. A feature descriptor introduces a 2D image as a feature vector. The main idea of HOG is that local object appearance and shape within an image can be described by the distribution of intensity gradients[16]. Yoshihashi et al. found that the best method for detection was Haar-like, and the best method for classification was CNN. The system was tested on two bird species groups: hawks and crows, and it achieved only moderate performance[13].

3 | THE SYSTEM

The proposed system consists of several hardware as well as software modules. See Fig. 1 for an illustration. We used a radar system supplied by Robin Radar Systems B.V. because they provide an avian radar system that is able to detect birds. They also have tracker algorithms for tracking a detected object over time (between the blips). The model we used is the ROBIN 3D FLEX v1.6.3, and it is actually a combination of two radars

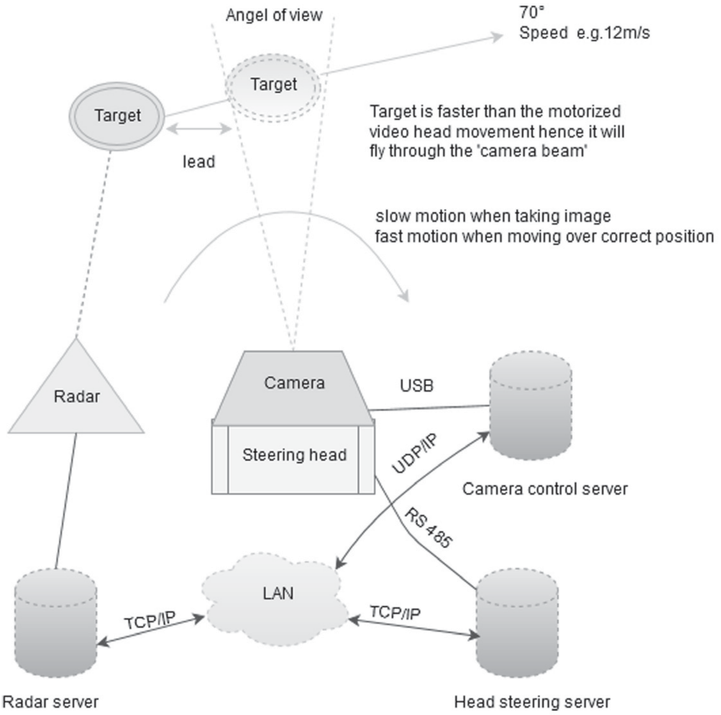

Angel of view
70°
Speed e.g.12m/s
Target
Target
lead
Target is faster than the motorized video head movement hence it will fly through the 'camera beam'
slow motion when taking image
fast motion when moving over correct position
Radar
Camera
Steering head
USB
UDP/IP
RS 485
Camera control server
LAN
TCP/IP
TCP/IP
Radar server
Head steering server

FIGURE 1 The hardware of the system and the principle of catching flying bird into the frame area of the camera.

and a software package for implementation of various algorithms such as the tracker algorithms[17]. The radar system was connected to a local area network (LAN), and thus it was able to communicate with the servers. The most important role of the radar system is to detect flying birds, but it also provides parameters. Among these parameters, we found the flight velocity (m/s) of a target bird to be the most usable for bird species identification. The developed system uses a PT-1020 Medium Duty motorized video head supplied by the 2B Security Systems. This video head has separate motors for horizontal and vertical steering, and it is operated by Pelco-D control protocol[18]. We used the Canon EOS 7D II camera with 20.2-megapixel sensor and the Canon EF 500/f4 IS lens. The camera is controlled by the application programmable interface (API) of the camera manufacturer. Initially, correct focusing was based on the autofocus system of the lens and the camera. We also used the automatic exposure system of the camera.

The system has three servers: the radar server, the video head steering server, and the camera control server. Software for the radar server is supplied by the manufacturer of the system, but software for the other two servers, for controlling the camera, and for steering the video head are developed by us. For more detailed description of the system, see[19].

4 | DATA

Input data of this application consist of digital images. All images for training the CNN have been taken manually at the test location in various weather conditions. The location is the same where the camera has been installed for taking images automatically. The collected image set was divided into two datasets: an original dataset for training the classifier, and a test dataset for measuring generalization ability of the classifier, and thus the classifier has not seen these test images during training. Both datasets are divided into 11 classes. The original dataset contains 23552 images, and test dataset consists of 425 images. The test dataset was created by randomly choosing images from all classes. The number of images in the test dataset follows the distribution of bird species in the images of the original dataset. Class labels and number of images of the original dataset and the test dataset for each class are presented in Table 1. In this Table, two classes are not defined at species level: LNSP and CATE. The first case is because there is no need to distinguish between loon species any further in this context, regardless the fact that two common, and two rare species of loons occur in the test area. The same applies to the second case as well, namely the common/arctic tern. In addition, it is generally very difficult to tell the difference between these two tern species[20], and thus the number of required data examples (images) would increase very large.

TABLE 1 The original data set and the test data set divided into 11 classes.

# Images	# Test Images	Class Name (Eng.)	Class Name (Lat.)	Class Label
396	7	Loon species	Gavia sp	LNSP
5458	100	Great Cormorant	Phalacrocorax carbo	GRCO
979	17	Common Eider	Somateria mollissima	COEI
1164	21	Common Goldeneye	Bucephala clangula	COGO
2436	44	White-tailed Eagle	Haliaeetus albicilla	WTEA
512	9	Great Black-backed Gull	Larus marinus	GBBG
3968	72	Herring Gull	Larus argentatus	HEGU
1481	26	Lesser Black-backed Gull	Larus fuscatus	LBBG
3581	65	Common Gull	Larus canus	COGU
1803	32	Black-headed Gull	Chroicocephalus ridibundus	BHGU
1774	32	Common/Arctic Tern	Sterna hirundo/paradisaea	CATE

4.1 | Data Augmentation

Data augmentation is applied to the original dataset. We have developed our own method, in which the images are converted into various color temperatures according to step size, *s*. The lower and upper limit to the color temperature is 2000 K and 15000 K, respectively. For example, if *s* = 50 (in K), the number of data examples of the augmented dataset is (15000 - 2000)/50 + 1 * 23552 = 6147072. In addition to the color conversion, the images are also rotated by a random angle between -20 degree and 20 degree drawn from the uniform distribution. Motivation for this is that CNN is invariant to small translations, but not image rotation[21] . For more details, see[19,22] . Number of images for each species (classes) in the augmented dataset when s = 50 and s = 200, respectively, are presented in Table 2 .

TABLE 2 Number of images for each class in the augmented data set.

Class Label	Original	s = 50	s = 200
LOSP	396	103752	26532
GRCO	5458	1424538	360228
COEI	979	256498	65593
COGO	1164	304968	77988
WTEA	2436	635796	160776
GBBG	512	134144	34304
HEGU	3968	1035648	261888
LBBG	1481	388022	99227
COGU	3581	934641	236346
BHGU	1803	472386	120801
CATE	1774	464788	118858

5 | IMAGE SEGMENTATION

We applied a fuzzy logic method to segmentation in our earlier models, but this method showed to be computationally expensive. For more details, see[22] . In this paper, we have used merely convolution without a neural network for segmentation[23] . A kernel of a size 8 × 8 is used to convolve images. We have applied constant to kernel parameters, instead of learning these parameters by CNN. The best results for segmentation was achieved by the kernel that has contour (border) parameters set to 1, and all the other parameters were set to -1, see Fig. 2 for illustration of this kernel. Padding was set to 1 and stride was set to 2. The result of the convolution is a feature map. The size of the feature map is given by:

$$F = (I - K + 2P)/S + 1, \tag{1}$$

1	1	1	1	1	1	1	1
1	-1	-1	-1	-1	-1	-1	1
1	-1	-1	-1	-1	-1	-1	1
1	-1	-1	-1	-1	-1	-1	1
1	-1	-1	-1	-1	-1	-1	1
1	-1	-1	-1	-1	-1	-1	1
1	-1	-1	-1	-1	-1	-1	1
1	1	1	1	1	1	1	1

FIGURE 2 Kernel of a size 8 × 8 with constant parameters for convolution.

where F is the size of the feature map, I is the size of an input image, K is the size of the kernel, P is padding, and S is stride. The sizes of F, I, and K are expressed as a vector [m n], where m is the number of rows and n is the number of columns. The input image is resized to 1600 × 2400 pixels before convolution, and thus the feature map size is 798 × 1198 pixels. The feature map is regarded as a grayscale image, which is converted to a binary image by applying a simple threshold. Finally, the binary image is used as a mask when the original image is cropped. The mask is used so that the uttermost white pixels of it define the four corners of the cropped image. In addition, an offset is added to all of the four corners in order to make sure that the cropped image has enough pixels for classification, and also to confirm that the cropped image has an aspect ratio of 1:1 (square). When the cropped image has the correct aspect ratio, it is resized to 200 × 200 pixels, which is the image size used in the classifier. Figure 3 shows an original image of the herring gull (3a), the feature map of the herring gull image (3b), this feature map converted to a binary image (3c) and finally, an image that is cropped from the original herring gull image by using the binary image as a mask (3d).

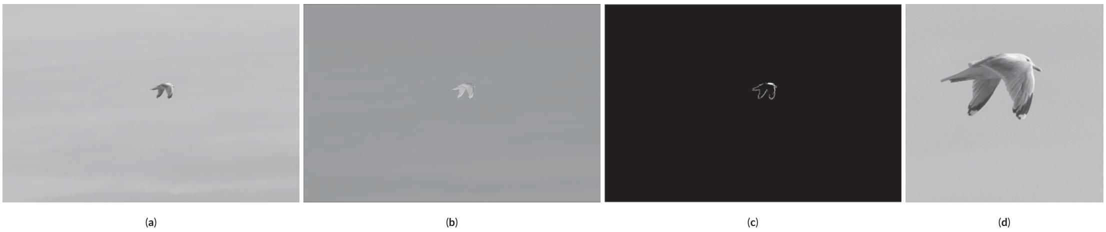


FIGURE 3 An original, uncropped image of the herring gull, the feature map of it, this feature map converted to a binary image, and the resulting cropped image.

6 | AUTOMATIC IMAGE COLLECTION

The ability of aiming the lens of the camera at a target bird by the video head, and thus capturing images automatically, is based on the angular resolutions of the two radars. Radar accuracy is measured as a range resolution and an angular resolution. The range resolution describes how long distance is needed in lengthwise between two objects, in order them to be detected as two different blips. If this distance is too short, the two objects will be detected as only one blip. Analogously, the angular resolution describes the perpendicular distance to the radar beam[24]. The angular resolutions of the two radars at a given distance define a rectangle that can be seen as a 2D resolution cell of the radar system in the given distance. In theory, the detected object can be located anywhere inside this resolution cell. However, the boundaries for the range and angular resolution are defined by the 3 dB beam width, i.e., the beam has attenuated to a half of its peak value at the boundaries[24]. This implies that a probability of the object detection is the largest in the center of the resolution cell, and it decreases towards the edges. The frame size at a given distance for the camera can be calculated when the angle of view of the lens is known. The effective frame size also depends on a crop factor of the sensor of a given camera. If a camera with a full frame (FF) sensor is used, the crop factor is 1, and other wise it is usually expressed with a number greater than one. Thus, the reciprocal of the crop factor is used in calculations. However, the rectangular area of interest in this paper is smaller than the effective frame size because the focusing points of the camera do not cover the whole frame area. The camera frame and its focusing points are illustrated in Fig. 4 . The larger square in the center denotes that the midmost focusing point is currently selected, but all of the focusing points can be selected simultaneously.

The size of the 2D angular resolution cell of the radar system and the size of rectangle covered by the focusing points at given distances are presented in Table 3 . The values for all of the rectangles are given as 2D, i.e., [horizontal vertical]. Focusing rectangle is given for FF sensor and for 1.6 crop sensor, respectively. All units in the table are in meters. It can be seen from the table that the horizontal resolution of the radar system is smaller than both focusing rectangles, but the vertical resolution of the radar system is clearly larger than the focusing rectangle of the 1.6 crop sensor, and it is also slightly larger than the focusing rectangle of the FF sensor, but the latter difference is insignificant. As a result, some of the detected objects may be outside of the focusing rectangle, if the used camera has the 1.6 crop sensor. The center-weighted probability distribution of the object detection should mitigate this possibility.

The automatic image collection is implemented by the motorized video head and WGS84 coordinates provided by the radar system. The system for automatic image collection is also depicted in Fig. 1 . The motors of the video head have only seven selectable speeds, rendering impossible to track a flying bird at the speed it flies. Thus, the steering software computes a lead point, where the camera should be turned in order to take images. Successful image collection requires that a target bird has a constant trajectory. Constant trajectory means that the flight path of a target

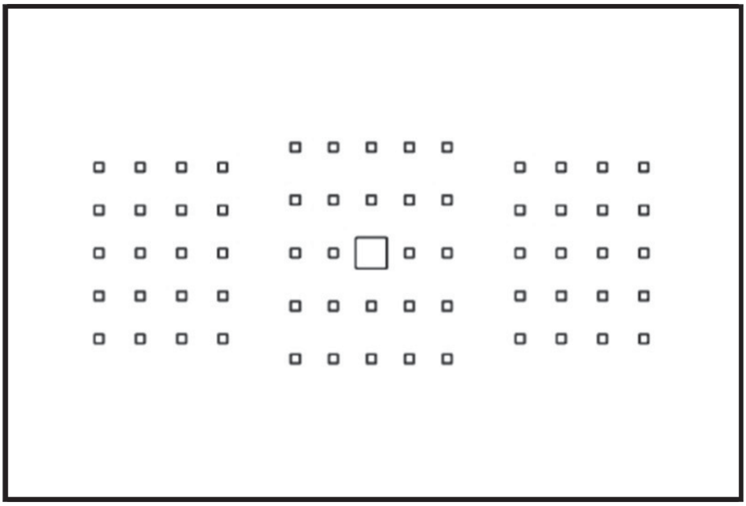


FIGURE 4 The focusing points of the applied camera may be seen, by and large, to form a rectangle over the area they cover.

TABLE 3 The sizes of the 2D angular resolution cell and the rectangle covered by the focusing points at a given distance.

Distance	FF	1.6 crop	2D Angular Resolution
100	[5.333 1.540]	[3.333 0.962]	[3.141 1.658]
200	[10.666 3.079]	[6.666 1.925]	[6.283 3.316]
300	[15.999 4.619]	[10.000 2.887]	[9.425 4.974]
400	[21.332 6.158]	[13.333 3.849]	[12.566 6.632]
500	[26.665 7.698]	[16.666 4.811]	[15.708 8.290]
600	[31.999 9.238]	[19.999 5.774]	[18.850 9.948]
700	[37.332 10.777]	[23.332 6.736]	[21.991 11.606]
800	[42.665 12.317]	[26.665 7.698]	[25.133 13.265]
900	[47.998 13.857]	[29.999 8.660]	[28.274 14.923]
1000	[53.331 15.396]	[33.332 9.623]	[31.416 16.581]
1100	[58.664 16.936]	[36.665 10.585]	[34.558 18.239]
1200	[63.997 18.475]	[39.998 11.547]	[37.699 19.897]
1300	[69.330 20.015]	[43.331 12.509]	[40.841 21.555]
1400	[74.663 21.555]	[46.665 13.472]	[43.982 23.213]
1500	[79.996 23.094]	[49.998 14.434]	[47.124 24.871]
1600	[85.330 24.634]	[53.331 15.396]	[50.265 26.529]

bird should be invariable enough for a certain period of time. The duration of this time period depends on a time delay between the timestamp of a track and the current time. Data from the radar system with respect to a single target is called a track. This data consist of WGS84 coordinates, altitude, and velocity of a target. We discovered with the used radar system that the time delay varies between 2 and 16 seconds. The probability distribution of the delay is shown in Fig. 5 . From the figure it is apparent that the time interval between 3 s and 4 s has the largest probability, i.e., 30.84 percent of the time delays fall in this time interval. More than half (56.13 percent) of the time delays fall in the intervals between 2 s and 5 s. In terms of aiming the camera, longer than 5 s delay is very unpredictable because the prerequisite of constant trajectory may no longer stand.

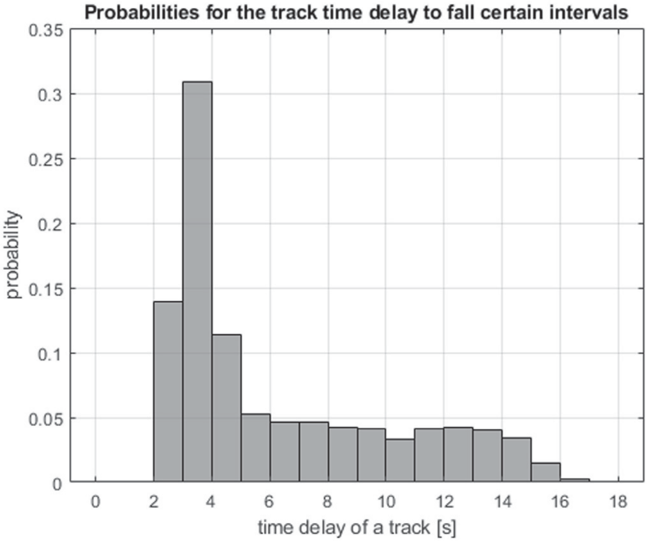


FIGURE 5 The probability distribution of the time delay between the timestamp of a track and the current time.

