DM

Usage of Convolutional Neural Networks for Identifying Marine Mammal Individuals

MASTER DISSERTATION

Jorge Miguel Vieira Gouveia
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

December | 2022

Usage of Convolutional Neural Networks for Identifying Marine Mammal Individuals

MASTER DISSERTATION

Jorge Miguel Vieira Gouveia
MASTER IN INFORMATICS ENGINEERING

ORIENTATION
Marko Radeta

FCEE

Mestrado em Engenharia Informática

Usage of Convolutional Neural Networks for Identifying Marine Mammal Individuals

Jorge Gouveia

supervisionado por

Prof. Dr. Marko Radeta

9 de março de 2023

Abstract

Identifying marine mammals is a common practice performed by whale-watching crew members. Typically, an experienced marine ecologist is the one who can identify not just the taxa, but also the individual. This process is however always done in the aftermath of data sampling, where the goal is to use photo identification and match the dorsal fins of individuals spotted at the different spatio-temporal scales. This dissertation provides the pipeline and addresses the challenges in the usage of Convolutional Neural Networks (CNNs) to discriminate marine mammal individuals, in this case (pilot whales) based on the dorsal fins. The dissertation uses as input the 1138 images dataset containing over 856 individuals, and through three experiments addresses the issues when discriminating such a high number of classes. In the first experiment, the dissertation studies the role of synthetic data augmentation in boosting model performance. In second, the dissertation benchmarks the existing state-of-the-art convolutional neural network architectures. In third, the dissertation focuses on discriminating other features from dorsal fins to identify individuals (scratches, nicks, roundness, wideness). The dissertation outlines the issues and proposes the guidelines for the next effort in discriminating marine mammal individuals.

Keywords: Convolutional Neural Networks (CNNs) · Deep Learning (DL) · Marine Mammals · Photo Identification · Object Detection · Image classification

Resumo

A identificação de mamíferos marinhos é uma prática comum realizada pelos tripulantes de observação de baleias. Normalmente, o ecologista marinho experiente é aquele que pode identificar não apenas os táxons, mas também o indivíduo. Este processo, é no entanto feito sempre após a amostragem de dados, onde o objetivo é usar a identificação por foto e combinar as barbatanas dorsais dos indivíduos localizados nas diferentes escalas espaço-temporais. Esta dissertação fornece o pipeline e aborda os desafios do uso de Redes Neurais Convolucionais (CNNs) para discriminar indivíduos de mamíferos marinhos, neste caso (baleias-piloto) com base nas barbatanas dorsais. A dissertação usa como input um dataset de 1138 imagens que contêm 856 indivíduos, e através de três experiências aborda os problemas de discriminar um número tão elevado de classes. Na primeira experiência, a dissertação estuda o papel do aumento de dados sintéticos no melhoramento do desempenho do modelo. Na segunda experiência, a dissertação avalia arquiteturas de redes neurais convolucionais de última geração existentes. Na terceira experiência, a dissertação foca-se em discrimar outras caraterísticas das barbatanas dorsais para identificar indivíduos (arranhões, cortes, redondeza, amplitude). A dissertação descreve os problemas e propõe as diretrizes para o próximo esforço em discriminar indivíduos de mamíferos marinhos.

Keywords: Redes Neuronais Convolucionais (CNNs) · Deep Learning (DL) · Mamíferos marinhos · Identificação por Fotos · Deteção de Objectos · Classificação de Imagens

Acknowledgements

After six years my academic journey reaches its end. During these years I have met a lot of important people that I thank for being part of this journey and for helping me when needed.

I would like to thank my parents, my sister, and my grandmother for always supporting, encouraging, and believing in me, thank you for always being there for me in the good and in the bad moments.

Of course, would also like to thank all my friends for the good moments and for turning this journey more fun and joyful.

And would also like to thank all the professors who were part of my journey at this university but most importantly my supervisor, Prof. Dr. Marko Radeta for his dedication, advice, availability, support, commitment, and knowledge-sharing that made the conclusion of this dissertation possible.

This dissertation is part of the following projects: **INTERWHALE** - Advancing Interactive Technology for Responsible Whale-Watching, with grant no. PTDC/CCI-COM/0450/2020 by FCT; **LARGESCALE** - Location-based Augmented Reality Gadgets and Environment-friendly Sightseeing of Cultural Attractions for Locals and Excursionists, with grant no. PTDC/CCI-CIF/32474/2017 by FCT and Portuguese National Funds (PIDDAC); **INTERTAGUA** - Interfaces Aquáticas Interativas para Deteção e Visualização da Megafauna Marinha Atlântica e Embarcacoes na Macaronésia usando Marcadores Rádio-transmissores, with grant no. MAC2/1.1.a /385 by Programa de Cooperación INTERREG V-A España-Portugal MAC (Madeira-Azores-Canarias) 2014-2020; This study had the support of FCT through the strategic project UIDB/04292/2020 awarded to MARE and through project LA/P/0069/2020 granted to the Associate Laboratory ARNET. Project has been conducted with the support of the Wave Labs[1] - interactive technologies for depicting the anthropogenic impact on the marine biosphere.

[1] `https://wave-labs.org`

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Importance of whale-watching

Whale-watching is an activity that consists of encountering cetaceans in their habitat for a series of purposes. The tours can last from 1 hour to 2 weeks and the use of cruise ships or kayaks is needed to perform this activity but can also be executed from land points or seaplanes in the air or even in diving activities [1]. Nowadays, whale-watching is a tourism activity that happens in a lot of countries and manages to have a much more positive and sustainable impact on an economic level than the killing of whales to obtain certain products due to the jobs that whale-watching will create without having to harm any species [2]. Studies suggest that whale-watching could generate about 2.5 billion dollars in yearly revenue and about 19000 jobs around the world [3]. In addition, through whale-watching, it is possible to analyze the behavior of whales as well as their movements in the different seasons. It is also possible to obtain the number of individuals of a certain population in a certain area through photos taken during the activity, such as the images that were used in the experiments of this dissertation. This last one is very important because only the number of members of a certain population can help us know whether the species in question is endangered or not.

1.2 Problem statement

Nowadays, it is possible to identify marine mammal species through deep learning, however, the correct identification of individuals remains a great challenge due to the limitations of datasets of individuals for models to predict the features [4]. Marine mammal individuals can not be detected from the sea vessels but can be detected in the aftermath of the field trip. On such occasions, marine ecologists typically invest a lot of time in correlating the gathered photographs of the species with the known individuals from their local databases [5]. Such process is performed mostly with the manual inspection, where two images are compared visually by the marine ecologist expert [6]. This dissertation thus addresses this challenge in an effort to identify marine mammal individuals, benchmarking the state-of-the-art Convolutional Neural Networks (CNNs).

1.3 Pitfalls of current approaches

Current techniques use edge detection of dorsal fins, however, such approaches remain laborious and time-consuming [7]. In addition, these techniques have some limitations. For instance, an Artificial Intelligence (AI) model trained with a lower amount of input imagery of individuals as training images may fail to learn the features correctly [8]. Such imagery thus needs to be subject to further data augmentation. Lighting conditions and angles of the photographs may be an issue, where images with different luminosity levels make the image classification process more challenging [9]. The scale and distance to the species may also be a problem, where dorsal fins of different scales or with greater distance make the feature recognition more difficult for the model to recognize. Thus such an approach may not be used for the photographs obtained by amateurs.

1.4 Proposed solution and evaluation

To address these challenges, the dissertation implements the pipeline based on Convolutional Neural Network (CNN), comparing state-of-the-art models to correctly classify marine individual

specimens. The dissertation uses as an input the MARE dorsal fin dataset from the Whale Insular Team[2] with 1138 images and 856 individual specimens, based on training the models in identifying the dorsal fins features.