The coordinates are given in decimal degrees instead of degrees, minutes, and seconds (DMS). DMS values are converted into decimal degrees as follows:

$$D_d = D + M \cdot 60 + S \cdot 3600 \tag{2}$$

where D_d is the value in decimal degrees, D is the value of degrees, M is the value of minutes, and S is the value of seconds.

The radius of the semi-major axis of the Earth at the equator is 6378137.0 m, and the circumference is 40075161.2 m. The equator is divided into 360 degrees of longitude, so that each degree at the equator represents 111319.9 m. This number, representing degrees in meters at the equator, is multiplied by the cosine of the latitude when a given latitude is moved from the equator towards the poles. This means that the number representing degrees in meters decreases as the latitude increases. Finally, the number is zero when either one of the poles is reached. Longitudes are positive to the east of a prime meridian (Greenwich, London, a.k.a zero meridian) and negative to the west of it. As the WGS84 reference geoid is applied, one arc minute along a meridian or along the Equator is 1855.3 m[25]. Accordingly, precisions for latitude in meters are presented in Table 4 . The latitude of the test site is approximately 60 degree. Precisions for the longitude at this latitude are also given in the table. The radar system gives coordinates with eight decimal places, and it can be seen from the table that this means 1.1 × 10^{-3} in meters of accuracy for latitude and 5.6 × 10^{-4} in meters of accuracy for longitude. Thus, the accuracy of the given coordinates is adequate for correct aiming.

TABLE 4 Precision of latitudes and longitudes in meters.

Decimal Degrees	DMS	Precision Lat	Precision Lon
1.0	1°00’00"	111 120	55 560
0.1	0°06’00"	11 112	5 556.0
0.01	0°00’36"	1 111.2	555.60
0.001	0°00’03.60"	111.12	55.560
0.000 1	0°00’00.36"	11.112	5.556
0.000 01	0°00’00.036"	1.1112	0.555 6
0.000 001	0°00’00.0036"	0.11112	0.055 56
0.000 000 1	0°00’00.00036"	0.011112	0.005 556
0.000 000 01	0°00’00.000036"	0.0011112	0.000 555 6

6.1 | Targeting the Camera

The video head used in this application was not designed for this type of use, which caused some problems. The video head cannot be steered by entering the desired horizontal and vertical angles, but it requires the driving time of the motors (separate motors for horizontal and vertical movement) in seconds. The video head has a fixed home position, which is at halfway in the steering range for both directions. The head is installed so that at the home position the camera is horizontally pointing to the west (bearing = 270°), and vertically so that the vertical turning angle at the home position is zero. We found in our tests that the video head has an increasing error towards each steering direction. In addition, this error is larger in horizontal steering than in vertical steering, and it also depends on which direction the head is steered from the home position. As a result, a method for targeting the camera by the head was needed in order to compensate the errors. We used the locations of the wind turbines in the test area as reference locations for error correction, because they are fixed and their exact WGS84 coordinates are known. Distances of the wind turbines range from 600 m to 2000 m from the camera location, resulting in relatively large error in meter with only a small error in turning angle.

We assumed that the errors are linear and used the least squares method (LSM) for finding the parameters, *k* and *b*, of lines that minimize these errors. This was done separately for the horizontal steering angle left and right from the home position. A constant was used as an error correction to the vertical turning angle, because it seems to have very small error. In addition, the actual vertical turning angle error was obstructed by the erratic trajectory of some bird species, and it was further amplified by the time delay between the timestamp of tracks and real time. In practice, we discovered that the error correction is more conveniently implemented to the horizontal and vertical turning angles rather than to the respective steering times, because computations for steering times are based on the turning angles. In horizontal steering, the idea is to find a line that gives a correction to the computed horizontal turning angle when the bearing (a compass direction that the head should be pointing) has been computed first. This is given by:

$$c_h = kB + b, \tag{3}$$

where c_h is correction in radians, *k* is slope, *B* is bearing in radians, and *b* is bias. The corrected value of the horizontal turning angle is,

$$t_{ch} = a_h + c_h, \tag{4}$$

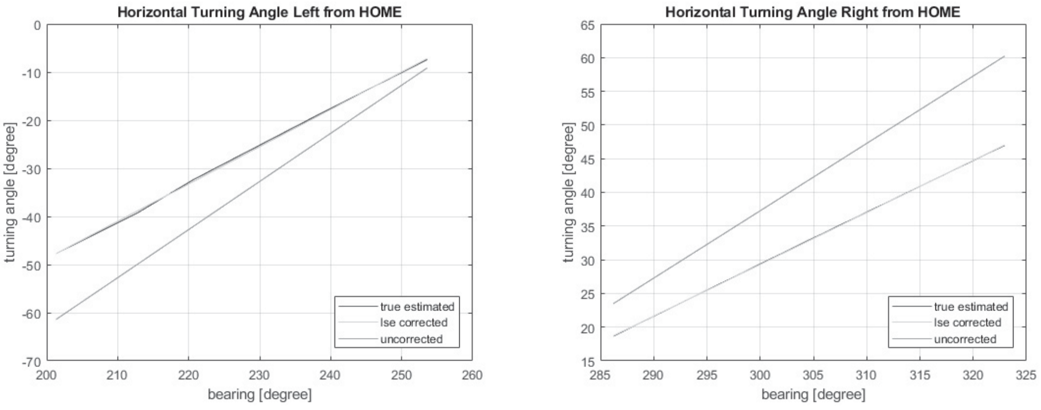
where t_{ch} is the corrected horizontal turning angle in radians, a_h is computed turning angle based on the WGS84 coordinates of a target, and c_h is as in equation 3 . In equation 4 , a_h is given:

$$a_h = \begin{cases} < 0, & \text{if } B < \frac{3}{2}\pi \text{ rad}, \\ > 0, & \text{if } B > \frac{3}{2}\pi \text{ rad}, \\ = 0, & \text{if } B = \frac{3}{2}\pi \text{ rad}, \end{cases} \tag{5}$$

where *B* is as in equation 3 . We have computed horizontal estimates of true turning angle for each reference location by measuring errors in pixels from test images. As the frame size of the camera and the distance are known, error in meters can be computed. These test images are taken automatically by the developed system, and aiming is perfect when the rotor hub of a wind turbine is in the center of these test images. See algorithm 1 for finding optimal values for the parameters, *k* and *b*. Figure 6 presents the estimate for the horizontal true turning angle, the computed horizontal turning angle without correction, and the corrected horizontal turning angle for each reference location, respectively. Figures 6a and 6b show that the error compensation was successful, i.e., the green lines (corrected turning angle) are on the blue lines (estimated true turning angle).

Algorithm 1 Pseudocode for finding optimal values of parameters *k* and *b*.

```
initialize(sumOfsquaredErrors, threshold)
initialize(EstimatedTrueTurningAngle, computedTurningAngle)
for $\text{sumOfsquaredErrors} > \text{threshold}$ do
    errorsOfReferences = computedTurningAngle - EstimatedTrueTurningAngle
    $[\text{k}, \text{b}]$= LSM(errorsOfReferences, bearing)
    computedTurningAngle = adjustTurningAngle(computedTurningAngle, k, b)
    run system with computedTurningAngle    //started manually
    take image of each reference wind turbine    //automatically by the started system
    errorsInPixels = measure horizontal and vertical errors from the images in pixels    //manually
    errorsInMeters = compute(errorsInPixels)
    sumOfsquaredErrors = LSE(errorsInMeters)
    update(EstimatedTrueTurningAngle, errorsInMeters)
end for
```



(a)

(b)

FIGURE 6 Estimated true, uncorrected and corrected horizontal turning angles for the reference wind turbine locations.

7 | CLASSIFICATION

Deep learning is a subfield of machine learning concerned with algorithms called artificial neural networks inspired by the structure and function of the brain. Deep learning networks are typically large, and they have many layers and large number of parameters. A CNN is one implementation of deep learning[23]. A CNN is a specialized kind of neural network for processing data that has a known grid-like topology like image data, which can be thought of as a 2D grid of pixels. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers[23].

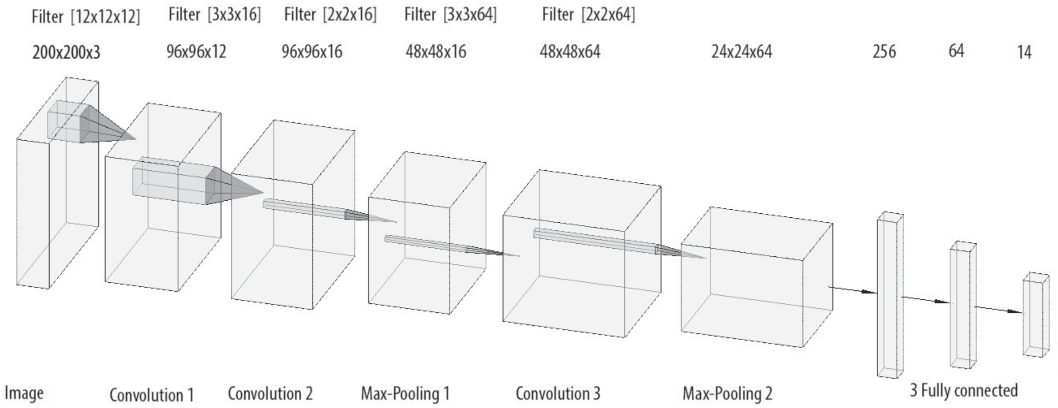


FIGURE 7 The CNN architecture.

7.1 | CNN Architecture

We applied a CNN for image classification and built an architecture of three convolution layers, each of which is followed by a rectified linear unit (ReLU) layer and the first two are also followed by a cross-channel normalization layer. Figure 7 shows the CNN architecture. The use of cross-channel normalization layer is motivated by its ability to aid the generalization as its function may be seen as brightness normalization[2]. There are two max-pooling layers, the first is before the third convolution layer and the second is before the first fully connected layer. There is no max-pooling layer before the second convolution layer. The reason for this is that by omitting a max-pooling layer, all of the finest edges detected by the first convolution layer are also transferred to the second convolution layer. The architecture is completed by three fully connected layers, and the first two of them are followed by dropout layers, and each dropout layer is followed by ReLUs. The dropout was implemented by randomly setting the output neurons of the layer to zero with a probability of 0.5. The architecture is finally terminated by softmax activation, which produces a distribution over the class labels with cross entropy loss function[26]. The input image is normalized and zero-centered before feeding it to the network. CNN with Mini-batch training and supervised mode as well as stochastic gradient descent with momentum is applied[27,28,29,26]. The L2 Regularization (weight decay) method for reducing over-fitting is also applied[29,26,30]. Due to limited capacity of computer resources the network size in terms of free parameters is kept small. Thus, resulting in total of 92 feature maps which are extracted by convolution layers with kernel sizes $[12 \times 12 \times 3] \times 12$, $[3 \times 3 \times 12] \times 16$, and $[3 \times 3 \times 16] \times 64$, respectively. Total number of weights is about 9.47×10^6.

Images of a size 200×200 pixels are fed to the classifier. In the first convolution layer, this image size produces $(200 - 12 + 2 \times 1)/2 + 1 = 96$ square feature maps (see equation 1), i.e., there are $96 \times 96 = 9216$ neurons in each feature map. Filter (kernel) size, number of feature maps, feature map size in neurons, stride, and padding for each convolution layers and max-pool layers are given in Table 5. For each filter, Fig. 7 displays the number of feature maps as the triplet [a, b, c].

TABLE 5 Parameters for the convolution layers and the max-pooling layers of the CNN model.

Layer	Filter	# Feature Maps	Feature Map Size	Stride	Padding
Convolution 1	12 × 12	12	96 × 96	[2 2]	[1 1]
Convolution 2	3 × 3	16	96 × 96	[1 1]	[1 1]
Max-pooling 1	2 × 2	16	48 × 48	[2 2]	[0 0]
Convolution 3	3 × 3	64	48 × 48	[1 1]	[1 1]
Max-pooling 2	2 × 2	64	24 × 24	[2 2]	[0 0]

7.1.1 | Tested CNN Structures and Hyperparameter Selection

In addition to the CNN model introduced above, we have tested the svm-on-top model, which has otherwise the same structure as this CNN model but a svm classifier was connected to the second fully-connected layer[22]. However, the tests for this study showed that the classification performance is the same or even slightly better when using only the CNN. More over, the svm-on-top model had significantly longer training time than the CNN model alone. We have also tested the applied CNN model with four and five convolution layers, respectively. Results of these tests showed that the classification performance did not increase according to the number of the convolution layers. This implies that the three-layered structure, trained on the used training dataset, is capable to extract all the relevant features from the training data.

We split the dataset into a training set and a validation set as 70% and 30%, respectively. We have selected hyperparameters as follows: the initial weights for all layers were drawn from the Gaussian distribution with mean 0 and standard deviation 0.01, initial biases were set to zero, the L2 value was set to 0.0005, and mini-batch size was set to 128. The values of these hyperparameters were fixed and we used manual tuning only for choosing the combination of the number of epochs and the learning rate drop period (LRDP) in our tests. The learning rate decay schedule (LRDS) method was also included in the tests. Motivation for using the LRDS method is as training proceeds with shorter leaps on the loss function surface from some point on, the optimal value for the weights can be found more accurately. If only the short leaps would be applied, the number of epochs should be very large, thus resulting in significant increase of training time. For more details, see[22]. Although the tests in our previous study were performed on the svm-on-top model, the results should be applicable with the CNN alone model as well, because these parameters affects only the CNN part of the structure. The outcome of the tests was that the number of epochs can be selected only empirically, albeit the number of training examples should give some insight into the decision. Thus, as the number of training examples increases, the number of epochs should decrease in order to avoid overfitting. Whether to keep learning rate fixed or not (i.e., use LRDP) is also to be chosen empirically. The tests showed that the number of epochs, the number of training examples, and the number of classes have the most significant contribution to the classification performance. For the aforementioned reasons, we kept the learning rate fixed (0.01), and we selected the number of epochs empirically in accordance with the number of training examples and the number of classes (11).

7.2 | Final Classification

The system takes series of images of each target bird in order to increase a probability that the target is in good position for identification. The result of the image classifier is presented as a vector, thus prediction for each image in a single series of images are combined into the vector, $\boldsymbol{P}_{\mathrm{i}}$:

$$\boldsymbol{P}_i = [c_1, c_2, ..., c_{\mathrm{nc}}], i = 1, ..., n, \tag{6}$$

where c_{j} is a probability of belonging to *class j*, *nc* is the number of classes and *n* is the number of images in each series. The combined ***P***-vector for a series of images is:

$$\boldsymbol{P} = \sum_{i=1}^{n} \boldsymbol{P}_i, \tag{7}$$

The velocity of the target bird is presented as a vector with elements placed according to the class order, i.e., *class 1*, *class 2*, ..., *class nc*, where *nc* denotes the number of classes. The composition of the vector is follows: calculate the average of the velocities of the image series, check probabilities for this average velocity from the velocity-look-up table for all of the classes, set the elements (probabilities), *p*, of the velocity vector accordingly, yielding

$$\boldsymbol{V} = [p_1, p_2, ..., p_{nc}], \tag{8}$$

where *nc* is the number of classes. The final classification is achieved by adding the velocity vector of a target to the respective prediction from the image classifier. The fusion vector, Φ, is:

$$\Phi = \boldsymbol{P} + \boldsymbol{V} \tag{9}$$

The score, *S*, for final prediction is:

$$S = \mathbf{max}_j(\Phi), \tag{10}$$

$$j = \mathbf{arg\ max}_j(\Phi), \tag{11}$$

where *j* is the index of the predicted class.

8 | RESULTS

We tested the accuracy for targeting the video head by visual observations, i.e., each time when the system was started, we monitor the bird movement in the test area with binoculars and telescopes to confirm which bird the system tracks. When the camera took series of images of the target bird, the system was halted and we checked whether the target bird is in any of the images or not. The accuracy was computed by dividing the number of successful targeting by the number of all targeting (accurate + inaccurate). The accuracy for targeting the video head was 88.91 %.

We tested the classification performance of the original dataset and two augmented datasets. The number of training examples was 23552 of the original (not augmented) dataset, and it was trained with 60 epochs. The augmented datasets were generated with *s* = 50 and *s* = 200, resulting in the number of training examples to be 6147072 and 1554432, respectively. Thus, the empirically chosen number of epochs was 4 and 13, respectively. TPR is used to evaluate classification performance. Classification performance of the original, the augmented with *s* = 50, and the augmented with *s* = 200 datasets was 0.7643, 0.8514, and 0.8688, respectively. The results show that the data augmentation improved classification performance by 13.7 %. However, because the best classification performance was achieved with *s* = 200, the results also imply that from some value on of the step, *s*, overfitting increases as *s* decreases. The Confusion matrix for classes is presented in Table 6 . Receiver operating characteristic curve (ROC) is given for the two key species, the white-tailed eagle (*Haliaeetus albicilla*) and the lesser black-backed gull (*Larus fuscus fuscus* , a.k.a. the baltic gull), in Fig 8 . Area under the curve (AUC) is 0.9854 for the white-tailed eagle, and 0.8562 for the baltic gull, respectively.

Initially, we assumed that classification performance can be increased by using the parameters provided by the radar system. Among these parameters, the flight velocity of a target bird seemed to be the most usable for bird species identification. Classification of some misclassified images can be corrected when the speed of a target bird is fused with the prediction of the image classifier. However, the fusion had only slight positive weight to the final classification. This was only distinguishable in the cases where the image classifier had misclassified the white-tailed eagle as the great cormorant, and vice versa.

TABLE 6 Confusion matrix for all of the classes.

	WTEA	LOSP	GRCO	COEI	COGO	GBBG	HEGU	LBBG	COGU	BHGU	CATE
WTEA	43	0	1	0	0	0	0	0	0	0	0
LOSP	0	6	0	0	0	0	0	0	1	0	0
GRCO	2	0	95	1	1	0	0	0	1	0	0
COEI	0	0	0	17	0	0	0	0	0	0	0
COGO	0	1	1	2	17	0	0	0	0	0	0
GBBG	0	0	0	0	0	7	1	1	0	0	0
HEGU	0	0	0	0	0	1	44	1	25	0	1
LBBG	0	0	0	0	0	3	1	21	1	0	0
COGU	0	0	0	0	0	0	1	0	64	0	0
BHGU	0	0	0	0	0	0	0	0	0	31	1
CATE	0	0	0	0	5	0	0	0	0	1	26

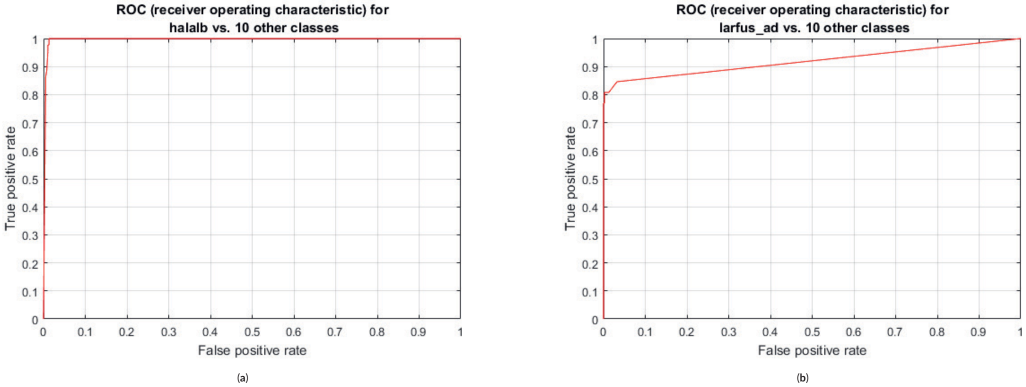


FIGURE 8 ROC curves for the white-tailed eagle (*Haliaeetus albicilla*) and for the lesser black-backed gull (*Larus fuscus fuscus*). (**a**) AUC for the white-tailed eagle is 0.9954; (**b**) AUC for the lesser black-backed gull is 0.8562.

Examples of images that the system correctly identified are presented in Fig. 9 . The last image (**c**) in this figure shows more than one bird in a (tight) flock, which is typical for images of waterfowl. Moreover, more than one bird species may occur in these flocks. A solution to the images of a flock is twofold. As result of the segmentation process, an image has only one bird left when there is a sparse flock of birds in the image. In the sparse flock case, the bird closest to the center of the image is retained, thus when a sparse flock has more than one bird species, the retained bird species is chosen randomly. In the tight flock case, the classification is based on the whole flock and thus it is biased toward the most numerous bird species in the flock. The training dataset includes examples of these tight flocks.

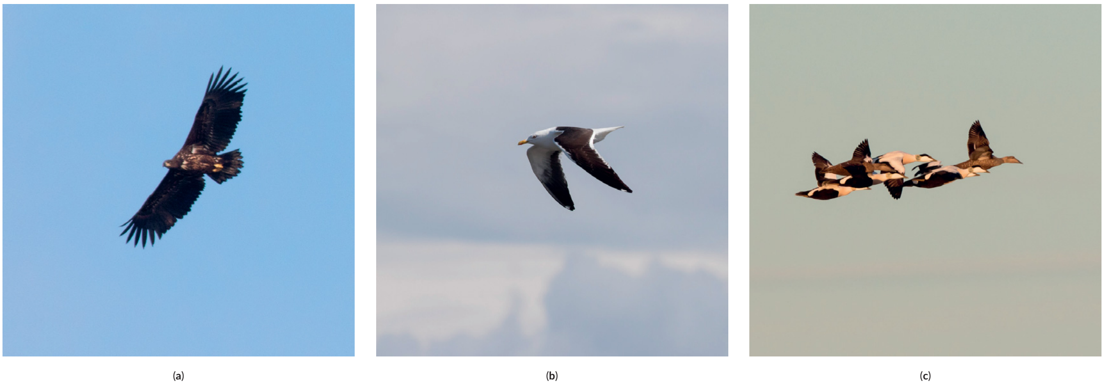


FIGURE 9 Examples of correctly identified images of the white-tailed eagle (**a**), the baltic gull (**b**), and the common eider (**c**), respectively.

9 | DISCUSSION

The delay between the timestamp of a track and the current time seems to be the most significant factor for inaccurate targeting of the camera. This is further evident in higher wind speed circumstances. We discovered by visual observations that flight paths of all bird species in the test area are increasingly erratic in severe weather conditions. This augments the effect of the delay in respect of targeting of the camera. Table 7 demonstrates the range that a flying bird with a given flight speed travels during the delay. Flight speeds are measured by the radar system.

TABLE 7 Ranges for a flying bird with a given flight speed during delay.

	Flight Speed = 8 m/s	Flight Speed = 15 m/s	Flight Speed = 25 m/s
Delay [s]	Range [m]	Range [m]	Range [m]
2	16	30	50
3	24	45	75
4	32	60	100
5	40	75	125
6	48	90	150
7	56	105	175
10	80	150	250
13	104	195	325
16	128	240	400

The precise vertical angle error was difficult to estimate because of the stochastic element of a target bird position, which is caused by the delay. Therefore, the constant vertical error compensation was discovered empirically during the targeting of the camera. An error of 0.25° means 1.31 m vertical range at the distance of 300 m. It can be seen from Table 3 that the focusing rectangle for a camera with the 1.6 crop factor is vertically 2.89 m at this distance. The position of a target can be 1.445 m (2.89/2 m) above or below from the horizontal center line, and regardless of the 0.25° error in the vertical turning angle, it will be inside the focusing rectangle. The probability that target birds will be captured in the focusing rectangle (given that all other errors are compensated) from any distance is large if the vertical turning angle error is 0.25° or smaller. If this error is greater than 0.25°, target birds will probably be missed.

We had problems with the autofocus system of the camera and the lens. This seemed to be linked to the far range that the images are taken from. If the autofocus system cannot reach the correct focus, the camera does not release the shutter, and therefore images are not taken. We did not find any solution from the API to release the shutter even if the correct focus is not reached. This is important because the classifier showed to have good generalization ability, when it was fed by slightly out-of-focus images. The far photographing range means deep depth of field, which is the zone of sharpness within an image that will appear in focus. When focal length of the lens and aperture (f-stop) are kept fixed, photographing range where images are taken from is the only factor that affects to depth of field. The further away a subject is from the camera, the deeper depth of field becomes. Based on above, we switched the autofocus system off and used manual focusing as a solution to the autofocus problem. We focused to a target at chosen range before the system was started and kept the autofocus system off. As a result, refocusing was needed when the photographing range was changed. The camera and the lens that we have used are both two generations old. The most recent camera and lens generation has better autofocus system, which may be a long term solution to this autofocus problem.

The applied CNN model showed sufficient performance as an image classifier. Mostly, it misclassified images that were totally out-of-focus or images in which the bird is in unfavourable position in the sense of identification even by the human eye. As we expected, the greatest challenges for the classifier were the two pairs of the following bird species: the herring and the common gull, the lesser black-backed and the great black-backed gull, which are not generally easy to identify [31] . Figure 10 shows three herring gull images; the leftmost (10a) image was correctly classified, and the other two were misclassified as the common gulls.

10 | CONCLUSIONS

The developed bird identification system is evaluated in terms of bird detection and identification (classification) ability. Bird detection was two-fold: initially, the applied radar system detected birds in the sky, and the developed steering software for the motorized video head drove the camera to the correct position. We showed successful results for the latter part of the detection problem. The radar system was beyond the scope of this study. We presented practical results for automatic bird identification by using real-world dataset and representative methods in machine learning (CNN). However, more research is needed for reducing the time delay between the timestamp of a track and the current time. The image classifier is not perfect but its performance is sufficient to classify one of the key species, the white-tailed eagle correctly. More complicated classifier is needed in order to classify the different gull species with higher performance. Hierarchical and/or cascade architecture of classifiers may be one solution to this problem. This system is the first part of the planned entirety for automatic bird monitoring and deterrence for offshore wind farms, and it can mitigate the social opposition of wind energy.

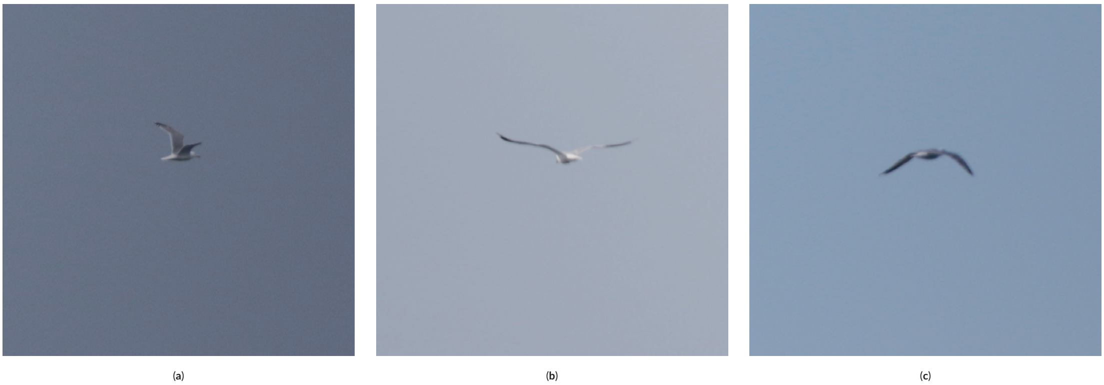


FIGURE 10 Three images of the herring gull. The first (**a**) image was correctly classified. The other two images (**b**, **c**) are examples of images in which the bird is in unfavourable position in the sense of identification resulting in misclassification.

ACKNOWLEDGMENTS

The authors wish to thank Suomen Hyötytuuli for the financial support of purchasing necessary equipment and Robin Radar Systems for the technical support with the applied radar system.

Author contributions

Conceptualization, J.N.; Data Curation, J.N.; Formal Analysis, J.N.; Funding Acquisition, J.T.T.; Investigation, J.N.; Methodology, J.N.; Project Administration, J.N.; Software, J.N.; Supervision, J.T.T.; Validation, J.T.T.; Visualization, J.N.;Writing-Original Draft, J.N.;Writing-Review and Editing, J.T.T.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Baxter AT, Robinson AP. A comparison of scavenging bird deterrence techniques at UK landfill sites. *Int. J. Pest Manag* 2007; 53(3): 347–356. https://doi.org/10.1080/09670870701421444.

2. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 2017; 60(6): 84–90.

3. Rachmadi RF, Uchimura K, Koutagi G, Komokata Y. Japan road sign classification using cascade convolutional neural network. In: ITS (Intelligent Transport System) World Gongress. ITS. ; 2016; Tokyo.

4. Mirzaei G. *Data Fusion in Multi-Sensory Environment of Infrared, Radar, and Acoustics Based Monitoring System*. PhD thesis. University of Toledo, Toledo, OH; 2013.

5. Wiggelinkhuizen EJ, Rademakers LWMM, Barhorst SAM, Boon dHJ. Bird collision monitoring system for multi-megawatt wind turbines WT-Bird: Prototype development and testing. report, Energy research Center of the Netherlands; LE Petten, Netherlands: 2006a. Report ECN-E-06-027.

6. Wiggelinkhuizen EJ, Rademakers LWMM, Barhorst SAM, Boon dHJ, Dirksen S, Schekkerman H. WT-Bird: Bird collision recording for offshore wind farms. report, Energy research Center of the Netherlands; LE Petten, Netherlands: 2006b. Report ECN-RX-06-060.

7. Wiggelinkhuizen EJ, Boon dHJ. Monitoring of bird collisions in wind farm under offshore-like conditions using WT-BIRD system, final report. report, Energy research Center of the Netherlands; LE Petten, Netherlands: 2013. Accessed on 11.02.2019.

8. Rosa IMD, Marques AT, Palminha G, et al. Classification success of six machine learning algorithms in radar ornithology. *Ibis* 2016; 158: 28-42.

9. DTBird. =http://www.dtbird.com/; 2010.

10. May R, Hamre O, Vang R, Nygård T. Evaluation of the DTBird video-system at the Smøla wind-power plan. report, Norwegian Institute for Nature Research - NINA; Trondheim, Norway: 2012. Accessed on 25.11.2019.

11. Berg T, Liu J, Lee SW, Alexander ML, Jacobs DW, Belhumeur PN. Birdsnap: Large-Scale Fine-Grained Visual Categorization of Birds. In: IEEE; 2014; San Diego, CA

12. Berg T, Belhumeur PN. POOF: Part-Based One-vs.-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation. In: CVPR. IEEE; 2013; Portland, OR

13. Yoshihashi R, Kawakami R, Iida M, Naemura T. Evaluation of Bird Detection using Time-lapse Images around a Wind Farm. *Wind Energy* 2017; 20(12): 1983-1995. doi: 10.1002/we.2135

14. Freund Y, Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory* 1995; 904: 23-37.

15. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: CVPR 2001. IEEE; 2001; Kauai, HI: 511-518

16. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: CVPR'05. IEEE; 2005; San Diego, CA: 886-893

17. Robin radar models. =http://www.robinradar.com; 2016.

18. Pelco-D protocol. =http://bruxy.regnet.cz/programming/rs485/pelco-d.pdf; 2016.

19. Niemi J, Tanttu JT. Automatic Bird Identification for Offshore Wind Farms. In: Wind Energy and Wildlife Impacts. CWW2017. Springer; 2019; Cham: 135–151. ISBN 978-3-030-05519-6, https://www.springer.com/us/book/9783030055196.

20. Malling Olsen K, Larsson H. *Terns of Europe and North America*. London: Helm . 1995. ISBN 0-7136-4056-1.

21. Jarrett K, Kavukcuoglu K, Ranzato MA, LeCun Y. What is the best multi-stage architecture for object recognition. In: IEEE 12th International Conference on Computer Vision, ICCV 2009. ICCV 2009. IEEE; 2009; Kyoto, Japan: 2146–2153.

22. Niemi J, Tanttu JT. Deep Learning Case Study for Automatic Bird Identification. *Applied Sciences* 2018; 8(11)(2089). https://doi.org/10.3390/app8112089.

23. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press . 2016. http://www.deeplearningbook.org.

24. Richards MA. *Fundamentals of Radar Signal Processing*. New York, NY: The McGraw-Hill companies . 2005. ISBN 0-07-144474-2.

25. NGA . World Geodetic System 1984. report, National Geospatial-Intelligence Agency; : 2004. Accessed on 14.02.2019, http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html.

26. Bishop CM. *Pattern Recognition and Machine Learning*. New York, NY: Springer . 2006. ISBN 978-0-387-31073-2.

27. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. In: No. 11 in Proceedings of the IEEE 26. IEEE. ; 1998: 2278–2324. https://doi.org/10.1109/5.726791.

28. Li M, Zhang T, Chen Y, Smola AJ. Efficient Mini-batch Training for Stochastic Optimization. In: The 20th ACM SIGKDD International Conference on Knowledge, Discovery and Data Mining. KDD 2014. ; 2014; New York, NY: 661–670. ISBN 978-1-4503-2956-9.

29. Murphy KP. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press . 2012. ISBN 978-0-262-01802-9.

30. Haykin S. *Neural Networks: A Comprehensive Foundation*. New York, NY: Prentice Hall/Pearson. 2nd ed. 1994. ISBN 0-13-908385-5.

31. Malling Olsen K, Larsson H. *Gulls of Europe, Asia and North America*. London: Helm . 2003. ISBN 0-7136-4377-3.

□

PUBLICATION
V

Deep Learning Case Study on Imbalanced Training Data for Automatic Bird Identification

J. Niemi and J. T. Tanttu

Deep Learning: Algorithms and Applications. Ed. by C. Shyi-Ming and W. Pedrycz. 2019
DOI: 10.1007/978-3-030-31760-7

Deep Learning Case Study on Imbalanced Training Data for Automatic Bird Identification

Juha Niemi, Juha T. Tanttu

Faculty of Computer and Information, Tampere University, Finland
juha.k.niemi@tuni.fi
juha.tanttu@tuni.fi

Abstract. Collisions between birds and wind turbines can be significant problem in wind farms. Practical deterrent methods are required to prevent these collisions. However, it is improbable that a single deterrent method would work for all bird species in a given area. An automatic bird identification system is needed in order to develop bird species level deterrent methods. This system is the first and necessary part of the entirety that is eventually able to, monitor bird movements, identify bird species, and launch deterrent measures. The system consists of a radar system for detection of the birds, a digital single-lens reflex camera with telephoto lens for capturing images, a motorized video head for steering the camera, and convolutional neural networks trained on the images with a deep learning algorithm for image classification. We utilized imbalanced data because the distribution of the captured images is naturally imbalanced. We applied distribution of the training data set to estimate the actual distribution of the bird species in the test area. Species identification is based on the image classifier that is a hybrid of hierarchical and cascade models. The main idea is to train classifiers on bird species groups, in which the species resembles more each other than any other species outside the group in terms of morphology (coloration and shape). The results of this study show that the developed image classifier model has sufficient performance to identify bird species in a test area. The proposed system produced very good results, when the hybrid hierarchical model was applied to the imbalanced data sets.