1.5 Contribution

The dissertation provides a threefold novelty: (i) it benchmarks the aforementioned architectures, and describes which is the appropriate architecture to discriminate pilot whales based on their dorsal fins; (ii) it provides the customizable deep learning pipeline for performing image classification; and (iii) provides a survey of existing approaches (including the AI) when identifying marine mammals.

1.6 Glossary of Used Acronyms

Below, the dissertation presents some of the commonly used acronyms throughout the document, facilitating the reference to the reader.

GPU Graphics Processing Unit

CNN Convolutional Neural Network

MARE Centro de Ciências do Mar e do Ambiente

GB Gigabyte

AI Artificial Intelligence

ILSVRC ImageNet Large Scale Visual Recognition Challenge

USA United States of America

ReLU Rectified Linear Units

CIFAR Canadian Institute For Advanced Research

OBIS-SEAMAP Ocean Biodiversity Information System Spatial Ecological Analysis of Megavertebrate Populations

MNIST Modified National Institute of Standards and Technology

GANs Generative Adversarial Networks

UK United Kingdom

NDD Northumberland Dolphin Dataset

[2] `http://oomdata.arditi.pt/prd/catalogofotoid/index.php`

1.7 Structure of the Document

The dissertation first describes the existing state-of-the-art architectures (Section 2.3). Existing deep learning datasets that are mostly used for image classification are briefly depicted (Section 2.4) together with the used datasets of marine mammal individuals. Literature review describing the usage of AI in identifying marine mammals as well as the popular software in performing such is shown (Section 2.1). The methodology covers the three performed experiments with the former identifying the role of data augmentation on single model architecture (Section 3.1) as well as the benchmarking of the existing state-of-the-art CNN architectures when identifying marine mammal individuals (Section 3.2), and the benchmarking of the same state-of-the-art CNN architectures to discriminate other features (Section 3.3). Results of the three experiments are depicted in Section 3, while the discussion outlining the limitations and room for improvement is in Section 4.

2 Related Work

In the forthcoming, state-of-the-art literature on the usage of AI for identifying marine mammals is described, outlining the gap for future research. Also, a method to obtain the correct photograph of marine mammal individuals is described, which is typically performed by marine ecology experts. The dissertation further lists some of the most popular deep learning architectures and datasets, briefly describing ones that be used for further benchmarks when identifying marine mammal individuals.

2.1 AI and Identification of Marine Mammals

A study has proposed a system based on convolutional neural networks to automatically detect and count whales in satellite images. This approach is divided into two parts, the first CNN finds all images containing a whale, and the second CNN locates and counts the whales in those pictures. The study managed to obtain an F1-score of 81% in detecting and 94% in counting whales [10].

In this work, they developed an open-source application that uses neural networks to autonomously locate dorsal fins in field images, quantify an individual's unique fin characteristics, and match them to an existing photograph catalog. In their tests, the correct identity was ranked in the first position in 88% of cases and was within the top 50 ranked positions in 97% of cases. They also observed that finFindR's matching capabilities are robust to moderate variation in image quality and fin distinctiveness. This app also allows users to build a catalog of known individuals through time and match an unlimited number of individuals instead of being restricted to a predefined set [11].

Additional efforts approached the problem of automatically cropping cetaceans images with a hybrid technique based on domain analysis and deep learning. The domain knowledge is applied to propose relevant regions with the objective of highlighting the dorsal fins then a binary classification of fin or no-fin is performed by a CNN. The achieved results show the feasibility of this approach, managing to reach an accuracy of 92%, a sensitivity of 85%, and a specificity of 95% [12].

Other scholars hosted a data science challenge on the crowd-sourcing platform Kaggle to automate the identification of endangered North Atlantic right whales more specifically Eubalaena glacialis. The best solution automatically identified individual whales with an 87% accuracy. Their model consisted of a fully automated pipeline utilizing several CNNs that identified the region of interest of an image with specific key points and then identify the correct individual whale from the images [13].

Recent works highlight the multidisciplinary approach that is needed to accelerate and bring innovations working in domains like marine biology and computer science in this case study. They also present recent advances in the photo identification of Risso's dolphins, covering the process from manual approaches to modern deep-learning techniques. In the first part, they present the state-of-the-art methods currently applied to the photo-identification task, on the second they describe the Smart Photo-Identification of Risso's dolphins (SPIR) methods that were developed [14].

In addition, a study of the feasibility of recognizing individual right whales (Eubalaena glacialis) using 4 CNNs was performed. To test the applicability of deep learning to whale recognition they have developed several models based on best practices from the literature and then described the performance of the models. The CNNs that they used were the following. The DumbNet, a CNN that was designed by them, consists of 5 convolution layers each one followed by a pooling layer and achieved a validation error of 7.3%. They also used the famous AlexNet, which was difficult to work with because of the nonstandard convolution and pooling layers. The VGGNet, that they had problems running due to the huge computational cost. And the DeepSenseNet which was inspired by the VGGNet model, this one reached an 80% classification accuracy on the training set and 15% on the validation set [15].

A new study proposes a highly accurate deep learning-based dolphin identification algorithm. They present an advanced approach for feature extraction and efficiently integrate several well-known techniques to make dolphins distinguishable. The proposed techniques, avoid the background part (seawater) to affect the identification results, which is usually a problem. In simulations, they also manage to achieve an accuracy rate of 85% on the identification of dolphins, an F1 score of 81%, and a precision of 70% [16].

Another study focuses on killer whales and on image identification in a fully automated, multi-stage, deep-learning framework with the name FIN-PRINT. FIN-PRINT is trained and evaluated on a dataset collected over an 8-year period in the waters of western North America. The first step is to perform object detection to identify killer whales' marks, then all the previously identified markings are extracted. In the third step, a data enhancement mechanism is introduced and in the final step, multi-class individual recognition is performed. When evaluated on the network test set, it managed to achieve an accuracy of 92.5% on the killer whales identification [17].

Others presented the next generation of photogrammetry methods utilizing a CNN to demonstrate the potential of a deep learning-based photogrammetry system for automatic species identification and measurement and then analyse the same data using conventional methods to validate their approach. On their results, they correctly predicted whale species with 98% accuracy for humpback whales, minke whales, and blue whales [18].

Photo-id has been leveraged in recent work where it described the development of a new CNN-based photo-identification algorithm for individual humpback whales. The method uses a DenseNet to extract special key points of an image of the fluke and then a separate DenseNet trained to look for features within these key points. The features that were extracted are then compared against those of the reference images of humpback whales for similarity. In testing, they reached an accuracy of 99% for excellent quality images and 85% for poor quality images [19].

Other authors challenged the current thinking that pigmentation is an unreliable feature for delphinid photo identification and suggests that this feature could be applied to common dolphins and other poorly-marked delphinids. They used a photo-identification catalog of 169 adult individuals collected between 2002 and 2013 and extracted features that quantified pigmentation in a way that was robust to lighting artifacts and dorsal fin orientation. They determined the proportion of individuals who exhibited pigmentation and examined temporal stability by visually examining individuals. They found out that 88/91% of images could possibly be manually matched to the

correct individual in the catalog based just on pigmentation patterns. Also, an linear discriminant analysis classifier used to discriminate between individuals in the catalog managed to reach a 77.2% accuracy [20].

More recently, a computer vision approach for the photo-identification of the Risso's dolphin has been shown, exploiting specific visual cues relying on SIFT and SURF feature detectors. Their experiments are using image data obtained in the Gulf of Taranto between 2013 and 2017, they realized a comparative analysis of the performance of both SIFT and SURF and also used the software Darwin as state-of-the-art for comparison. This study managed to prove the usefulness of their approach and suggested that its application would be suitable for large-scale studies. The SIFT was capable to outperform the SURF by reaching a 99% accuracy while the SURF only achieves the 89%, the state-of-the-art Darwin only managed to reach a 68% accuracy [21].

Scholars also investigated the employment of machine learning strategies for the automated photo-ID of Risso's dolphin. In particular, the performances of the RUSBoost algorithm are outstanding in identifying the unknown dolphins, experimental results reinforce the potential of machine learning in the automation of the photo-ID process, and also enhance the necessity of collecting more data in order to obtain more effective data analysis. Their method is divided into three parts, the pre-processing that consists of the automated segmentation of the fin in the image in analysis, the features extraction of each image, and the classification performed by the RUSBoost algorithm. In the results, they reached an accuracy of 84% and specificity of 69% [22].

Since the great quantity of image data and the constrained properties of the dorsal fin images make photo identification problematic. A system has been developed to combat this problem, DARWIN. DARWIN is a software system that allows marine scientists to maintain information for the study of various behavioral and ecological patterns of bottlenose dolphins. This software gives a graphical user interface to access a series of dorsal fin images along with textual information describing the individual animals. Researchers may compare the dorsal fin image of an unknown individual against a database of images of identified dolphins previously collected in order to identify those fins which are more similar. A modified mean squared error is utilized as a measure of similarity and the images identified that are most similar are presented in rank order to the user [23].

There are a lot of research organizations that use photo-identification on cetaceans, for example, the Cascadia Research from the USA, the Dolphin Biology and Conservation from Italy, the Duke University Marine Lab also from the USA, Murdoch University Cetacean Research Unit from Australia, Sea Mammal Research Unit from Scotland, University of Otago Marine Mammal Research Group from New Zealand and some more. In the process of photo-identification researchers are concerned with the presence or absence of individuals in the capture histories used to estimate abundance, but this process begins with the selection of the images for inclusion in the analysis. The inclusion of poor-quality images increases the risk of making incorrect matches, so researchers have to define a certain quality threshold to consider an image suitable for use in the process [24].

In this work the computer program DARWIN was used to catalog and match fin IDs of white shark individuals collected between 2007 and 2011 in South Africa, this was the first study to successfully use the software for white shark identification. They also tried to discover the popula-

tion size in this area but one of the problems was that some of the sharks used for the estimation were killed and consequently were no longer part of the population. The program performed well despite a number of individual fins showing drastic changes in dorsal fin shape over time. Of the 1682 fin IDs used, the software managed to identify 532 unique individuals [25].

Some scholars also examined if and how researchers use computer vision systems in their photo-ID process and also developed an experiment to evaluate the performance of the most recent and used ones to identify bottlenose dolphin individuals using a photo-ID database of known individuals curated by the Chicago Zoological Society's Sarasota Dolphin Research Program in the USA. Realized surveys show that nowadays most researchers still use manual methods to identify unknown dorsal fin images. Their experimental evaluations of finFindR R application, CurvRank, CurvRank v2, and finFindR implementations in Flukebook point out that high match rates can be achieved with these systems, the highest match rates were found only when good or excellent quality images of fins with good distinctiveness are included in the process. In 2019 they organized a workshop as part of the World Marine Mammal Conference in Barcelona that included presentations of several computer vision and database systems currently used for dorsal fin photo identification like Finscan, Photo-ID ninja, a Google Application, Happywhale, finFindR, CurvRank, Flukebook, FinBase and OBIS SeaMap/GoMDIS. The finFindR R application, the CurvRank, and the CurvRank2 algorithms were able to achieve accuracy over 91.94% on the identification of bottlenose dolphin individuals [26].

Finscan is a system that creates and maintains a database of images of dorsal fins or flukes, the user inserts a new image acquired in the field, then the system searches the database for similar images based on the details of the dorsal fin or fluke, after that the system returns a list of database members similar to the input image. This system has already been tested with datasets of different species, but it was possible to notice that performs better on animals with well-marked dorsal fins. The system got good results with dolphins and white sharks, but the results weren't so good with pilot whales due to the dorsal fins having a shape that introduces some difficulties in matching them. The system also wasn't so effective in matching the sperm whale flukes, because the overall shape of flukes differs a lot from the originally targeted dorsal fins [27].

Photo-ID ninja is a system divided into 2 parts, in part 1 the system automatically crops dorsal fins from photographs, and in part 2 it matches those dorsal fins. This system has 3544 images of 185 dolphins and a deep convolutional neural network to learn features of the cropped dolphin fin images [3].

Others created a fully automatic photo-id tool to detect and identify various cetacean species, that is capable to provide the most likely matches based on the information available without the need for data preprocessing, they also compared their approach with other known photo-id tools. The dorsal fin detection is done through a Mask R-CNN model that locates the regions of interest in the images, then the Morphological Transformations are done based on the former knowledge of cetaceans, and after that, the colour thresholding is executed. Then the input images are cropped and in the last step, the most likely catalog matching is applied using a trained Siamese Neural

[3] https://sites.google.com/mote.org/evaluation-of-photo-id-methods/rise-of-the-machines-workshop?pli=1

Network (SNN) with a triplet loss function. In order to evaluate their approach the Northumberland Dolphin Dataset (NDD) was used, this dataset contained images of bottlenose dolphins and also white-beaked dolphins, the images were collected in 2019 on the coast of Northumberland in the UK. They also used another dataset that consisted of 250 images of 23 individuals of bottlenose dolphins in the waters around Naples in the USA. In the results their method reached an accuracy of 41% in the NDD, while in the Naples dataset the performance was better, achieving a 74% accuracy [28]. The Google Cloud AI is being used for tracking individuals through time, they are able to identify individual dolphins in seconds that have been traveling hundreds of miles over the last 10 years, something that would take many hours to make by eye.

Another system presented was FinFindR, FinFindR reduces the time of the photo-identification process by autonomously finding and isolating dolphin fins in field photos, tracing the trailing edge of fins in the images, and then presenting to the user the top 50 most likely matching identities. This software also allows the user to do the final identity determinations. In testing, FinFindR achieved an 88% accuracy. During a comparison test between FinFindR and human-only matching methods, both achieved an accuracy of 97% but while the human-only method needed 124 images on average before making a match the FinFindR only needed 10. It's possible to say with confidence that this software will facilitate a lot the photo-identification process in this case [29].

CurvRank is a software that first uses a dynamic programming time-warping algorithm to align two representations and interprets the alignment cost as a measure of similarity. Secondly, it interprets the representation as a feature descriptor. Descriptors for the set of known individuals are stored in a tree structure, which allows to perform queries given the descriptors from an unknown trailing edge. This software got good results in the classification of individuals for bottlenose dolphins and humpback whales, it achieved a 95% accuracy for bottlenose dolphins and an 80% accuracy for humpback whales [30].