Keywords: Machine learning, Deep learning, Convolutional neural networks, Classification, Data augmentation, Intelligent surveillance systems

1 Introduction

Demand for automatic bird identification systems for wind farms has increased recently. This kind of system is especially required for offshore wind farms. The objective of this application is twofold: it has first to detect two key bird species, which are particularly required for monitoring in the environmental license, and secondly to classify maximum number of other bird species while the first part still stands. The two key species are the white-tailed eagle (*Haliaeetus albicilla*) and the lesser black-backed gull (*Larus fuscatus fuscatus*). An automatic identification system is in development that consists of a separate commercial radar system to detect the birds, a digital single-lens reflex camera with telephoto lens for capturing images, a motorized video head for steering the camera, and a convolutional neural network (CNN) trained on the images with a deep learning algorithm for image classification. The conventional approach to this image classification problem is to presume that equally distributed data are fed into the classifier. However, this is a real-world application, in which it is difficult and time-consuming to collect large number of images for each class. Due to the nature of this application, it is conceivable that imbalanced data are utilized because the distribution of the captured images is naturally imbalanced, i.e., there are common and scarce bird species in the test area. It is also possible to include scarcer classes into the classification process with this approach. Researchers have proposed a class-imbalance aware loss function for the problem of class imbalance. This loss function adds an extra class-imbalance aware regularization term to the normal softmax loss [1]. However, we have applied the distribution of the training data set to estimate the actual distribution of the bird species in the test area. Training data set and test data set both share this distribution. Species identification is based on the image classifier that is a hybrid of hierarchical and cascaded models. The main idea is to train classifiers on bird species groups, in which the species resembles more each other than any other species outside the group in terms of morphology. The first classifier is hierarchical determining the group of the test image and the subsequent classifiers within the groups are in cascades. We have also applied our data augmentation method, which rotate and convert the images in accordance with the desired color temperatures. The hybrid hierarchical and cascade model is compared to two single classifiers. One of the classifiers is trained on balanced data set and the other is trained on imbalanced data set without grouping.

CNN has been successfully applied to image classification problems [2]. The number of training examples in image classification is typically large. This may cause problems when dealing with real-world applications, as collection of large number of images is not always possible. As a result, some data augmentation is usually needed [3,4]. Cascade CNN has been successfully applied to face detection and road-sign classification system [5,6].

The remainder of this chapter is organized as follows. In Section 2. we present the system and its components for collecting images automatically. In Section 3. related work is discussed. We describe our data, its grouping, and data augmentation algorithm in Section 4. Classification algorithms, applied CNN models, and feature extraction are described in Section 5. Results for hybrid of hierarchical and cascade CNN model, trained on imbalanced data set and compared to conventional CNN model, are presented in Section 6. We then offer conclusions in Section 7.

2 The system

The proposed system consists of several hardware as well as software modules. See Fig. 1 for an illustration. At first, there is the radar system, which is connected to

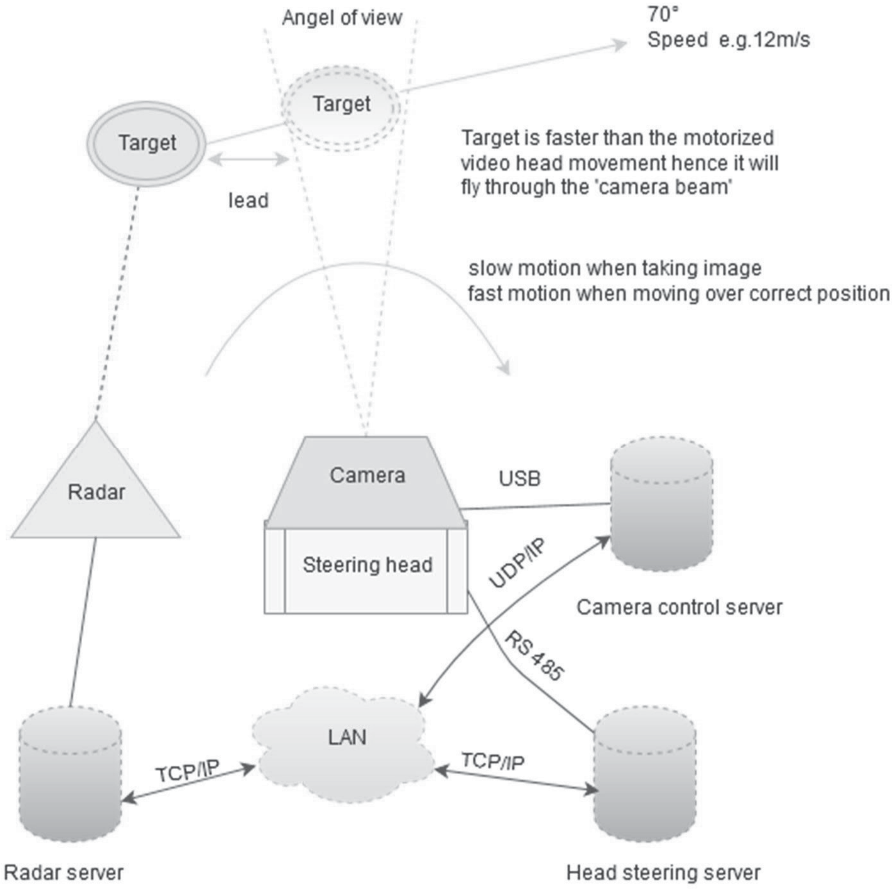

Angel of view
70°
Speed e.g.12m/s
Target
Target
lead
Target is faster than the motorized video head movement hence it will fly through the 'camera beam'
slow motion when taking image
fast motion when moving over correct position
Radar
Camera
Steering head
USB
UDP/IP
RS 485
Camera control server
LAN
TCP/IP
TCP/IP
Radar server
Head steering server

Fig. 1. The hardware of the system and the principle of catching flying bird into the frame area of the camera.

a local area network (LAN), and thus it is able to communicate with the servers, in which the various programs are running.

We use a radar system supplied by Robin Radar Systems B.V. because they provide an avian radar system that is able to detect birds. They also have algorithms for tracking a detected object over time (between the blips). The model we use is the ROBIN 3D FLEX v1.6.3 and it is actually a combination of two radars and a software package for implementation of various algorithms such as the tracker algorithms [7]. The role of the radar is to detect flying birds and pass the WGS84 coordinates of the target bird to the video head control software. The system includes the PT-1020 Medium Duty motorized video head of the 2B Security Systems. The video head is operated by Pelco-D control protocol [8], and the control software for it is developed by us. The System uses Canon EOS 7D II camera with 20.2-megapixel sensor and the Canon EF 500/f4 IS lens. Correct focusing of the images relies on the autofocus system of the lens and the camera. Automatic exposure is also applied. The camera is controlled by the application programmable interface (API) of the camera manufacturer, and the software for controlling the camera has been developed by us. In addition, the radar system provides parameters, which can be applied to increase the performance of classifiers. These parameters are the distance in 3D of a target (m), velocity of a target (m/s), and trajectory of a target (WGS84 coordinates). For the details of the system hardware, see [9].

3 Related work

Researchers proposed a multi-sensor data fusion approach via acoustics, infrared camera, and marine radar for avian monitoring. The objective is to preserve the population of birds and bats especially those listed in endangered list, by observing their activity and behavior over the migration period. Species-level identification was not aimed mainly. They address to this problem by a fuzzy Bayesian based multi-sensory data fusion approach to provide the activity information regarding the targets in avian (birds and bats) monitoring [10].

Researchers have implemented machine learning (ML) algorithms on radar data for bird species classification. They used data collected from two locations in Portugal with two marine radar antennas (volume search radar, VSR and high sensitivity reception, HSR). The performance of six widely used ML algorithms: random forests (RF), support vector machine (SVM), artificial neural networks, linear discriminant analysis, quadratic discriminant analysis, and decision trees (DT), was tested. They found that all algorithms performed well (area under the receiver operating characteristic and accuracy > 0.80, P < 0.001) when discriminating birds from non-biological targets such as vehicles, rain or wind turbines, but the algorithms showed greater variance in their performance when they classified different bird functional groups or bird species (e.g. herons vs. gulls). In this study, only RF was able to hold an accuracy > 0.80 for all classification tasks, although SVM and DT also performed well. All algorithms correctly classified 86% and 66% (VSR and HSR, respectively) of the target points, and only 2% and 4% of these points were misclassified by all algorithms. The results suggest that ML algorithms are suitable for classifying radar targets as birds, and thereby separating them from other non-biological targets. The ability of these algorithms for correct identification between bird species functional groups was much weaker [11].

Time-lapse photography is a method in which the frame rate of taking a sequence of images is higher than the frame rate used to view the sequence. Time-lapse images can make subtle time-related processes distinct, and the process that is analyzed, can be too fast or too slow to the human eye. Time-lapse images have been used to detect birds around a wind farm. An Image-based detection using cameras have been applied to build a bird monitoring system. This system utilized an open-access time-lapse image data set that is collected around the wind farm. The system applied algorithms: AdaBoost, Haar-like, histogram of oriented gradients (HOG), and CNN. AdaBoost is a two-class classifier, which is based feature selection and weighted majority voting.

A strong classifier is made as a weighted sum of many weak classifiers, and the resulting classifier is shallow but robust [12]. Haar-like is an image feature that utilizes contrasts in images. It extracts the light and the shade of objects by using black-and-white patterns [13]. HOG is a feature used for grasping the approximated shape of objects. At first, it computes the spatial gradient of the image and makes a histogram of the quantized direction of the gradient in each local region, called a cell in the image. Subsequently, it concatenates the histograms of the cells in the neighboring groups of the cells (the blocks) and normalizes them by dividing by their Euclidean norms in each block [14]. The best method for detection was Haar-like, and the best method for classification was CNN. The system was tested on only two bird functional groups, hawks and crows, and it achieved only moderate performance [15].

4 Data

Input data of this application consist of digital images. All images for training the CNN have been taken manually at the test location in various weather conditions. The location is the same where the camera will be installed for taking images automatically. The collected image set was divided into two data sets: an original data set for training classifiers, and a test set for measuring generalization of the classifiers, and thus the classifiers will not see these test images during training. Both data sets are divided into 14 classes. It became clear during image collection that there would be low number of images of the scarcest bird species, resulting in classes with very low number of data examples. Therefore, in order to be able to classify the scarcest bird species, all the collected images are included with an acceptance that the resulting data set will be imbalanced. The distribution of the number of images for each class is used as an estimate for the actual distribution of bird species in the test area. This is justified by the fact that images are collected in all four seasons and in all hours during day light. The estimate is not necessary reliable in terms of bird species census, because only the species that usually fly at approximately same height with the wind turbines are taken into account, but it is sufficient in the context of this application. The total number of images of the original data set is 24631, and number of images of the test data set is 439. The test data set was created by randomly choosing images from all classes. The number of images in the test data set follows the distribution of the original data set, thus reflecting the actual distribution in the test site. Class labels and number of images of the original data set for each class are presented in Table 1. In this Table, three classes are not defined in species level: LNSP, SWSP, and CATE. The first two cases are because there is no need to distinguish between loon species or swan species any further in this context, regardless the fact that two common, and two rare species of loons occur in the test area, and analogously there occur two common and one rare species of swans. The same applies to the third case too, the common/arctic tern. In addition, it is generally very difficult to tell the difference between these two tern species [16], and thus the number of required data examples (images) might be too large, considering the time needed to collect them. The number of images for each class in the test data set are also given in Table 1.

Table 1. The original data set divided into 14 classes.

# Images	# Test Images	Class Name (Eng.)	Class Name (Lat.)	Class Label
396	7	Loon Species	Gavia sp	LNSP
260	5	Swan Species	Cygnus sp	SWSP
5612	100	Great Cormorant	Phalacrocorax carbo	GRCO
979	17	Common Eider	Somateria mollissima	COEI
1164	21	Common Goldeneye	Bucephala clangula	COGO
236	4	Velvet Scoter	Melanitta fusca	VESC
263	5	Red-breasted Merganser	Mergus serrator	RBME
2450	44	White-tailed Eagle	Haliaeetus albicilla	WTEA
512	9	Great Black-backed Gull	Larus marinus	GBBG
4053	72	Herring Gull	Larus argentatus	HEGU
1481	26	Lesser Black-backed Gull	Larus fuscus fuscus	LBBG
3648	65	Common Gull	Larus canus	COGU
1803	32	Black-headed Gull	Larus ridibundus	BHGU
1774	32	Common/Arctic Tern	Sterna hirundo/paradisaea	CATE

No preprocessing, other than cropping, is applied to the images before feeding them into the classifiers. The cropping is based on a segmentation, and it is motivated by being able to dispose the most of the pixels representing only sky. The resolution of the camera sensor measured by the total number of pixels and the focal length of the lens are important qualities because of the long range, of which images are to be taken. The effective number of pixels (ENP) is defined by the number of pixels representing a bird. The remaining number of pixels are considered noise, thus ENP has a significant effect on the performance of image classification model as birds will be very small (they consist of only a small number of pixels) in the images. ENP depends on the sensor resolution of the camera and the focal length of the lens, and if the sensor resolution is fixed, ENP can be increased by choosing a long (in terms of focal length) telephoto lens. An additional advantage is that there is no need to feed classifiers with large (in terms of the number of pixels) images. For more details about the segmentation, see [9]. Examples of the original data set images are presented in Fig. 2. The first image in this figure illustrates that there can be more than one bird in the image. There are species in the test area that have a custom to fly in tight flocks, and in these cases, the result (in terms of data examples) is an image of several birds. Moreover, there might be more than just one bird species in these flocks. The custom of flying in tight flocks is an important feature in terms of identification for certain bird species [17]. As result of the segmentation, an image has only one bird left when there is a sparse flock of birds in the image. In the sparse flock case, the

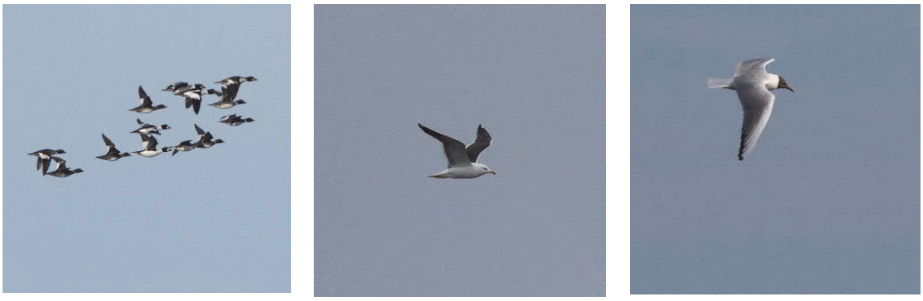


Fig. 2. Data example of the common goldeneye, the black-headed gull and the lesser black-backed gull, respectively.

bird closest to the center of the image is retained, thus when a sparse flock has more than one bird species, the retained bird species is chosen randomly. In the tight flock case, the identification is based on the whole flock, and thus it is biased toward the most numerous bird species in the flock.

4.1 Data Augmentation

Data augmentation is applied to the original data set. We have used our own method, in which the images are converted into various color temperatures according to step size, *s*. The lower and upper limit to the color temperature is 2000 K and 15000 K, respectively. For example, if *s* = 50 (in K), the number of data examples of the augmented data set is (15000 - 2000)/50 + 1 * 24631 = 6453322. For more details, see [9]. In addition to the color conversion, the images are also rotated by a random angle between -20 degree and 20 degree drawn from the uniform distribution. Motivation for this is that CNN is invariant to small translations, but not image rotation [18]. Number of images for each species (classes) in the augmented data set when *s* = 50 and *s* = 200, respectively, are presented in Table 2. Figure 3 presents data examples of the output of the augmentation algorithm. The original image, fed into the data augmentation algorithm, has a color temperature of 5800 K, and the two augmented images have a color temperature of 3800 K and 7800 K, respectively.

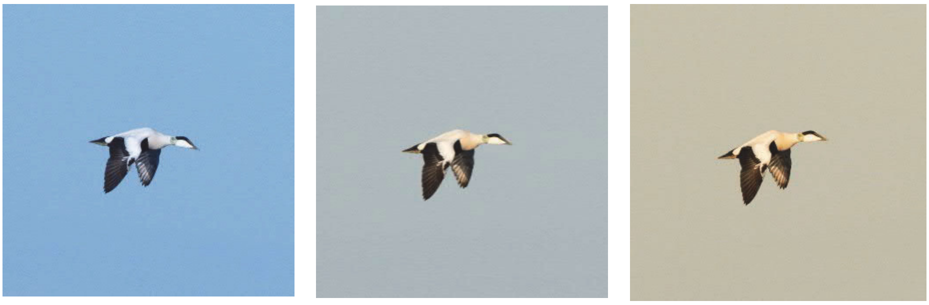


Fig. 3. Data example of the common eider. The image on the left is an augmented image with color temperature of 3800 K. The original image is in the middle with color temperature of 5800 K. The image on the right is an augmented image with color temperature of 7800 K.

Table 2. Number of images for each class in the augmented data set.

Class Label	# Original	# s = 50	# s = 200
WTEA	2450	641900	164150
SWSP	260	68120	17420
LOSP	396	103752	26532
GRCO	5612	1470344	376004
COEI	979	256498	65593
COGO	1164	304968	77988
VESC	236	61832	15812
RBME	263	68906	17621
GBBG	512	134144	34304
HEGU	4053	1061886	271551
LBBG	1481	388022	99227
COGU	3648	955776	244416
BHGU	1803	472386	120801
CATE	1774	464788	118858

4.2 Grouping data

We trained our first classifier models on the entire data set, which was divided into the same number of classes as the data had species. However, there are more and less easily separable classes (assessed by human eye), and that led to an idea of grouping those species together that seem similar to human eye, thus our proposal in this respect is hierarchical [19]. In this approach, the number of classes decreases on the top level of the classifier hierarchy, and thus resulting better separability of the data set. Classification inside the groups is dealt with the subsequent levels in cascades [20]. Figure 4 illustrates examples of clearly and weakly separable classes. The white-tailed eagle and the mute swan are examples of clearly separable classes, and the herring gull and the common gull are examples of weakly separable classes.

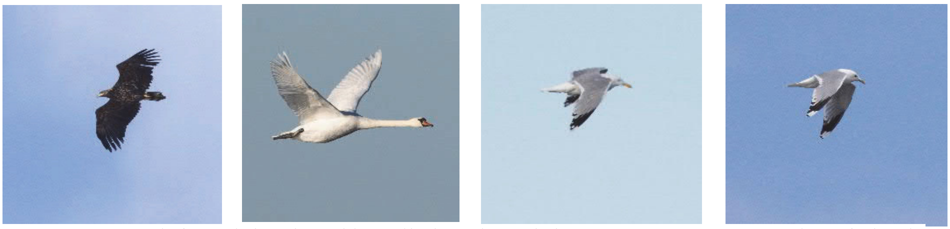


Fig.4. From left to right: the white-tailed eagle and the mute swan are examples of clearly separable classes. The herring gull and the common gull are examples of weakly separable classes.

There are four groups on the top-level of the classification hierarchy. Two of these groups are actually single clearly separable species; *swans* (treated as single species here) and *white-tailed eagle*, respectively. *Gulls-and-terns* and *waterfowl* (including loons and common cormorant) form the other two groups, respectively. More groups are defined below the top level in order to get species level classification. Division of the classes into the groups are given in Table 3. The number of images for each group formed from the original data set and from two augmented data sets with *s* = 50 and *s* = 200, respectively, are given in Table 4.

Table 3. Division of the classes into groups.

Class Label	Top Level	Second Level	Third Level	Fourth Level
WTEA	white-tailed eagle	-	-	-
SWSP	swans	-	-	-
LOSP	waterfowl-1	waterfowl-2	waterfowl-2	-
GRCO	waterfowl-1	great cormorant	-	-
COEI	waterfowl-1	waterfowl-2	waterfowl-2	-
COGO	waterfowl-1	waterfowl-2	waterfowl-2	-
VESC	waterfowl-1	waterfowl-2	waterfowl-2	-
RBME	waterfowl-1	waterfowl-2	waterfowl-2	-
GBBG	gulls-and-terns-1	gulls-and-terns-2	black-backed-gulls	GBBG
HEGU	gulls-and-terns-1	gray-backed-gulls	HEGU	-
LBBG	gulls-and-terns-1	gulls-and-terns-2	black-backed-gulls	LBBG
COGU	gulls-and-terns-1	gray-backed-gulls	COGU	-
BHGU	gulls-and-terns-1	gulls-and-terns-2	blackheaded-tern	BHGU
CATE	gulls-and-terns-1	gulls-and-terns-2	blackheaded-tern	CATE

Table 4. Number of images for each group.

Group	# Original	# s = 50	# s = 200
white-tailed eagle	2450	641900	164150
swans	260	68120	17420
waterfowl-1	8387	2266300	579550
waterfowl-2	2775	795956	203546
gulls-and-terns-1	13271	3477002	889157
gulls-and-terns-2	5570	1459340	373190
gray-backed gulls	7701	2017662	515967
blackheaded-tern	3577	937174	239659
black-backed gulls	1993	522166	118858

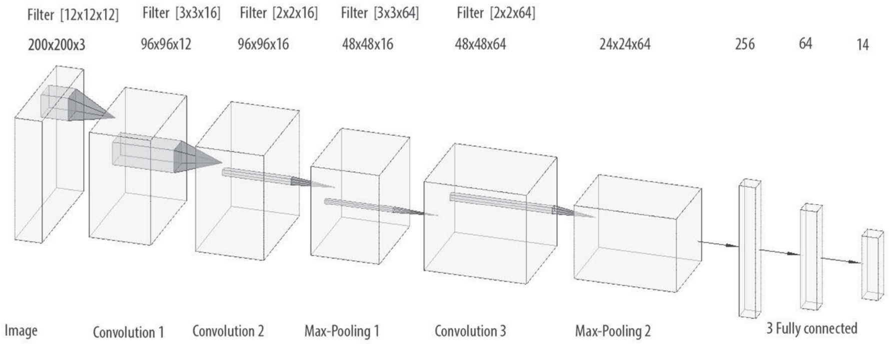


Fig. 5. The CNN model for each classifier.

5 Classification

All classifiers in this application share the same CNN model, which is shown in Fig. 5. Only the number of the neurons in the output changes according to the number of classes. This model has three convolution layers, each of which is followed by a rectified linear unit (ReLU) layer and the first two are followed by a cross-channel normalization layer (Local Response Normalization, LRN). The use of LRN is motivated by its ability to aid the generalization as its function may be seen as brightness normalization [2]. There are two max-pooling layers, the first is before the third convolution layer and the second is before the first fully connected layer. There is no max-pooling layer before the second convolution layer. The reason for this is the small ENP, and thus by omitting a max-pooling layer, all of the finest edges detected by the first convolution layer are transferred to the second convolution layer. The architecture is completed by three fully connected layers. The first two of them are followed by dropout layers, and each dropout layer is followed by ReLUs. The dropout was implemented by randomly setting the output neurons of the layer to zero with a probability of 0.5. The architecture is finally terminated by softmax activation, which produces a distribution over the class labels with cross entropy loss function [21]. The input image is normalized and zero-centered before feeding it to the network. CNN with Mini-batch training and supervised mode as well as stochastic gradient descent with momentum is applied [21,22,23,24]. The L2 Regularization (weight decay) method for reducing over-fitting is also applied [21,24,25]. We kept the network size, in terms of free parameters, small due to limited capacity of computer resources. Thus, resulting in total of 92 feature maps which are extracted by convolution layers with kernel sizes [12 x 12 x 3] x 12, [3 x 3 x 12] x 16 and [3 x 3 x 16] x 64, respectively. Total number of weights is about 9.47 x 10^6.

Images of a size of 200 x 200 pixels are fed to each classifier. In the first convolution layer, this image size produces (200 - 12 + 2 * 1)/2 + 1 = 96 square feature maps, i.e., there are 96 x 96 = 9216 neurons in each feature map. Filter size, number of feature maps, feature map size in neurons, stride, and padding for each convolution layers and max-pool layers are given in Table 5. For each filter, Fig. 5 displays the number of feature maps as the triplet [a, b, c].

Table 5. Parameters for the convolution layers and the max-pooling layers of the CNN model.

Layer	Filter	# Feature Maps	Feature Map Size	Stride	Padding
Convolution 1	12 x 12	12	96 x 96	[2 2]	[1 1]
Convolution 2	3 x 3	16	96 x 96	[1 1]	[1 1]
Max-pooling 1	2 x 2	16	48 x 48	[2 2]	[0 0]
Convolution 3	3 x 3	64	48 x 48	[1 1]	[1 1]
Max-pooling 2	2 x 2	64	24 x 24	[2 2]	[0 0]

5.1 Hyperparameter selection

We split the data set into a training set and a validation set as 70 % and 30 %, respectively. We used manual tuning for choosing the number of epochs. Initial weights for all layers are drawn from the Gaussian distribution with mean 0 and standard deviation 0.01. Initial biases are set to zero. The L2 value is set to 0.0005 and mini-batch size is set to 128.

5.2 Feature extraction

The three convolution layers are designed to detect spatially distributed features from the training images. Usual disjunctive features are shape and general coloration of the bird. The ReLU (to introduce non-linearity) layer and the max-pooling (to increase spatial invariance) layer after the second and third convolution layers, respectively, may be seen as a refinement for the detected features due to the rectifying and down sampling properties of these layers. Figures 6 and 7 illustrate the features, extracted by the CNN, for the classes LBBG and GBBG, respectively. These feature maps are from the second convolution layer. There is one frame for each 16 feature maps in the figure. These images are normalized, so that the minimum weight is 0 and the maximum is 1, i.e., the most negative weight has turned into zero (black). The mid-gray color (0.5) shows those areas in the image that have the minimum contribution to the features, and the most blackish or the most whitish areas denote maximum contribution to the features. The plain gray, or almost so, feature maps indicate that no significant features have been found in these maps. These feature maps show that the CNN is capable to give large weights on those areas of the bird plumage that are relevant for species identification. These areas are mainly: wing tips, feet, and a bill with these two pairs of gull species. As flying gulls usually have their feet concealed by feathers, and their underside is not always visible in the images, the usage of this feature is minor. This leaves us the bill and the wing tip, and because the differences in the bill color and structure are only subtle, the most significant identification point is the wing tip. The great black-backed gull and the lesser black-backed gull also have a slight difference in the hue of their upper wing color, but this does not always seem to result in larger weights produced by the CNN for those areas, at least not large enough, because images of the great black-backed gull are even misclassified as the herring gull. Yet, the upper wing color is the key feature to distinguish between the *gray-backed gulls* and *black-backed gulls* [26].

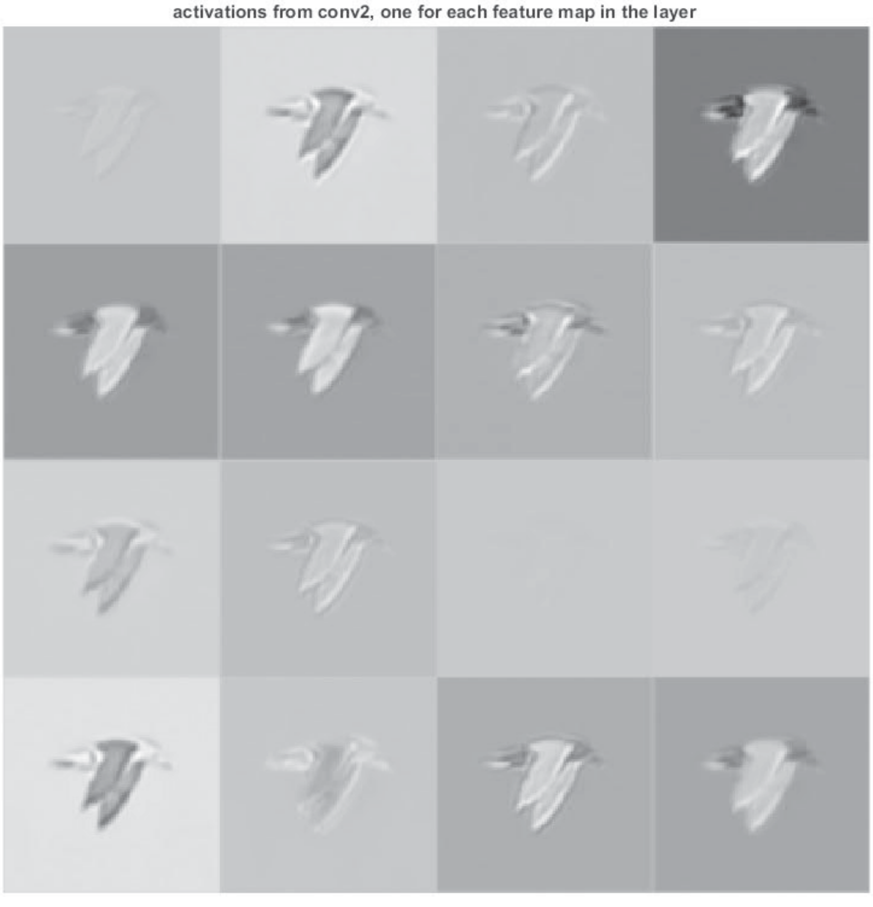


Fig. 6. Visualization of the feature extraction by the CNN model for the class LBBG. There are 16 feature maps in the figure extracted by the second convolution layer.

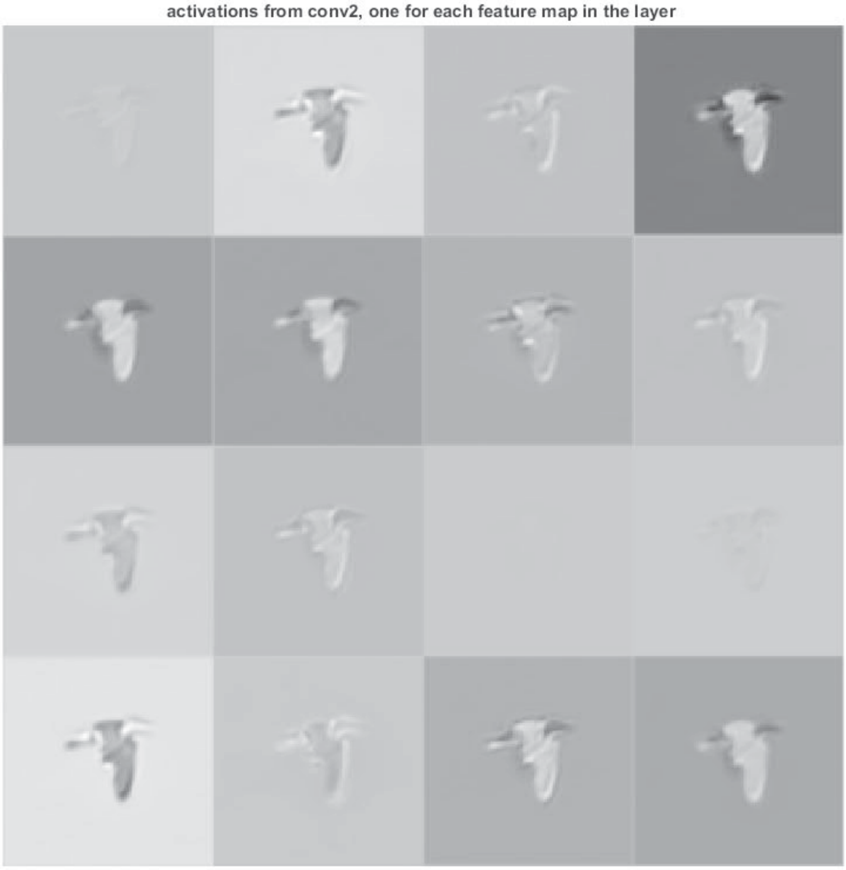

activations from conv2, one for each feature map in the layer

Fig. 7. Visualization of the feature extraction by the CNN model for the class GBBG. There are 16 feature maps in the figure extracted by the second convolution layer.

5.3 Tests for deeper CNN model

It became clear during the development of this algorithm that the challenge, in terms of classification, lies in the group of *gulls-and-terns-1*, especially in the groups of *gray-backed gull* and *black-backed gull*. Considering the CNN model, the first option for a better performance should be a deeper model, i.e., more convolution layers. We modified the original model by adding the fourth convolution layer, followed by ReLU and max-pooling layers. This model had 128 filters with filter size of [5 x 5] in its fourth convolution layer. The first modified model was tested on the group of *black-backed gull*, but it failed to increase the performance of the original CNN model. Then we tested even deeper model by adding the fifth convolution layer, again followed by ReLU and max-pooling layers. In this case, the max-pooling layer before the third convolution layer in the original model was removed in order to have sufficient number of neurons left at the output of the architecture. We also modified the filter sizes of the second modified model. The modified filter sizes are given in Table 6. The two new max-pooling layers at the end of the second modified model have filter size of [2 2], respectively. When this model was tested on the group of *black-backed gull*, the result was the same as the first modified model, i.e., it did not achieve a better performance than the original CNN model in terms of true positive rate (TRP). Both test classifiers were trained on the augmented set, with $s = 50$, of only the images from the group of *black-backed gull*.

Table 6. Filter sizes for the second modified model of the original CNN model.

Layer	Filter	# Feature Maps	Feature Map Size	Stride	Padding
Convolution 1	12 x 12	12	96 x 96	[2 2]	[1 1]
Convolution 2	7 x 7	16	90 x 90	[1 1]	[0 0]
Convolution 3	5 x 5	32	86 x 86	[1 1]	[0 0]
Convolution 4	4 x 4	64	40 x 40	[1 1]	[0 0]
Convolution 5	3 x 3	128	18 x 18	[1 1]	[0 0]

5.4 Dealing with imbalanced data

If we want to identify (classify) all the species that occur in the test area, we must accept that the training data set will be imbalanced, because there will be low numbers of training examples of the scarcest species. However, there are methods that can be used for imbalanced data set. Naturally, the first option would be to collect more data into the training data set, but this is not a very realistic option in our case. Resampling is a method that is easy to implement, and fast to run. This means that copies of data examples are added into the under-represented class, i.e., over-sampling, or data examples are deleted from the over-represented class, i.e., under-sampling [11]. However, we have augmented the original data set (resampling is not used) with $s = 50$, and trained a reference classifier on the augmented data set. The results, in terms of performance, of the hybrid model (hierarchical and cascade model combined) trained on the grouped data set are compared with this reference classifier. The grouped data set is also augmented with $s = 50$, and both data sets are imbalanced. Class imbalance ratios (i.e., ratio of the number of images in a class to the class with the largest number of images) of the original data set for 13 classes, rounded to the nearest integer, are given in Table 7. The class with the largest number of images is GRCO, and it is omitted from the table. It can be seen from the table that there is severe imbalance between several classes and the class GRCO.

Table 7. Imbalanced ratios of 13 classes to the class with the largest number of images (GRCO).