FinBase is a database system that facilitates data entry and analyses, expedites the matching and cataloging processes, and reduces errors associated with manual image file management. This system is a Microsoft Access database customized using Microsoft Visual Basic for Applications that utilize ActiveX control to provide digital image analysis functionality. This system was initially created for a bottlenose dolphin research application but it can also be used on a variety of other species. In this system, a lot of the tasks associated with input data, analysis of photos, and data visualization were automated using a collection of customized database forms with easy-to-use interfaces. The primary forms related to the photo analysis process include a series of forms like catalog search, match fin, new fin, and clean fin. The catalog search form allows users to search the already existing catalog for matches to individuals that were photographed. When the users realize the search individuals in the catalog are presented in a sorted order based on the collection of fin attributes that were identified. An individual's catalog position is defined through five criteria, first one is the number of attributes matching search attributes, the second one is the number of additional attributes kept by the catalog individual, the third one is the priority of the search attributes matched with the individual's attributes, the next one is the calculated weighted-difference score for the individual and the fifth and last criteria is the catalog id [31].

Others implemented a convolutional neural network to identify and recognize humpback whales through their tail patterns. Their approach collects whale tail images, then trains the neural network by analyzing and pre-processing those images using TensorFlow and Keras frameworks, and then tries to identify the whale class of each image in the testing set. The dataset used in this study contained 25000 images, 9851 images for training and 15600 for testing, and a total of 4251 different individuals. The model implemented in the study was able to reach an accuracy of 78.5%. They also use models who participated in the Kaggle competition as state-of-the-art to compare with the results obtained by their model [32].

An interesting study has estimated the population size of Indo-Pacific bottlenose dolphins on the coast of South Africa. The dataset used was collected during a period of three years and composed of over 10000 images, 1569 individuals were identified. When using a POPAN formulation of the software program MARK, it was estimated population size of 28482 individuals [33].

Another study compared the performance of PIE v2, a triplet loss network with the performance of two fluke trailing edge-matching algorithms, the CurvRank v2 and the Dynamic Time Warping on the identification of sperm whale individuals through fluke images. The dataset used in the study was collected between 2005 and 2018 on the coast of the island of Dominica and consisted of 1592 images with a total of 512 individual sperm whales. Before the algorithm's classification, a fluke detection was performed using a customized PyTorch implementation of YOLO v2, whose job was to draw the bounding boxes around the flukes. In the classification results, the trained PIE v2 model was capable of outperforming the CurvRank and DTW algorithms reaching an 87% accuracy [34].

Other scholars have developed a new method based on deep learning named NNPool to classify Risso dolphin individuals, the used dataset was collected between 2013 and 2018 in the Northern Ionian Sea [35]. The results obtained were also validated by another dataset of Risso dolphins collected in 2019 in the Northern Ionian Sea and the Azores. The performance of the model was compared to an existing one named RUSPool which also can be used to identify Risso dolphin individuals, where both of them used the dorsal fins to classify the individuals. The CNN built by them is composed by applying three times the sequence, Convolutional layer, ReLu, and Max-Pooling layer. The NNPool method managed to obtain an 87% accuracy, a 70% sensitivity, and 90% specificity while the RUSPool model only achieved an accuracy of 78%, a sensitivity of 58%, and a specificity of 81%. These results show clearly that their model was capable to outperform the RUSPool model in the assigned task. Another interesting fact was that the time for RUSPool training was 2289 seconds while the NNPool training time was 1817 seconds [36].

In addition, the authors also managed to make a survey about systems that researchers have heard about and used. The researchers who participated in this survey were from 22 different countries and were studying 36 different marine mammal species and 1 shark species. The results clearly point out that the most heard system was Darwin with 37 votes, followed by Finscan with 27 votes and finFindR with 26 votes. In the 4th position, there's Flukebook with 20 votes. In terms of use, the most used system by them was finFindR with 15 votes, second, it was the Flukebook with 5 votes, and third the Darwin system with 4 [26].

Existing state-of-the-art software is shown in Table 1, while existing efforts in applying CNNs to identify marine mammal individuals are depicted in Table 2. While most of the work showcases relatively high accuracy, not much work uses the focus on dorsal fins and pilot whales, combining such input into the deep neural networks, which is the topic of this dissertation.

Table 1: Comparison of existing software to identify marine mammal individuals.

Software Name	**Species**	**Focus**	**Software**	**Method**
finFindR [11]	dolphins	dorsal fins	desktop app	statistics (trailing edge)
Flukebook [37]	whales/dolphins	dorsal fins/flukes	web app	neural network
Darwin [23]	dolphins	dorsal fins	desktop app	statistics (trailing edge)
FinBase [31]	dolphins	dorsal fins	desktop app	statistics (trailing edge)
Google Application [26]	dolphins	dorsal fins	web app	neural network
Identifin Software [23]	sharks	dorsal fins	desktop app	statistics (trailing edge)
Photo-ID Ninja [5]	dolphins	dorsal fins	web app	neural network
Finscan [27]	marine animals	dorsal fins/flukes	desktop app	statistics (trailing edge)
Wildbook [38]	whales/sharks	dorsal fins/flukes/others	web app	neural network
I3S [28]	marine animals	head/dorsal fins/flukes	desktop app	statistics (trailing edge)
HappyWhale [26]	whales	flukes	web app	neural network
Fluke Matcher [34]	whales	fluke	desktop app	statistics (trailing edge)
CurvRank [30]	dolphins/whales	dorsal fins/flukes	desktop app	neural network

Table 2: Comparison of existing approaches and performances.

Author Name	**Method**	**Results**
Guirado 2019 [10]	Faster R-CNN	F1- 81% in detecting and 94% in counting
Thompson 2022 [11]	finFindR	accuracy - 88%
Renò 2020 [12]	CNN from scratch	acc - 92% sensitivity - 85% specificity - 95%
Bogucki 2019 [13]	3 CNNs	accuracy - 87%
Renò 2022 [14]	DARWIN	not available
Polzounov 2016 [15]	4 CNNs	DeepSenseNet training accuracy - 80%
Hsu 2018 [16]	DenseNet121	acc - 85.53% F1 - 81.46% precision - 69.8%
Bergler 2021 [17]	FIN-PRINT	accuracy - 92.5%
Gray 2019 [18]	Mask R-CNN	accuracy - 98%
Cheeseman 2021 [19]	DenseNet based	lowest acc - 84.6% highest acc - 99.2%
Pawley 2018 [20]	LDA classifier	accuracy - 77.2%
Reno 2019 [21]	SURF and SIFT	SURF accuracy - 89% SIFT accuracy - 99%
Maglietta 2018 [22]	RUSBoost	accuracy - 84% specificity - 69%
Moore 2022 [26]	finFindR R/CurvRank/CurvRankv2	accuracy - 91.94%
Weideman 2017 [30]	CurvRank	accuracy - 95% and 80%
Gilman 2019 [5]	Photo-ID Ninja	accuracy - 90.6% precision - 80.8%
Trotter 2022 [28]	SNN	NDD accuracy - 41% Naples accuracy - 74%
Blas 2020 [32]	CNN	accuracy - 78.5%
Holmberg 2021 [34]	PIE v2	accuracy - 87%
Maglietta 2020 [36]	NNPool	acc - 87% sensitivity - 70% specificity - 90%

2.2 Photo-Identification

Photo-identification is a very used research tool utilized to identify individual whales and other animals [4]. Basically the camera takes an image of the animal when this one is sighted, the amount of times that the animal is "captured" gives the size of the population in an area thought a statistical method named “mark–recapture” [39]. During these encounters, researchers try to photograph marks present on the animal to help the process of individual recognition, for example on dolphins or whales marks located on the trailing edge of the dorsal fin are important for individual recognition. After the data is collected, the photos go through a selection process where only the best photos are selected, then these photos are matched with each other to understand how many individuals were photographed during that photo session and after this is done a new and final process starts, that involves matching the newly encountered individuals with photos from the existing catalog.