Class Label	Ratio
WTEA	1:2
SWSP	1:22
LOSP	1:14
COEI	1:6
COGO	1:5
VESC	1:24
RBME	1:21
GBBG	1:11
HEGU	1:1
LBBG	1:4
COGU	1:2
BHGU	1:3
CATE	1:3

Another reference classifier is trained on a balanced data set. This data set is created by under-sampling method, so that the original data set is augmented with $s = 50$, and then 236 x 262 = 61832 images are randomly chosen from each class, except for the class VESC, from which all of the images are chosen, because this class has the lowest number of data examples.

It is important to choose a suitable performance metric for classifiers trained on imbalanced data set. We have used confusion matrix as a tool to compare the classifiers. Precision (a measure for classifier exactness) and recall (a measure for classifier completeness, a.k.a. TPR) are metrics that have been calculated from confusion matrices. Receiver operating characteristic (ROC) curves and histograms of predictions are the tools that have been applied to determine thresholds for various classifiers trained on the grouped data set. Histograms present the predictions of a classifier fed by a test data set, which the classifier has never seen before. Thus, histogram shows the distribution of prediction of a classifier over a class by presenting the number of the predicted probabilities that falls into each bin. There are always only two classes in the histograms: the positive class (in red), and the negative class (in blue). If it is necessary to use histograms for more than two classes, then one of the classes is treated as the positive class, and the other classes are combined to form the negative class. For all histograms, the x-axis is probability, and y-axis is the number of hits for each bin. Y-axis ranges from zero to the largest probability that hits a single bin in the histogram. The number of bins is always set to 10, and thus the bin width is 0.1.

5.5 Hybrid of hierarchical and cascade model

In order to improve the performance of the classification algorithm compared to the single CNN classifier, we can use more than just one classifier, and we can divide the data set into suitable groups, which is done in Section 4.2. Eight classifiers have been trained on the grouped data sets. These classifiers form a hierarchy that is applied to classify the original data set. This architecture may also be seen as a hybrid between hierarchical and cascade models. The architecture is depicted in Fig. 8. The level of a classifier in the hierarchy, the data set (the groups) that it has been trained on, and the number of classes for each classifier are given in Table 8.

Table 8. Classifiers trained for the hybrid model.

Classifier	Level	Data Set (the group)	# Classes
1	1	WTEA, SWSP, waterfowl-1, gulls-and-terns-1	4
2	2	cormorant, waterfowl-2	2
3	2	gray-backed-gulls, gulls-and-terns-2	2
4	3	waterfowl-2	5
5	3	blackheaded-tern, black-backed-gulls	2
6	3	HEGU, COGU	2
7	4	BHGU, CATE	2
8	4	LBBG, GBBG	2

The first idea was merely to use cascade model of classifiers, so that the commonest species, determined by the distribution of the data set, would be filtered out (classified) at the first classifier. The second commonest species would be filtered out at the second classifier, and so forth. However, the early tests showed that there is no significant difference in performance, if those classes with better separability would be classified in a single classifier. Moreover, the cascaded approach would have led to a relatively large number of classifiers to be trained on, and thus increasing the training time. Nevertheless, the cascaded model was applied to some groups, and especially to the groups of the weakest separability, which are the gulls-and-terns groups. The prediction of a classifier for a test image is the vector-output of the softmax-layer, and it is given,

$$\boldsymbol{P} = [p_1, p_2, \dots, p_n], \tag{1}$$

where p_i is a probability for a $class_i$ as a result of the classification, and n is the number of classes. Classes are alphabetically ordered by their class labels. Thresholds are applied as follows:

$$c_i = \begin{cases} 1, & \text{if } p_i > threshold_i, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

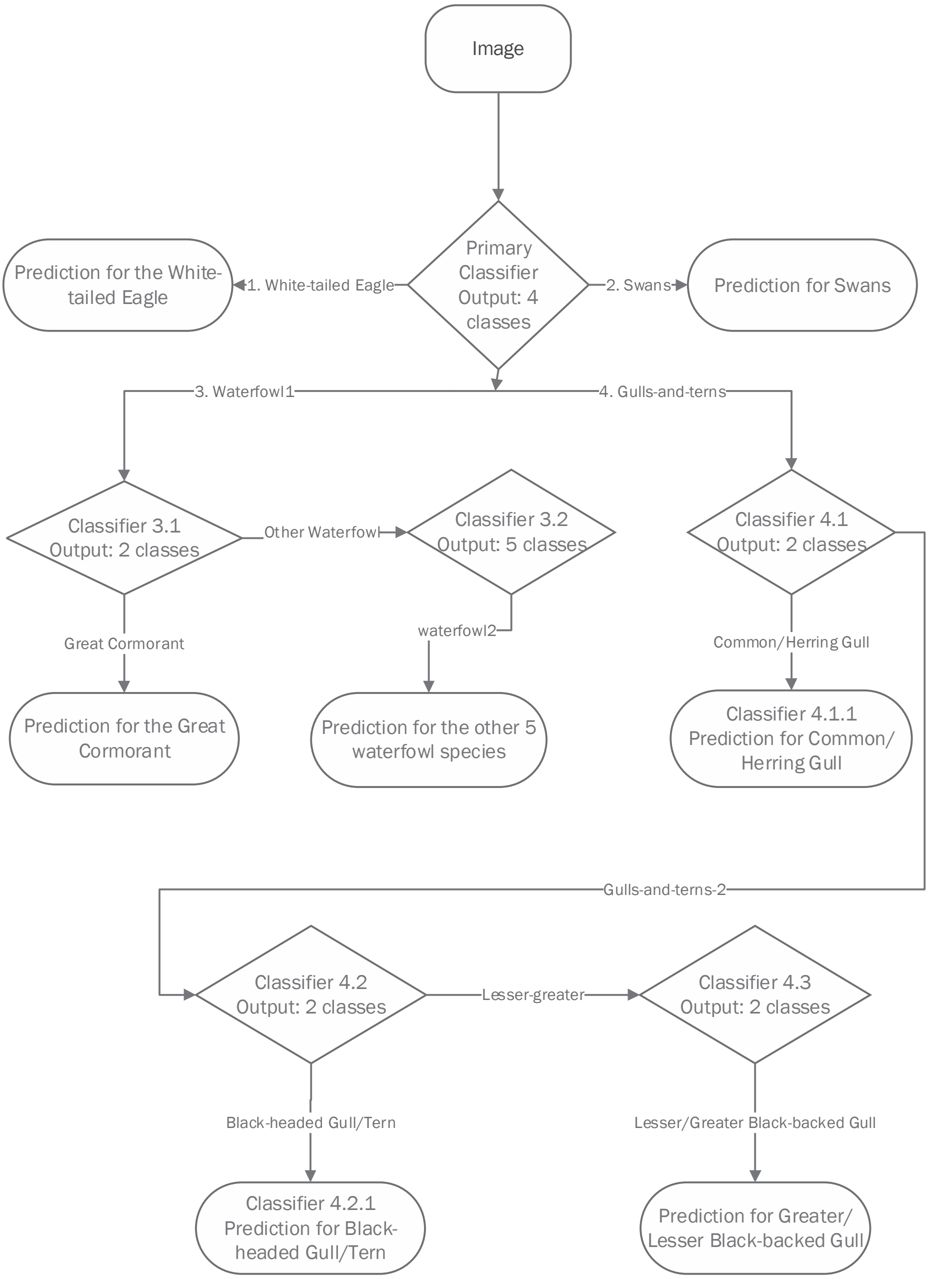


Image
Primary Classifier Output: 4 classes
1. White-tailed Eagle
Prediction for the White-tailed Eagle
2. Swans
Prediction for Swans
3. Waterfowl1
4. Gulls-and-terns
Classifier 3.1 Output: 2 classes
Other Waterfowl
Classifier 3.2 Output: 5 classes
Classifier 4.1 Output: 2 classes
Great Cormorant
waterfowl2
Common/Herring Gull
Prediction for the Great Cormorant
Prediction for the other 5 waterfowl species
Classifier 4.1.1 Prediction for Common/ Herring Gull
Gulls-and-terns-2
Classifier 4.2 Output: 2 classes
Lesser-greater
Classifier 4.3 Output: 2 classes
Black-headed Gull/Tern
Lesser/Greater Black-backed Gull
Classifier 4.2.1 Prediction for Black-headed Gull/Tern
Prediction for Greater/ Lesser Black-backed Gull

Fig. 8. Hierarchy of classifiers.

$$\boldsymbol{C} = [c_1, c_2, \dots, c_n], \tag{3}$$

where p_i is as in the $\boldsymbol{P}$-vector (1), and $threshold_i$ is the threshold for $class_i$. As result of Equations (2) and (3) there will be exactly one element, c_i, turned to one in $\boldsymbol{C}$-vector, and the rest of the elements are turned to zeros. The class label is found according to the index of the element that is turned to one:

$$j = arg\ max_j(\boldsymbol{C}), \tag{4}$$

where j is the index of the predicted class.

5.6 Top-level classification

The top-level classifier is the most important in terms of TPR, because a possible misclassification will recur in subsequent hierarchy. This classifier deals with the groups: *swans*, *waterfowl-1*, *white-tailed eagle*, and *gulls-and-terns-1*. Class imbalance ratios of the top-level groups, rounded to the nearest integer, are given in Table 9. Considering the environmental license requirements, it is crucial to keep the number of false negative (FN, a data example from the positive class that is misclassified as the negative class) of the group of *white-tailed eagle* and the group of *gulls-and-terns-1* as low as possible, preferably at zero. Figure 9 and Fig. 10 illustrate the choice of possible threshold for the group of *white-tailed eagle*, and for the group of *gulls-and-terns-1,* respectively. These histograms are formed from the predictions of the top-level classifier (a.k.a. primary classifier), so that a histogram for the positive class and the negative class are plotted in the same graph, respectively. Equivalent ROC curves are computed based on the histograms, from which the TPRs and false positive rates (FPR) are calculated. ROC curves for the group *white-tailed eagle* and for the group *gulls-and-terns-1* are shown in Fig. 11 and Fig. 12, respectively. Both figures, the histogram and the ROC curve, for the group of *white-tailed eagle* show that this group is clearly separable, and thus it is easy to choose a suitable threshold for perfect classification of the group. Generally, two values of probability can be read from the histogram, and use as a threshold: the lowest probability value of the positive class (LPPC), and the highest probability value of the negative class (HPNC).

Table 9. Imbalanced ratios of the top-level groups to the group with the largest number of images (*gulls-and-terns-1*).

Class Label	Ratio
white-tailed eagle	1:5
swans	1:51
Waterfowl-1	1:2

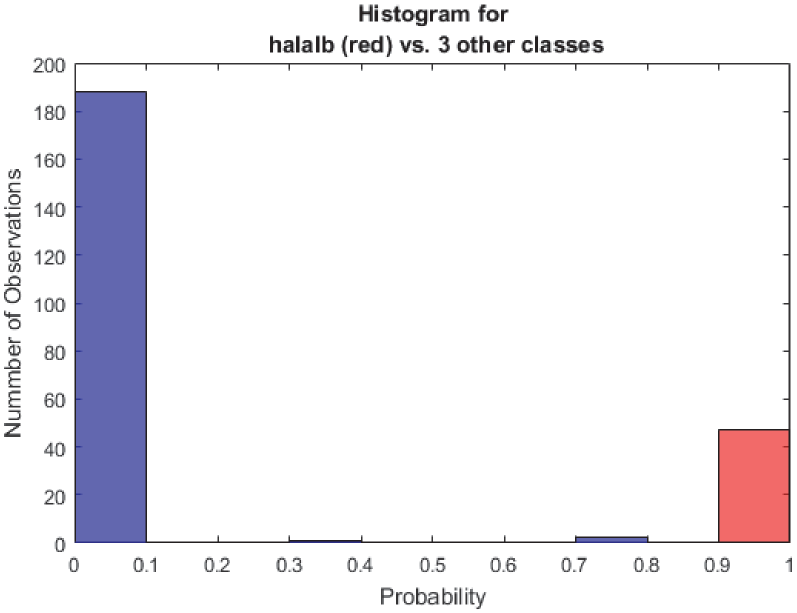


Fig. 9. Histogram for the group *white-tailed eagle*.

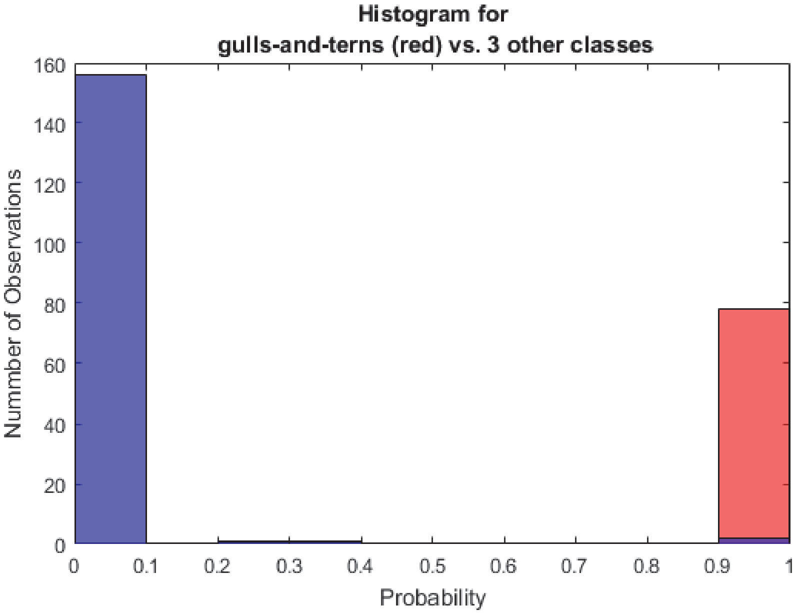


Fig. 10. Histogram for the group *gulls-and-terns-1*.

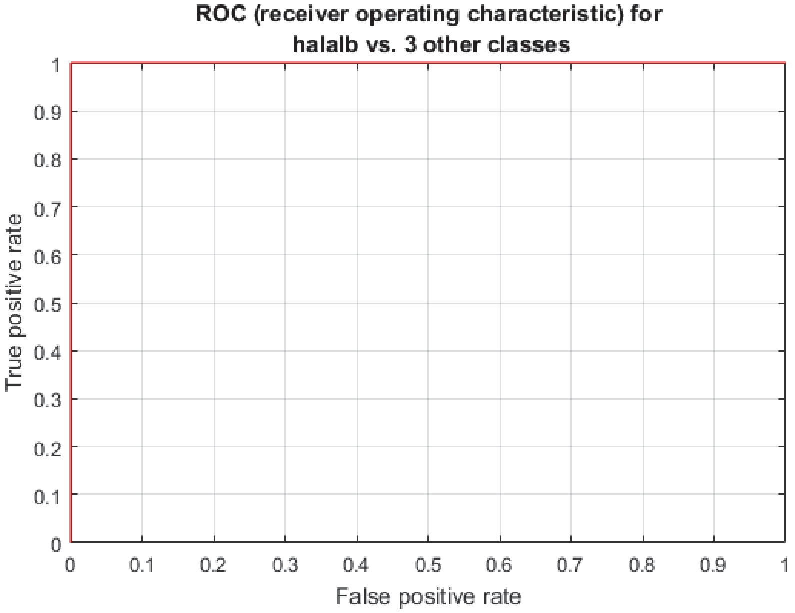


Fig. 11. ROC curve for the group *white-tailed eagle*.

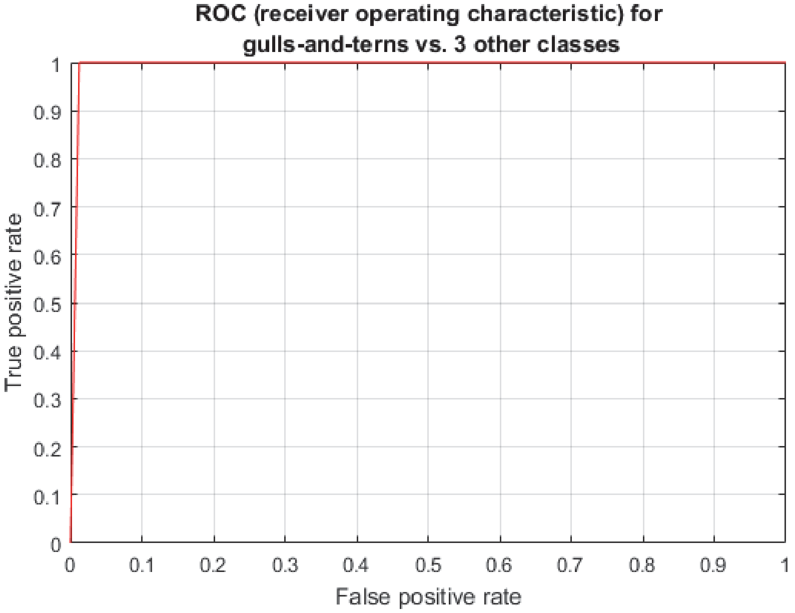


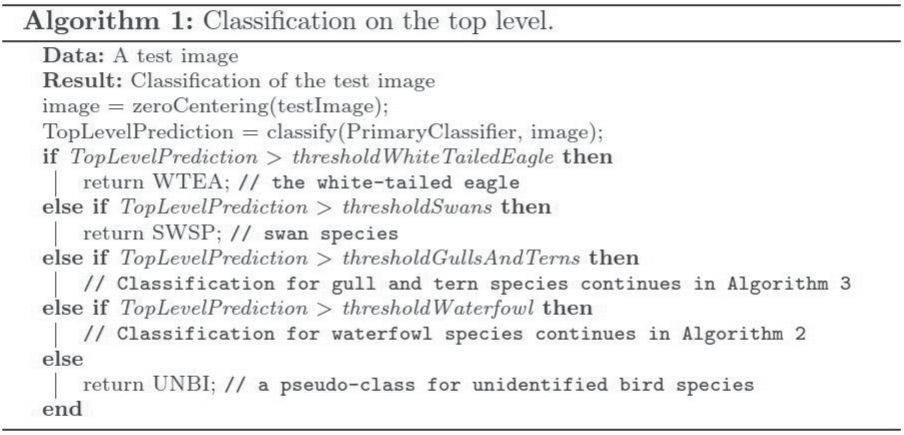
Fig. 12. ROC curve for the group *gulls-and-terns-1*.

In the case of the group of *white-tailed eagle*, these two probability values are not overlapped, and thus this class is clearly separable. The threshold can be set anywhere between 0.8 and 0.9 in order to classify this class perfectly. All true positives (TP, a data example from the positive class that is correctly classified) will be classified correctly and there will be no false positives (FP, a data example from the negative class that is misclassified as the positive class), nor FNs. As the group of *white-tailed eagle* actually consists of only that single bird species, this also means that this classifier is capable to classify the white-tailed eagle in accordance with the environmental license.

In the case of the group of *gulls-and-terns-1*, the LPPC and the HPNC are overlapped. There are two data examples from the negative class that have the probability between 0.9 and 1, and all of the probabilities from the positive class fall into the same bin. Probabilities inside the bins cannot be read from the histograms, but the plotting software (MatLab) also prints the exact values for the probabilities. The LPPC and the HPNC for the group of *gulls-and-terns-1* is 0.9000 and 0.9643, respectively. If we choose 0.9 for the threshold, there will be two FPs, but if we choose 0.9643 for the threshold, there will be no FPs, nor FNs. In this case the number of FNs is the most important, because the lesser black-backed gull belongs to the group of *gulls-and-terns-1*, and it is particularly taken into account in the environmental license, so we cannot take the risk of misclassifying a gull at the top level of the hierarchy. Therefore, we must choose 0.9643 for the threshold. Table 10 shows the applied threshold for the top-level group. The threshold for the group of *white-tailed eagle* is set to 0.7415, because this is the exact value printed by the plotting software. One image of the great cormorant in the test data set is misclassified as the white-tailed eagle, and this causes the number of FPs to be one for the white-tailed eagle and the number of FNs to be one for the group *waterfowl-1*. This is acceptable error rate, because no white-tailed eagles are missed. Algorithm 1 describes the top-level classification process. This algorithm also defines a new pseudo-class, which means that this class does not exist in the data set, but it is used when the primary classifier fails to classify a test image correctly. Thus, it enables a definition of an unidentified bird (UNBI) class without explicitly including it in the real-world classes.

Table 10. Thresholds for the top-level group.

Group	Threshold	# FNs	# FPs
white-tailed eagle	0.7415	0	1
swans	0.7000	0	0
waterfowl-1	0.0083	1	0
gulls-and-terns-1	0.9643	0	0



Algorithm 1: Classification on the top level.

```
Data: A test image
Result: Classification of the test image
image = zeroCentering(testImage);
TopLevelPrediction = classify(PrimaryClassifier, image);
if $TopLevelPrediction > thresholdWhiteTailedEagle$ then
    return WTEA; // the white-tailed eagle
else if $TopLevelPrediction > thresholdSwans$ then
    return SWSP; // swan species
else if $TopLevelPrediction > thresholdGullsAndTerns$ then
    // Classification for gull and tern species continues in Algorithm 3
else if $TopLevelPrediction > thresholdWaterfowl$ then
    // Classification for waterfowl species continues in Algorithm 2
else
    return UNBI; // a pseudo-class for unidentified bird species
end
```

5.7 Classification of waterfowl

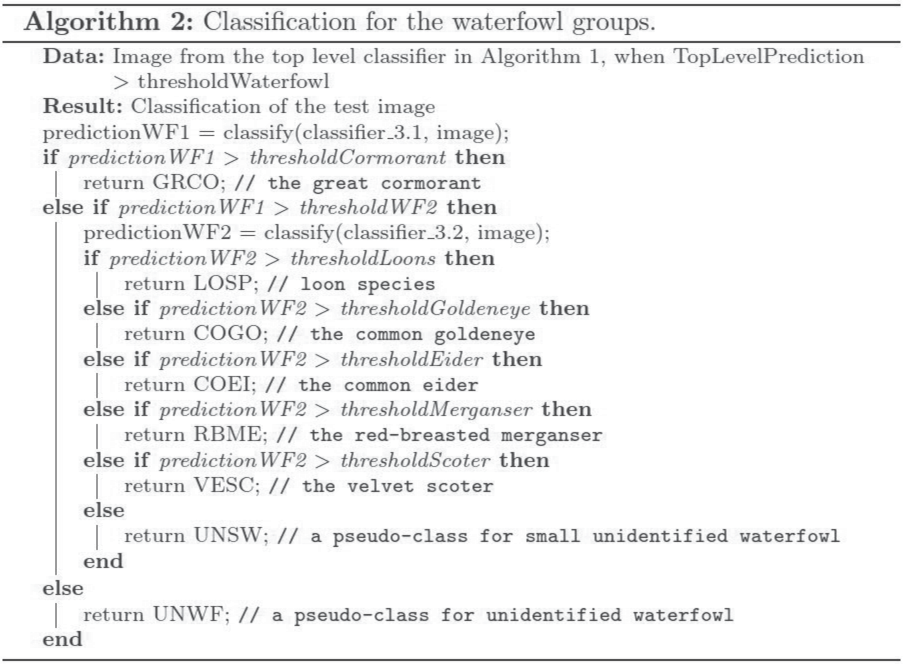
Waterfowl are classified in the second level of the hierarchy, so that two classifiers are cascaded. The first one filters out the commonest class, GRCO, and all the other waterfowl are classified in the second classifier. Thresholds for the first classifier are given in Table 11. There is one FN, and accordingly, one FP as these thresholds are applied. The misclassified class is GRCO, which is the only class in the group of *cormorants*. Thresholds for the group of *waterfowl-2* is given in Table 12. All other classes have one FN, respectively, except for the class LOSP, which is clearly separable. Algorithm 2 shows the classification process for both waterfowl groups. Two new pseudo-classes are defined in the algorithm: unidentified waterfowl (UNWF), and unidentified small waterfowl (UNSW).

Table 11. Thresholds for the group waterfowl-1.

Class Label	Threshold	# FNs	# FPs
cormorant	0.2627	1	0
waterfowl-2	0.7000	0	1

Table 12. Thresholds for the group waterfowl-2.

Class Label	Threshold	# FNs	# FPs
LOSP	0.0402	0	0
COEI	0.9909	1	0
COGO	0.0120	1	0
VESC	0.8810	1	0
RBME	0.9831	1	0



Algorithm 2: Classification for the waterfowl groups.

```
Data: Image from the top level classifier in Algorithm 1, when TopLevelPrediction
      > thresholdWaterfowl
Result: Classification of the test image
predictionWF1 = classify(classifier_3.1, image);
if predictionWF1 > thresholdCormorant then
    return GRCO; // the great cormorant
else if predictionWF1 > thresholdWF2 then
    predictionWF2 = classify(classifier_3.2, image);
    if predictionWF2 > thresholdLoons then
        return LOSP; // loon species
    else if predictionWF2 > thresholdGoldeneye then
        return COGO; // the common goldeneye
    else if predictionWF2 > thresholdEider then
        return COEI; // the common eider
    else if predictionWF2 > thresholdMerganser then
        return RBME; // the red-breasted merganser
    else if predictionWF2 > thresholdScoter then
        return VESC; // the velvet scoter
    else
        return UNSW; // a pseudo-class for small unidentified waterfowl
    end
else
    return UNWF; // a pseudo-class for unidentified waterfowl
end
```

5.8 Classification of gulls and terns

Gulls and terns are classified in the second and third level of the hierarchy in cascade classifiers. We have used a larger test data set with more images for the groups of gulls-and-terns, which is enabled by the fact that the scarcest classes in the original data set are not included in these groups. In this way, we gain more robust threshold, though the original distribution is retained, and the test data set still has only images that the classifiers have never seen before. The number of images in these data set sets are given in Table 13. In this Table, the pair of groups or classes is in the first column from the left, so that the positive class is mentioned first. The following two columns are the number of images of the positive class and the negative class, respectively. We can calculate from the table that the class imbalance ratio for the most of the pairs of the fourth level of the hierarchy (the species level) is almost balanced. The pair {LBBG, GBBG} is the only significant exception having the class imbalanced ratio of 1:3, and because this pair also has the weakest separability, poor result for classification is expected in terms of TPR.

At the second level the commonest group, *gray-backed-gulls* is filtered out first from the group of *gulls-and-terns-2*, and then subsequently the group *blackheaded-tern*. Finally, the group *black-backed-gulls* is the only one left. Figure 13 shows the histogram for the group of *gray-backed-gulls*. It becomes clear from the histogram that the distributions of the positive class (*gray-backed-gulls*) and the negative class

Table 13. Number of images in the larger test data sets for gulls and terns.

Pair of Groups	# Positive Class	# Negative Class
{gray-backed-gulls, gulls-and-terns-2}	174	126
{blackheaded-tern, black-backed-gulls}	192	108
{HEGU, COGO}	92	82
{BHGU, CATE}	97	95
{LBBG, GBBG}	80	28

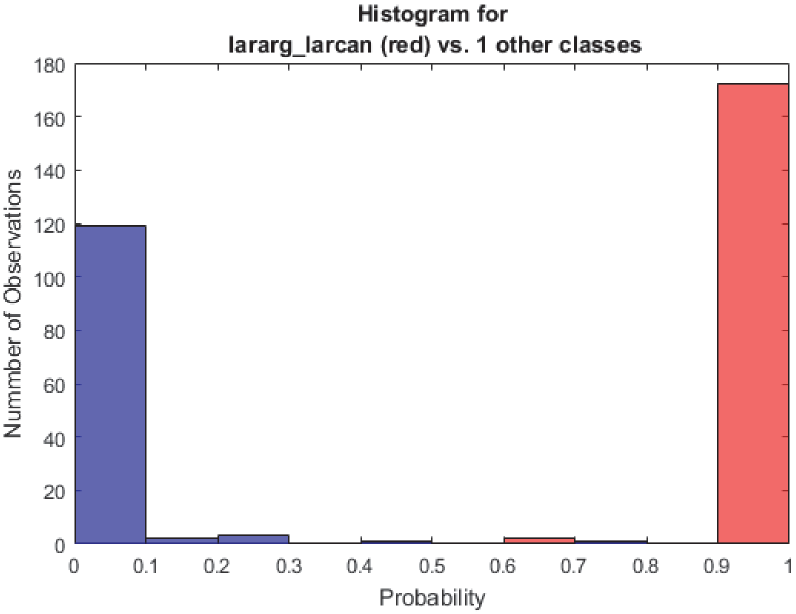


Fig. 13. Histogram for the group *gray-backed-gulls*.

(*gulls-and-tern-2*) are overlapped, and that given we must make a choice for a suitable threshold while keeping in mind the terms of the environmental license. There are two choices for the threshold: 0.6000 with one FP, and 0.7590 with two FNs. We must choose 0.7590 even though it means weaker general performance for the hierarchical classifier. This is because we do not want any member of the class LBBG misclassified on the second level. As result of applying this threshold, there will be two images of the group *gray-backed-gulls* misclassified as *gulls-and-terns-2*.

Species level classification is reached on the fourth (third for gray-backed gulls) level of the hierarchical classifiers. This includes pairs of classes with the weakest separability: {HEGU, COGU}, and {LBBG, GBBG}. The overlap of the distributions of the positive and the negative classes are illustrated in Fig. 14 and Fig. 15. The class

HEGU is the positive class in Fig. 14, and the class COGU is the negative class. The class LBBG is the positive class in Fig. 15, and the class GBBG is the negative class. The best value for a threshold for separating the HEGU from COGU, in terms of classifier performance, is 0.2134. This means zero FP, but eleven FNs, i.e., eleven images of herring gulls will be misclassified as common gulls. Classification of the pair {BHGU, CATE} is straightforward owing to the fact that with the chosen threshold it has the number of FNs and FPs equal to zero. See Table 14 for thresholds for the groups of gulls and terns. The best option for a threshold of the class LBBG is 0.9993 when the number of FNs is 12. Algorithm 3 shows the classification process for the group of *gulls-and-terns-1*. Five new pseudo-classes are defined in this algorithm: gray-backed gull (GBGU, either the herring gull or the common gull), black-headed (BHTE, either the black-headed gull or tern species), black-backed gull (BBGU, either the lesser black-backed of the great black-backed gull), non-gray-backed-gull (NGGU, either BHTE or BBGU), and unidentified gull or tern (UNGU).

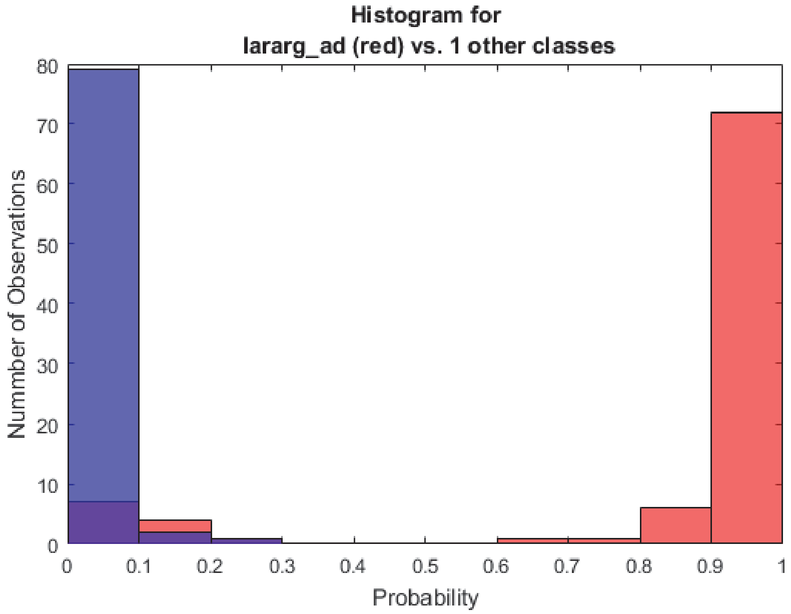


Fig. 14. Histogram for the class HEGU (herring gull).

Table 14. Thresholds applied to the pair of groups of gull and tern species.

Class Label	Threshold	# FNs	# FPs
{gray-backed-gulls, gulls-and-terns-2}	0.7590	2	0
{blackheaded-tern, black-backed-gulls}	0.2524	0	0
{HEGU, COGO}	0.2134	11	0
{BHGU, CATE}	0.8124	0	0
{LBBG, GBBG}	0.9993	12	0

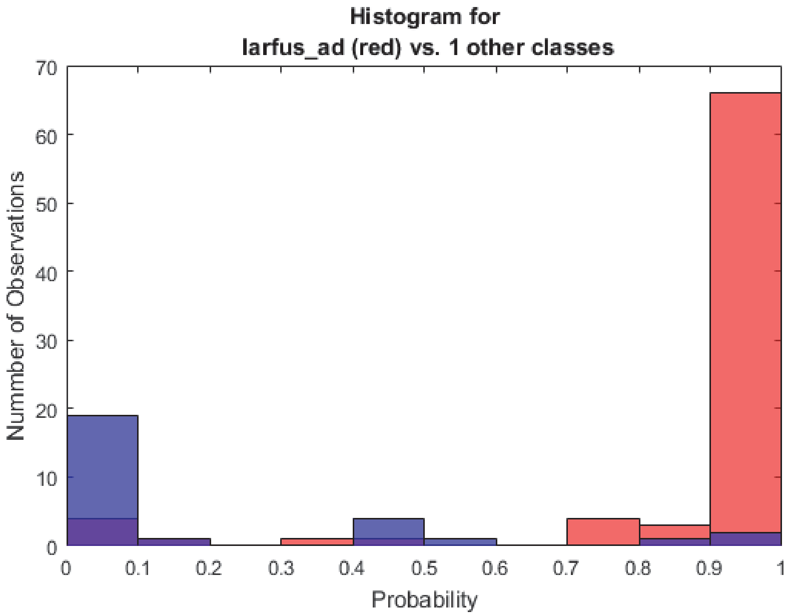
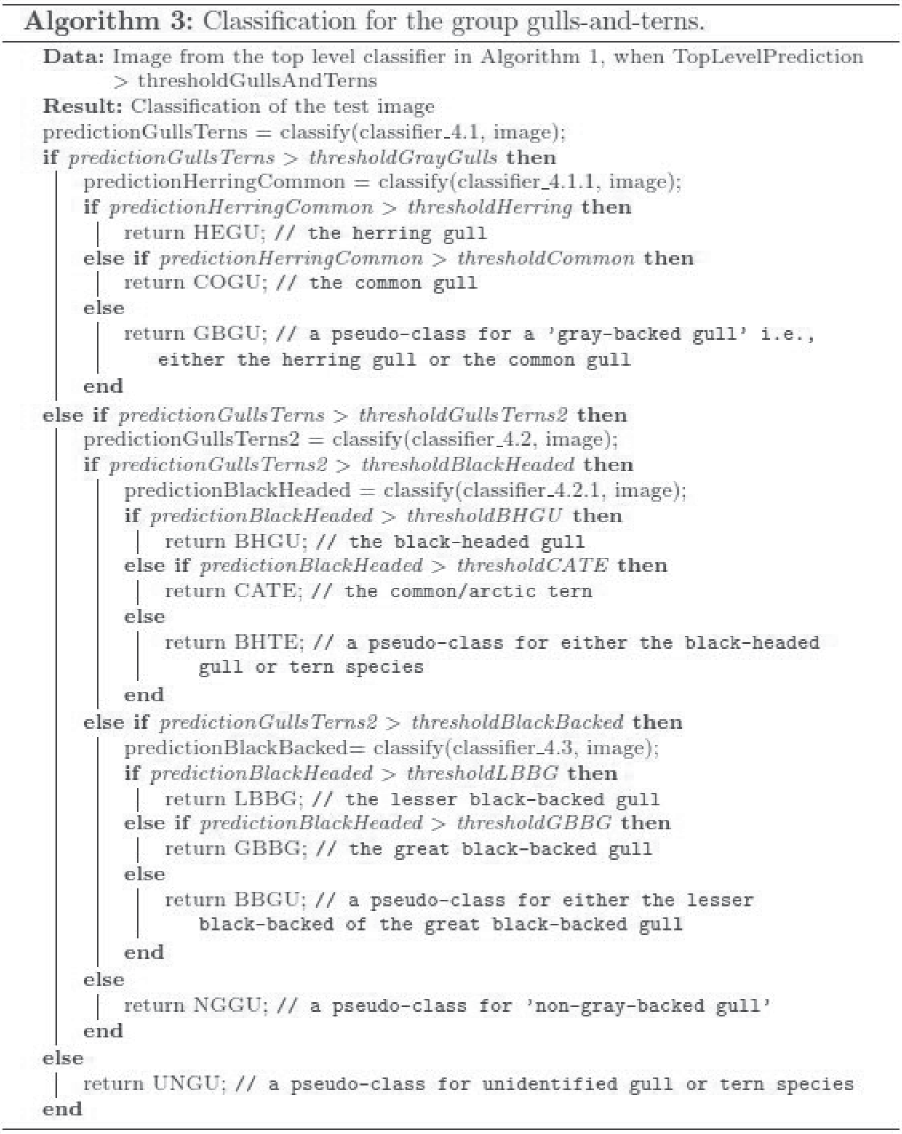


Fig. 15. Histogram for the class LBBG (lesser black-backed gull).