One of the advantages of this approach is that this is an indirect and non-invasive method to record the presence of a specific animal in a sighting since it doesn’t bother the animal in question. The identification of individual cetaceans has started a long time ago, an example of that is the killer whales in Australia in the nineteenth and early twentieth centuries, where whalers and fishermen identified some of at least 27 individuals, mainly by markings on and near the dorsal fin [6]. Nowadays, it’s possible to say with good enough photographs, a good portion of the population of almost any cetacean species can be individually identified. Recognizable animals allow researchers to know the surfacing-respiration-dive cycles and their general behavior patterns like resting, socializing, traveling, and feeding [6].

[4] `https://www.blue-world.org/research/research-methods/photo-identification/`

One of the limitations of photo-identification is the longevity and changeability of marks, over the years is normal that some dorsal fins gain more scars or marks which will make the identification process more difficult as in the previous images these marks were not present. In order to take good photos to use in photo-identification from shore, the researchers have to follow some rules, for example, photographs shouldn't be taken from more than 15 m above sea level, nor from further than about 500 m from the specie, high vantage points are good places to photograph individuals also [6]. On photos from boats which are the case of the photos used in the experiments of this thesis, the boats have to move with them and maneuver near the group or specie for the best possible view, the best photos are the ones taken from a 90-degree angle with a good focus and contrast on specie [5]. The problem is that boats are potentially disruptive to the natural behavior of species, so boat operators must approach dolphins slowly with constant velocity, in general use common sense in order to minimize noise so it doesn't disrupt the normal behavior of the specie.

Based on the aforementioned photo-id technique, the dissertation will further enhance the existing dorsal fins datasets, exploring the role of AI in identifying marine mammal individuals. In the forthcoming, the dissertation first outlines existing deep learning architectures, capable of addressing such challenges.

2.3 Existing Deep Learning Architectures

Below, the dissertation describes popular state-of-the-art architectures which will be used for benchmarking the AI models when identifying marine mammal individuals.

AlexNet. AlexNet is a classic convolutional neural network architecture that consists of convolutions, max pooling, and dense layers as the building blocks, here the convolutions are used in order to adjust the model across 2 GPUs [40]. This deep convolutional neural network was trained to classify the 1.3 million high-resolution images in the LSVRC-2010 ImageNet training set into 1000 different classes. They were able to achieve achieved top-1 and top-5 error rates of 37.5% and 17%. The error rate is a measure of the difference between what the Artificial Intelligence Network predicts and the real label of data. It's calculated through the difference between the value of a variable given by the model and the true value of that variable, the lower the error rate the better. AlexNet also managed to win the 2012 ImageNet competition with a top-5 error rate of 15.3%. Their neural network has 60 million parameters, 500000 neurons, and consists of 5 convolutional layers. To speed up the training process they used non-saturating neurons and a very efficient GPU implementation of convolutional nets by allowing multi-GPU training, this means putting half of the model's neurons on one GPU and the other half on another GPU. Due to a large number of parameters, AlexNet had a big issue in terms of overfitting. So they implemented 2 methods to reduce this, data augmentation and dropout. Dropout is a technique that consists of switching off neurons with a predetermined probability. In other words, every iteration uses a different sample of the model's parameters, which makes that each neuron needs to have more robust features that can be used with other random neurons. AlexNet has brought new approaches to convolutional neural networks, like the use of ReLU (Rectified Linear Units) instead of the standard tanh function. The advantage is that a CNN using ReLU is able to reach a certain error

[5] `https://archipelago.gr/en/photo-identification-an-invaluable-tool-in-cetacean-research/`

on the CIFAR-10 dataset six times faster than a CNN using tanh. Another new approach was the introduction of overlapping pooling that allowed a reduction in error of 0.5% and made the model harder to overfit. Every model has trainable and non-trainable parameters. Trainable parameters are those whose value changes according to their gradient, while non-trainable parameters mean the number of weights that don't change during training. To calculate the number of trainable parameters only the convolutional layers and the fully connected layers need to be looked at, in the convolutional layers it's used a mathematical equation $((M \times N \times D)+1) \times K$, where M stands for width, N for height, D for previous layer's filters and K for all the filter in the current layer. On the fully connected layer, the mathematical equation applied is $((C \times P)+1 \times C)$, here C represents the number of neurons in the current layer and P represents the number of neurons in the previous layer. Then the total number of trainable parameters is calculated by adding the parameters present on these two types of layers. In table 3 is possible to see the AlexNet architecture that was used in the 3 experiments.

Table 3: AlexNet Model Architecture.

#	**Layer (type)**	**Output Shape**	**Param #**
1	input (InputLayer)	(256, 256, 3)	0
2	conv2d (Conv2D)	(256, 256, 16)	208
3	max_pooling2d (MaxPooling2D)	(128, 128, 16)	0
4	conv2d_1 (Conv2D)	(128, 128, 32)	2080
5	max_pooling2d_1 (MaxPooling 2D)	(64, 64, 32)	0
6	conv2d_2 (Conv2D)	(64, 64, 64)	8256
7	max_pooling2d_2 (MaxPooling 2D)	(32, 32, 64)	0
8	dropout (Dropout)	(32, 32, 64)	0
9	flatten (Flatten)	(65536)	0
10	dense (Dense)	(500)	32768500
11	dropout_1 (Dropout)	(500)	0
12	dense_1 (Dense)	(1240)	621240
	Total params: 33,400,284		
	Trainable params: 33,400,284		
	Non-trainable params: 0		

LeNet. LeNet is a classic convolution neural network small and easy to understand but also large enough to obtain good results. This model was utilized for the handwritten digit recognition task with the MNIST dataset and served as inspiration for the development of more complex networks like AlexNet and VGGNet. Their neural network, which contained 60 million parameters and 650000 neurons, also had 5 convolutional layers, some of which are followed by max-pooling layers and 3 fully-connected layers with a final 1000-way softmax. To accelerate the training they used non-saturating neurons and a very efficient GPU (Graphics Processing Unit) implementation of the convolution operation. In order to reduce overfitting in the fully-connected layers they implemented a method called "dropout" that was successful. There are a series of variants of this model like LeNet-1, LeNet-4, LeNet-5, and Boosted LeNet-4. But the most popular of them is clearly the LeNet-5 which was released in 1998. LeNet-5 is constituted of 7 layers. The layer composition consists of 3 convolutional layers, 2 subsampling layers, and 2 fully connected layers [41]. The size of the input image is 32×32 pixels, first, there's a convolutional layer of 6 filters of size 5×5 and stride 1, and the resulting feature map is 28×28×6. Then there's an average pooling layer with a filter size of 2×2 and stride 1, the output feature map is 14×14×6. Next, it's another convolutional layer with 16 filters of 5×5 and stride is 1, after that there's again an average pooling layer of 2×2 and stride 2, resulting in a feature map of 5×5×16. Then there is the final convolutional layer with 120 filters of 5×5 and stride 1 creating a feature map of size 120. After that, there's a fully connected layer with 84 neurons that result in the output of 84 values. The last layer is an output layer with 10 neurons (since there are 10 classes) and the softmax function. What the softmax function does is to give the probability that a data point belongs to a particular class. Due to this neural network being the oldest compared with the others models, it still uses the Tanh activation functions instead of the ReLU activation function, who tends to give much better classification accuracy. An activation function basically chooses whether a neuron should be activated or not. Table 4 shows the LeNet architecture used in the 2 experiments described further ahead.

Table 4: LeNet Model Architecture.

#	**Layer (type)**	**Output Shape**	**Param #**
1	input (InputLayer)	(256, 256, 3)	0
2	conv2d (Conv2D)	(252, 252, 128)	9728
3	max_pooling2d (MaxPooling2D)	(126, 126, 128)	0
4	dropout (Dropout)	(126, 126, 128)	0
5	conv2d_1 (Conv2D)	(122, 122, 256)	819456
6	max_pooling2d_1 (MaxPooling 2D)	(61, 61, 256)	0
7	dropout_1 (Dropout)	(61, 61, 256)	0
8	flatten (Flatten)	(952576)	0
9	dense (Dense)	(100)	95257700
10	dense_1 (Dense)	(1240)	125240
	Total params: 96,212,124 Trainable params: 96,212,124 Non-trainable params: 0		

VGGNet. Released in 2014, this work focused on investigating the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting [42]. The main contribution was a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows us that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16-19 weight layers. In this architecture, the 2 most popular models are the VGG-16 and VGG-19 but there are also VGG-11, VGG-11 (LRN), and VGG-13. The only difference between the 2 most popular ones is that one has 16 layers and the other has 19. The VGG-16 takes as input an RGB image of size 224×224, then it goes through 2 convolutional layers with 64 filters of size 3×3 and activation function ReLU, after that there's a max pooling layer of 2×2. Next, there are 2 convolutional layers with 128 filters again with size 3×3 and a max pooling layer of 2×2. After, there are 3 convolutional layers with 256 filters and max pooling. Then there are 2 times more 3 convolutional layers with 512 filters and max pooling. After that comes 3 fully connected layers with 25088, 4096, and 4096 neurons respectively. The last layer is an output layer with 1000 neurons (same number of classes) and the softmax function. They also showed that their representations generalize well to other datasets, where they achieve state-of-the-art results. They have made their two best-performing ConvNet models publicly available to facilitate the life of researchers on the use of deep visual representations in computer vision. In the ILSVRC-2014 (ImageNet Large Scale Visual Recognition Challenge) the "VGG" team managed to obtain a second place with 7.3% test error using an ensemble of 7 models. This challenge basically evaluates algorithms for object detection and image classification at a large scale.

Inception. The Inception is a deep convolutional neural network architecture that achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 [43]. The Inception architecture uses 12 times fewer parameters than the winning architecture of the year 2012 while being significantly more accurate. The biggest difference of this architecture is the better use of computational resources inside the network. The depth and width of the network were increased while keeping the computational budget constant. To improve quality, the Hebbian principle and the intuition of multi-scale processing were used in architectural decisions. The popular versions of this architecture are, Inception v1, Inception v2, Inception v3, Inception v4, InceptionResNet v1, and InceptionResNet v2 in short each version is an improvement of the previous one. Due to the fact that the one used in experiments 2 and 3 is the InceptionResNet v2 you can see its summary in table 5. Let's look at Inception v1 (GoogLeNet), this model was revolutionary because of the Inception module, a module that has filters with multiple sizes operating at the same level instead of stacking them sequentially like in the other convolutional neural networks. This approach solves problems like choosing the right kernel size, very deep networks tend to overfit, and stacking large convolution operations becomes computationally expensive. The simplified version of an inception module works by performing a convolution on an input with 3 different sizes of filters, 1×1, 3×3, 5×5, and also a max pooling layer, after that the resulting outputs are concatenated and sent to the next layer. But as was previously said deep neural networks are computationally expensive, so in order to battler this, they made some changes and invented the Inception module with dimension reduction, basically they added a convolutional layer size 1×1 before the 3×3 and 5×5 convolutions thus reducing the number of input channels, they also added a 1×1 convolution not before but after the max

pooling layer. This module is the one that was used on the building of GoogleNet (Inception v1), GoogleNet is 27 layers deep and has 9 inception modules, all the convolutional layers present in the architecture use the activation function ReLU. They also presented a solution to a typical problem that appears in deep neural networks, the vanishing gradient problem. The vanishing gradient problem consists that when exists many layers in a network, the value of the product of the derivative decreases until at some point the partial derivative of the loss function reaches a value near zero, and the partial derivative vanishes, and consequently no learning is being performed. So they made two auxiliary classifiers that essentially apply softmax to the outputs of two of the inception modules and computed an auxiliary loss over the same labels.

Table 5: InceptionResNetV2 Model Architecture.

#	**Layer (type)**	**Output Shape**	**Param #**
1	input_4 (InputLayer)	(256, 256, 3)	0
2	conv2d_4 (Conv2D)	(127, 127, 32)	864
3	batch_normalization (BatchNormalization)	(127, 127, 32)	96
4	activation (Activation)	(127, 127, 32)	0
5	conv2d_5 (Conv2D)	(125, 125, 32)	9216
6	batch_normalization$_1$($BatchNormalization$)	(125, 125, 32)	96
7	activation_1 (Activation)	(125, 125, 32)	0
8	conv2d_6 (Conv2D)	(125, 125, 64)	18432
9	batch_normalization$_2$($BatchNormalization$)	(125, 125, 64)	192
10	activation_2 (Activation)	(125, 125, 64)	0
11	max_pooling2d$_4$($MaxPooling2D$)	(62, 62, 64)	0
12	conv2d_7 (Conv2D)	(62, 62, 80)	5120
13	batch_normalization_3 (BatchNormalization)	(62, 62, 80)	240
14	activation_3 (Activation)	(62, 62, 80)	0
15	conv2d_8 (Conv2D)	(60, 60, 192)	138240
16	batch_normalization_4 (BatchNormalization)	(60, 60, 192)	576
.	...	...	...
.	...	...	...
.	...	...	...
x	activation_202 (Activation)	(6, 6, 256)	0
x+1	block8_10_mixed (Concatenate)	(6, 6, 448)	0
x+2	block8_10_conv (Conv2D)	(6, 6, 2080)	933920
x+3	block8_10 (Lambda)	(6, 6, 2080)	0
x+4	conv_7b (Conv2D)	(6, 6, 1536)	3194880
x+5	conv_7b_bn (BatchNormalization)	(6, 6, 1536)	4608
x+6	conv_7b_ac (Activation)	(6, 6, 1536)	0
x+7	flatten (Flatten)	(55296)	0
x+8	dense_2 (Dense)	(128)	7078016
x+9	dense_3 (Dense)	(1240)	159960
	Total params: 61,574,712		
	Trainable params: 61,514,168		
	Non-trainable params: 60,544		

ResNet. Due to deeper neural networks being more difficult to train, they created a residual learning framework to ease the training of networks that are much deeper than those used previously [44]. They have reformulated the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. They have provided comprehensive empirical evidence, which shows that these residual networks are easier to optimize, and can gain more accuracy from considerably increased depth. On the ImageNet dataset, they evaluated residual nets with a net 8 times deeper than VGG nets but still having lower complexity. With their results, they managed to achieve 1st place on the ILSVRC 2015 classification task having an error of 3.57%. Every layer of ResNet is composed of several blocks because when ResNets go deeper they normally do it by increasing the number of operations within a block, but the number of total layers stays the same. ResNet solved the vanishing gradient by introducing the residual blocks, here a technique called skip connections it's used. This technique connects activations of a layer to further layers by skipping some layers in between, forming a residual block. Resnets are made by stacking these residual blocks together, the residual blocks are used basically to improve the accuracy of models. The advantage of using skip connections is that if a layer hurts the performance of the architecture then it will be skipped by regularization. The ResNet model has a lot of variants like ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, ResNet-152, ResNet-164, and ResNet-1202. In these variants the concept is basically the same, the number of layers is the only thing that changes. In table 6 is possible to see the summary of ResNet50 architectures due to the fact that it's the one used in experiments 2 and 3.

Table 6: ResNet50 Model Architecture.