6 Results

Results for comparing the classifiers are viewed through generalization. The hybrid of hierarchical and cascaded model achieved average performance of 0.9460 (TPR). The reference classifiers have average TPRs as follows: for the imbalanced reference classifier (IMBRC), 0.8195 and for the balanced reference classifier (BRC), 0.8307, respectively. The total number of misclassification for the hybrid model was 16. This number for the reference classifiers was 45 for the IMBRC, and 71 for the BRC. Average precision for the hybrid model was 0.9619. Average precision for the IMBRC was 0.8809, and for the BRC 0.7919. TPRs for the top-level groups and the class LBBG are given in Table 15. The reference classifiers were trained on ungrouped classes, therefore the numbers for TPRs of the groups have been averaged of the numbers of those classes that the groups consist of.

Confusion matrix for the top-level groups is given in Table 16. This confusion matrix also includes the pseudo-class UNBI. Naturally, the number of TPs are zero for pseudo-classes, because these classes are only defined for failure of the classifiers. Confusion matrix for the classes are given in two parts, because it is too big to fit in the page. Table 17 presents the first part of the confusion matrix including the group of *waterfowl-1*, i.e., the classes: LOSP, GRCO, COEI, COGO, VESC, and RBME. This confusion matrix also includes the pseudo-classes: UNSW, and UNWF. One test image of GRCO is presented in the top-level confusion matrix, therefore the number of test images for the class GRCO is 99



Algorithm 3: Classification for the group gulls-and-terns.

```
Data: Image from the top level classifier in Algorithm 1, when TopLevelPrediction
      > thresholdGullsAndTerns
Result: Classification of the test image
predictionGullsTerns = classify(classifier_4.1, image);
if predictionGullsTerns > thresholdGrayGulls then
    predictionHerringCommon = classify(classifier_4.1.1, image);
    if predictionHerringCommon > thresholdHerring then
        return HEGU; // the herring gull
    else if predictionHerringCommon > thresholdCommon then
        return COGU; // the common gull
    else
        return GBGU; // a pseudo-class for a 'gray-backed gull' i.e.,
            either the herring gull or the common gull
    end
else if predictionGullsTerns > thresholdGullsTerns2 then
    predictionGullsTerns2 = classify(classifier_4.2, image);
    if predictionGullsTerns2 > thresholdBlackHeaded then
        predictionBlackHeaded = classify(classifier_4.2.1, image);
        if predictionBlackHeaded > thresholdBHGU then
            return BHGU; // the black-headed gull
        else if predictionBlackHeaded > thresholdCATE then
            return CATE; // the common/arctic tern
        else
            return BHTE; // a pseudo-class for either the black-headed
                gull or tern species
        end
    else if predictionGullsTerns2 > thresholdBlackBacked then
        predictionBlackBacked= classify(classifier_4.3, image);
        if predictionBlackHeaded > thresholdLBBG then
            return LBBG; // the lesser black-backed gull
        else if predictionBlackHeaded > thresholdGBBG then
            return GBBG; // the great black-backed gull
        else
            return BBGU; // a pseudo-class for either the lesser
                black-backed of the great black-backed gull
        end
    else
        return NGGU; // a pseudo-class for 'non-gray-backed gull'
    end
else
    return UNGU; // a pseudo-class for unidentified gull or tern species
end
```

in the waterfowl confusion matrix. Table 18 presents the second part of the confusion matrix including the group *gulls-and-terns-1* (the classes: GBBG, HEGU, LBBG, COGU, BHGU, and CATE). The five pseudo-classes defined in Algorithm 3 for gulls and terns are omitted in order to save space, and because no image of any of the subgroups of the *gulls-and-terns-1* was misclassified as any of these pseudo-classes.

Table 15. TPRs for the hybrid model and the reference classifiers.

Classifier	WTEA	SWSP	waterfowl-1	gulls-and-terns-1	LBBG
Hybrid	1	1	0.9935	1	0.9231
Imbalanced reference	0.9773	0.4000	0.7629	0.7691	0.6923
Balanced reference	1	0.8000	0.8621	0.7762	0.8846

Table 16. Confusion matrix for the top-level groups of the hierarchy.

	WTEA	SWSP	waterfowl-1	gulls-and-tern-1	UNBI
WTEA	44	0	0	0	0
SWSP	0	5	0	0	0
waterfowl-1	1	0	153	0	0
gulls-and-terns-1	0	0	0	236	0
UNBI	0	0	0	0	0

Table 17. Confusion matrix for the classes of the group *waterfowl-1*.

	LOSP	GRCO	COEI	COGO	VESC	RBME	UNSW	UNWF
LOSP	7	0	0	0	0	0	0	0
GRCO	0	98	0	1	0	0	0	0
COEI	0	0	16	0	0	0	1	0
COGO	0	0	0	19	0	1	1	0
VESC	0	0	0	0	3	0	1	0
RBME	0	0	0	0	0	4	1	0
UNSW	0	0	0	0	0	0	0	0
UNWF	0	0	0	0	0	0	0	0

Table 18. Confusion matrix for the classes of the group *gulls-and-terns-1*.

	GBBG	HEGU	LBBG	COGU	BHGU	CATE
GBBG	4	1	4	0	0	0
HEGU	0	66	0	6	0	0
LBBG	0	0	26	0	0	0
COGU	0	1	0	64	0	0
BHGU	0	0	0	0	32	0
CATE	0	0	0	1	0	31

As the number of images in the test data set is 439, we must split this number between the three confusion matrices. The first confusion matrix is for all 439 images, but because the classes WTEG and SWSP are only presented in this confusion matrix, the sum of the number of test images in the other two confusion matrices is 439 - 50 = 389. The second confusion matrix presents results for 153 test images and the third for 236 test images, so that the sum is 153 + 236 = 389 test images.

Confusion matrix for the IMBRC is given in two tables: Table 19, and Table 20, respectively. The class WTEA is included in both tables, because there are FPs and/or FNs for it in the two tables. However, the number of TPs for the class WTEA is only given in the first table. The same test data set, as with the hybrid model, has been used when the reference classifiers were tested. The total number of images, 439, is again divided into two tables. The first table covers 197 test images and the second table covers 242 test images.

Confusion matrix for the BRC is also given in two tables: Table 21, and Table 22, respectively. There are no FPs or FNs for the class of WTEA in the second confusion matrix, so the class can be omitted from this table.

Table 19. Confusion matrix for the imbalanced reference classifier, part one.

	WTEA	LOSP	GRCO	COEI	COGO	VESC	RBME
WTEA	43	0	0	0	0	0	0
LOSP	0	6	0	0	1	0	0
GRCO	2	0	97	0	1	0	0
COEI	0	0	0	17	0	0	0
COGO	1	0	0	1	19	0	0
VESC	0	0	0	1	0	3	0
RBME	0	0	0	0	0	0	5

The results for the modified CNN models compared to the original CNN model are given in Table 23. All three models were trained on the same augmented data set, which only consisted of the images from the group of *black-backed gull*. These models were tested as single classifiers. There are TPRs for both training and generalization (tested on images that the classifier has never seen before) in the Table. The first modified model had four convolution layers, and the second had five convolution layers. The models were tested only on the group of *black-backed gull* in these tests.

Table 20. Confusion matrix for the imbalanced reference classifier, part two.

	WTEA	SWSP	GBBG	HEGU	LBBG	COGU	BHGU	CATE
WTEA	-	0	0	0	0	1	0	0
SWSP	0	2	0	2	0	1	0	0
GBBG	1	0	3	2	2	1	0	0
HEGU	1	0	0	67	0	4	0	0
LBBG	0	0	4	2	18	1	1	0
COGU	0	1	0	8	0	56	0	0
BHGU	0	0	0	0	1	0	30	1
CATE	0	0	0	1	0	1	3	27

Table 21. Confusion matrix for the balanced reference classifier, part one.

	WTEA	LOSP	GRCO	COEI	COGO	VESC	RBME
WTEA	44	0	0	0	0	0	0
LOSP	0	6	0	0	1	0	0
GRCO	4	1	91	1	1	2	0
COEI	0	0	0	16	0	1	0
COGO	0	0	2	1	15	2	1
VESC	0	0	0	1	0	3	0
RBME	0	0	0	0	0	0	5

Table 22. Confusion matrix for the balanced reference classifier, part two.

	SWSP	GBBG	HEGU	LBBG	COGU	BHGU	CATE
SWSP	4	0	0	0	1	0	0
GBBG	0	5	0	4	0	0	0
HEGU	1	0	48	4	18	0	1
LBBG	0	2	0	23	0	1	0
COGU	0	1	10	1	52	1	0
BHGU	0	0	0	1	2	28	1
CATE	0	0	1	0	3	0	28

Table 23. TPRs for the modified CNN models compared to the original CNN model.

Model	Training	Generalization
Modified 1	0.9977	0.8384
Modified 2	0.9827	0.7464
Original	0.9989	0.8597

7 Discussion

The tests showed that the hybrid model is significantly better, in terms of performance, than the reference classifiers. The only problematic class, in terms of the environmental license, is the LBBG. Even though it had the number of FNs zero in the test for the hybrid model, the number of FNs was 12 in the test for the gulls and terns only. The number of test images in the latter test was larger, and this gives insight into real-world implementation. The number of possible FPs is not significant in this context, because it would just mean that other gull species, more likely great black-backed gulls, are misclassified as LBBG. Therefore, it is advisable to combine the classes LBBG and GBBG into a single class, i.e., not classify the group *black-backed-gulls* any further.

The BRC performed better than the IMBRC, in terms of TPR. However, the number of misclassification is 71 for the BRC and 45 for the IMBRC. The difference is explained by the better average precision of the IMBRC. Precision increases as the number of FPs decreases, and TPR increases as the number of FNs decreases. This means that TPR is more significant metric than precision in our context, and thus the BRC would be the second choice after the hybrid model. The IMBRC showed poor performance even though it was trained on larger data set ($6.45 * 10^6$ versus $8.66 * 10^5$) than the BRC. This implies that straightforward use of a single classifier on an imbalanced data set gives poor performance in terms of TPR. This result is based on relatively low number of data examples, which is often the case in real-world application, but this method could perform better when trained on significantly larger training data set. However, if precision is an important criterion, then this method may be considered for real-world usage.

The top-level group has the number of FPs equal to one in its confusion matrix (Table 16). This FP is a misclassified GRCO as WTEA. This is, of course, a FN for the class GRCO. However, this is acceptable as no WTEA is misclassified, and thus the number of FNs for the class WTEA is zero. The group *waterfowl-1* also shows good results, and there are only five misclassification. It seems that grouping the original classes is useful approach to this kind of real-world classification problem. By grouping, you can confine the most difficult classification problem to the one group or even just to one subgroup. This approach indicates where the challenge lies. In this context the challenge are the groups of *gray-backed-gulls* and *black-backed-gulls*, respectively. The bird species that these groups consist of are very similar in terms of morphology. This leads to a conclusion (assessed by human eye) that the overlapped area of the classification boundary is clearly wide for both groups. This suggests that significant increase of classification performance can be achieved only by collecting more images of these groups.

Surprisingly, the modified CNN models with architecture of more than three convolution layers, did not perform better than the original CNN model. This implies that the original model, with the architecture of three convolution layers, is capable to extract all relevant features from the training images, and additional convolution layers cannot adduce any more information.

The measured performance of the image classification has been obtained without using the parameters supplied by the radar. Those parameters, especially the speed of an object, provide additional and relevant a-priori knowledge to the system. It is measured by the radar system that there are significant differences in flight speed between the groups of *waterfowl-1* and *gulls-and-terns-1*, and this can be utilized to turn a misclassification into the correct one.

References

1. Li, F., Li, S., Zhu, C., Lan, X., Chang, H.: Class-imbalance aware CNN extension for high resolution aerial image based vehicle localization and categorization. In : 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu (2017)
2. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM, 84-90 (2017)
3. Mao, R., Lin, Q., Allebach, J.: Robust Convolutional Neural Network Cascade for Facial Landmark Localization Exploiting Training Data Augmentation. In : Imaging and Multimedia Analytics in a Web and Mobile World 2018, pp.374-1-374-5(5) (2018)
4. Jia, S., Wang, P., Jia, P., Hu, S.: Research on data augmentation for image classification based on convolution neural networks. In : 2017 Chinese Automation Congress (CAC), Jinan (2017)
5. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A Convolutional Neural Network Cascade for Face Detection. In : 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston (2015)
6. Rachmadi, R., Uchimura, K., Koutagi, G., Komokata, Y.: Japan road sign classification using cascade convolutional neural network. In : ITS (Intelligent Transport System) World Gongress, Tokyo, pp.1-12 (2016)
7. Robin radar models. In: Robin Radar Systems B.V. (Accessed 2019) Available at: https://www.robinradar.com/
8. pelco-D protocol. In: Bruxy REGNET. (Accessed 2019) Available at:

http://bruxy.regnet.cz/programming/rs485/pelco-d.pdf

9. Niemi, J., Tanttu, J.: Automatic Bird Identification for Offshore Wind Farms. In Bispo, R., Bernardino, J., , C., Costa, J. L., eds. : Wind Energy and Wildlife Impacts, Cham, pp.135-151 (2019)

10. Mirzaei, G., Jamali, M., Ross, J., Gorsevski, P., Bingman, V.: Data Fusion of Acoustics, Infrared, and Marine Radar for Avian Study. IEEE Sensors Journal 15(11) (2016)

11. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 20-29 (2004)

12. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory 904, 23-37 (1995)

13. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In : Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, pp.511-518 (2001)

14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In : 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, pp.886-893 (2005)

15. Yoshihashi, R., Kawakami, R., Iida, M., Naemura, T.: Evaluation of Bird Detection using Time-lapse Images around a Wind Farm. Wind Energy 20 (12), 1983-1995 (2017)

16. Malling Olsen, K., Larsson, H.: Terns of Europe and North America. Helm, London (1995)

17. Madge, S., Burn, H.: Wildfowl, an identification guide to the ducks, geese and swans of the world. Helm, London (1988)

18. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition. In : International Conference on Computer Vision, Kyoto, pp.2146-2153 (2009)

19. Silla, C., Freitas, A.: A survey of hierarchical classification across different application domains. Data Min Knowl Disc 22(31) (2011)

20. Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp.3476-3483 (2013)

21. Bishop, C. M.: Pattern Recognition and Machine Learning. Springer, New York (2006)

22. Y., L., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In : Proceedings of the IEEE 86, 11, New York, pp.2278-2324 (1998)

23. Li, M., Zhang, T., Chen, Y., Smola, A. J.: Efficient Mini-batch Training for Stochastic Optimization. In : Proceedings of the 20th ACM SIGKDD international conference on Knowledge, New York, pp.661-670 (2014)

24. Murphy, K. P.: Machine Learning: A Probabilistic Perspective. The MIT Press,

Cambridge, USA (2012)

25. Haykin, S.: Neural Networks: A Comprehensive Foundation 2nd edn. Prentice Hall/Pearson, New York (1994)

26. Malling Olsen, K., Larsson, H.: Gulls of Europe, Asia and North America. Helm, London (2003)