#	**Layer (type)**	**Output Shape**	**Param #**
1	input_3 (InputLayer)	(256, 256, 3)	0
2	conv1_pad (ZeroPadding2D)	(262, 262, 3)	0
3	conv1_conv (Conv2D)	(128, 128, 64)	9472
4	conv1_bn (BatchNormalization)	(128, 128, 64)	256
5	conv1_relu (Activation)	(128, 128, 64)	0
6	pool1_pad (ZeroPadding2D)	(130, 130, 64)	0
7	pool1_pool (MaxPooling2D)	(64, 64, 64)	0
8	conv2_block1_1_conv (Conv2D)	(64, 64, 64)	4160
9	conv2_block1_1_bn (BatchNormalization)	(64, 64, 64)	256
10	conv2_block1_1_relu (Activation)	(64, 64, 64)	0
11	conv2_block1_2_conv (Conv2D)	(64, 64, 64)	36928
12	conv2_block1_2_bn (BatchNormalization)	(64, 64, 64)	256
13	conv2_block1_2_relu (Activation)	(64, 64, 64)	0
14	conv2_block1_0_conv (Conv2D)	(64, 64, 256)	16640
15	conv2_block1_3_conv (Conv2D)	(64, 64, 256)	16640
16	conv2_block1_0_bn (BatchNormalization)	(64, 64, 256)	1024
.	...	...	...
.	...	...	...
.	...	...	...
x	conv5_block3_2_conv (Conv2D)	(8, 8, 512)	2359808
x+1	conv5_block3_2_bn (BatchNormalization)	(8, 8, 512)	2048
x+2	conv5_block3_2_relu (Activation)	(8, 8, 512)	0
x+3	conv5_block3_3_conv (Conv2D)	(8, 8, 2048)	1050624
x+4	conv5_block3_3_bn (BatchNormalization)	(8, 8, 2048)	8192
x+5	conv5_block3_add (Add)	(8, 8, 2048)	0
x+6	conv5_block3_out (Activation)	(8, 8, 2048)	0
x+7	flatten (Flatten)	(131072)	0
x+8	dense_2 (Dense)	(128)	16777344
x+9	dense_3 (Dense)	(1240)	159960
	Total params: 40,525,016 Trainable params: 40,471,896 Non-trainable params: 53,120		

DenseNet. DenseNet is a convolutional neural network developed to improve the declined accuracy because of the loss of information before reaching its destination due to the longer path between the input layer and the output layer. On this convolutional neural network, inside the Dense Blocks, each layer obtains additional inputs from all preceding layers and passes on its own feature maps to all subsequent layers. In other words, each layer receives collective knowledge from the preceding layers. Since each layer receives feature maps from all preceding layers, it allows the network to be thinner and more compact. The DenseNet is divided into Dense Blocks with a different number of filters, but the dimensions within the different blocks are the same. The Transition layers between the Dense Blocks apply batch normalization using downsampling. With this said DenseNet has several advantages, it alleviates the vanishing-gradient problem, strengthens feature propagation, encourages feature reuse, and substantially reduces the number of parameters. This architecture was tested on 4 highly competitive object recognition benchmark tasks, CIFAR-10, CIFAR-100, SVHN, and ImageNet. DenseNet managed to obtain significant improvements over the state-of-the-art on most of them while requiring less computing to achieve high performance. The DenseNet model has 4 variants, DenseNet121, DenseNet161, DenseNet169, and DenseNet201. The numbers denote the number of layers in each neural network. DenseNet121 was the variant used in experiments 2 and 3, and for that reason, table 7 presents its summary. The DenseNet121 architecture after the input has a convolution layer with filter size 7×7, then there is a pooling layer of 3×3, next it's Dense Block 1 with 6 convolution layers with filter size 1×1 and 3×3, and after that, it's Transition Layer, followed by Dense Block 2, then there's another Transition layer, next there is Dense Block 3 and again a Transition Layer. Finally, there's Dense Block 4, an average pooling layer size of 7×7 and a fully connected layer using a softmax activation function giving us the output.

Table 7: DenseNet121 Model Architecture.

#	**Layer (type)**	**Output Shape**	**Param #**
1	$\text{input}_1(InputLayer)$	(256, 256, 3)	0
2	$\text{zero}_p adding2d\ (ZeroPadding2D)$	(262, 262, 3)	0
3	conv1/conv (Conv2D)	(128, 128, 64)	9408
4	conv1/bn (BatchNormalization)	(128, 128, 64)	256
5	conv1/relu (Activation)	(128, 128, 64)	0
6	zero_padding2d_1 (ZeroPadding2D)	(130, 130, 64)	0
7	pool1 (MaxPooling2D)	(64, 64, 64)	0
8	conv2_block1_0_bn (BatchNormalization)	(64, 64, 64)	256
9	conv2_block1_0_relu (Activation)	(64, 64, 64)	0
10	conv2_block1_1_conv (Conv2D)	(64, 64, 128)	8192
11	conv2_block1_1_bn (BatchNormalization)	(64, 64, 128)	512
12	conv2_block1_1_relu (Activation)	(64, 64, 128)	0
13	conv2_block1_2_conv (Conv2D)	(64, 64, 32)	36864
14	conv2_block1_concat (Concatenate)	(64, 64, 96)	0
15	conv2_block2_0_bn (BatchNormalization)	(64, 64, 96)	384
16	conv2_block2_0_relu (Activation)	(64, 64, 96)	0
.	...	...	...
.	...	...	...
.	...	...	...
x	conv5_block16_1_conv (Conv2D)	(8, 8, 128)	126976
x+1	conv5_block16_1_bn (BatchNormalization)	(8, 8, 128)	512
x+2	conv5_block16_1_relu (Activation)	(8, 8, 128)	0
x+3	conv5_block16_2_conv (Conv2D)	(8, 8, 32)	36864
x+4	conv5_block16_concat (Concatenate)	(8, 8, 1024)	0
x+5	bn (BatchNormalization)	(8, 8, 1024)	4096
x+6	relu (Activation)	(8, 8, 1024)	0
x+7	flatten (Flatten)	(65536)	0
x+8	dense_2 (Dense)	(128)	8388736
x+9	dense_3 (Dense)	(1240)	159960
	Total params: 15,586,200		
	Trainable params: 15,502,552		
	Non-trainable params: 83,648		

MobileNet. This one is a class of convolutional neural networks that was open-sourced by Google, designed to be used in mobile applications and it's TensorFlow's first mobile computer vision model. This model uses depthwise separable convolutions, which reduces a lot the number of parameters compared to networks with regular convolutions. A depthwise separable convolution is made from 2 operations, a depthwise convolution and a pointwise convolution. Depthwise convolution is basically a type of convolution where it's applied a single convolutional filter for each input channel. The pointwise convolution is a type of convolution that uses a 1×1 kernel, this kernel iterates through every single point and has a depth of however many channels the input image has [45]. The main purpose of the MobileNet model is to maximize accuracy while being mindful of the restricted resources for an on-device or embedded application. Tests carried out to compare with the most popular models prove that this model outperforms the GoogLeNet and VGGNet models while the multi-adds and the parameters are much fewer, the model reached 70.6% of ImageNet accuracy while having only 4.2 million parameters and 569 million multi-adds. Currently, there are 3 variants of this model, MobileNetV1, MobileNetV2, and MobileNetV3. In table 8 is possible to see the summary of MobileNetV2 architectures due to the fact that it's the variant used in experiments 2 and 3.

Table 8: MobileNetV2 Model Architecture.

#	**Layer (type)**	**Output Shape**	**Param #**
1	input_2 (InputLayer)	(256, 256, 3)	0
2	Conv1 (Conv2D)	(128, 128, 32)	864
3	bn_Conv1 (BatchNormalization)	(128, 128, 32)	128
4	Conv1_relu (ReLU)	(128, 128, 32)	0
5	expanded_conv_depthwise (DepthwiseConv2D)	(128, 128, 32)	288
6	expanded_conv_depthwise_BN (BatchNormalization)	(128, 128, 32)	128
7	expanded_conv_depthwise_relu (ReLU)	(128, 128, 32)	0
8	expanded_conv_project (Conv2D)	(128, 128, 16)	512
9	expanded_conv_project_BN (BatchNormalization)	(128, 128, 16)	64
10	block_1_expand (Conv2D)	(128, 128, 96)	1536
11	block_1_expand_BN (BatchNormalization)	(128, 128, 96)	384
12	block_1_expand_relu (ReLU)	(128, 128, 96)	0
13	block_1_pad (ZeroPadding2D)	(129, 129, 96)	0
14	block_1_depthwise (DepthwiseConv2D)	(64, 64, 96)	864
15	block_1_depthwise_BN (BatchNormalization)	(64, 64, 96)	384
16	block_1_depthwise_relu (ReLU)	(64, 64, 96)	0
.	...	...	...
.	...	...	...
.	...	...	...
x	block_16_depthwise_BN (BatchNormalization)	(8, 8, 960)	3840
x+1	block_16_depthwise_relu (ReLU)	(8, 8, 960)	0
x+2	block_16_project (Conv2D)	(8, 8, 320)	307200
x+3	block_16_project_BN (BatchNormalization)	(8, 8, 320)	1280
x+4	Conv_1 (Conv2D)	(8, 8, 1280)	409600
x+5	Conv_1_bn (BatchNormalization)	(8, 8, 1280)	5120
x+6	out_relu (ReLU)	(8, 8, 1280)	0
x+7	flatten (Flatten)	(81920)	0
x+8	dense_2 (Dense)	(128)	10485888
x+9	dense_3 (Dense)	(1240)	159960
	Total params: 12,903,832 Trainable params: 12,869,720 Non-trainable params: 34,112		

Given the succinct list of architectures, the dissertation will focus on image classification and benchmarking aforementioned architectures, by further tuning the hyperparameters. Next, some of the existing popular datasets used in CNNs are also described.

2.4 Existing Datasets for Image Classification

Deep learning models are typically used with large input datasets. Below, the dissertation outlines some of the popular datasets used in CNNs and shows the existing ones when identifying marine mammals.

2.4.1 Traditional CNN Datasets

Below, traditional CNN datasets are described, which are typically used by deep learning and CNN. Existing datasets do not contain individual marine specimens and the gap remains in the creation of more robust datasets.

ImageNet dataset. ImageNet is an image database organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by a big number of images [40]. The project has been crucial in advancing computer vision and deep learning research. The data is available for free to researchers.

MNIST dataset. The MNIST database of handwritten digits, has a training set of 60000 examples and a test set of 10000 examples [46]. It is a subset of a larger set available from NIST. It is a good database for people who want to learn techniques and pattern recognition methods on

real-world data while spending little effort on preprocessing and formatting. The MNIST database was made from NIST's Special Database 3 and Special Database 1 which contain binary images of handwritten digits. NIST designated SD-3 as the training set and SD-1 as the test set. But SD-3 is much easier to recognize than SD-1. The reason for this is the fact that SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students. Drawing sensible conclusions from learning experiments requires that the result be independent of the choice of the training set and test among the complete set of samples. Therefore it was necessary to build a new database by mixing NIST's datasets. The MNIST training set is composed of 30000 patterns from SD-3 and 30000 patterns from SD-1. The test set was composed of 5000 patterns from SD-3 and 5000 patterns from SD-1. The 60000 pattern training set contained examples from approximately 250 writers.

MS COCO dataset. COCO is big-scale object detection, segmentation, and captioning dataset [47]. This dataset has several features like object segmentation, recognition in context, superpixel stuff segmentation, 330000 images, 1.5 million object instances, 80 object categories, 91 stuff categories, 5 captions per image, and 250000 people with keypoints. This dataset is also sponsored by companies like Facebook and Microsoft.

CIFAR-10. CIFAR-10 is a dataset that consists of 60000 32×32 color images and 10 classes, with 6000 images per class. Of the 60000 images, about 50000 images are for training, and 10000 are for testing.

80 Million Tiny Images. Currently, billions of images are now freely available online. In this research, they used a variety of non-parametric methods and a dataset of 79302017 images collected from the Internet [48]. The images in the dataset were stored as 32×32 color images, where each image is labeled with one of the 75062 non-abstract nouns in English, as listed in the Wordnet lexical database. The semantic information from Wordnet can be used in conjunction with other methods to perform object classification over a range of semantic levels successfully minimizing the effects of labeling noise.

Open Images Dataset. This dataset has almost 9 million URLs to images that have been annotated with labels spanning over 6000 categories. The dataset contains 9011219 training images, 41260 validation images, and 125436 testing images.

2.4.2 Potential CNN Datasets for Marine Species

There are existing efforts based on photo-identification which curate the datasets of the individuals. However, they remain with a low sample of marine mammal individuals.

MARE Pilot Whale Dataset. This is a dataset created by OOM (Observatório Oceânico da Madeira) in order to help assess the parameters and movements of populations of the kind pilot-whale, the dataset includes photographs of crew members and passengers of maritime tourist vessels operating in Madeira. This catalog is curated by Filipe Alves, Ana Dinis, and Rita Ferreira and had the contribution of Carlota Molina, Annalisa Sambolino, Francesca Guizzi, Anita Alessandrini, Mieke Weyn and companies like VMT Madeira, Ventura|NatureEmotions, Seaborn, H20 Madeira, and Lobosonda. The catalog provides useful information for the cetacean-watching industry on the occurrence and number of tropical pilot whales in their area of operation and also

serves as a baseline for future publications [49]. The data was collected between 2003 and 2011 around Madeira, Desertas, and Porto Santo islands. The research team used digital cameras with lenses ranging between 70-400 mm. Surveys had a mean duration of 6 hours and 16 minutes, efforts to photograph the left and right side of each individual were made, and over 45000 photographs were taken from 405 encounters. In total, they have done 7014 minutes of photo-id. Then, photo processing was performed which consisted of cropping the dorsal fin and the surrounding body area in order to facilitate the photo matching next [50]. After performing the selection process, the dataset used in the dissertation has a total of 1138 images and 856 individuals, so the ratio is 1.3 images per individual. An example of images of the MARE Dataset is shown in Figure 1.

Fig. 1: Some images of MARE Pilot Whale Dataset

MONICET. The MONICET is a platform that possesses datasets of whales and dolphins from different species along with the location where the photos were taken and the date, in total they have 13 different species present in their datasets. The photos were all taken in the waters of the Azores region. This project is based on an association between 3 whale-watching companies and the regional research center with the support of external consultants. The local researchers are Ana Neto, José Azevedo, and Marc Fernandez. The consultants are Carole Carlson, Jonathan Gordon, Lenin Oviedo, Natacha Soto, and Peter Evans [51]. An example image of the dataset is shown in Figure 2.

Happywhale. Happywhale is a web-based marine mammal photo ID platform released in August 2015. Initially, the platform only focused on collecting fluke photos of humpback whales, then it matched these photos with known individuals through automated image recognition. Now they are expanding to other whales and species too, like dolphins, seals, and penguins. On this platform is possible to see photos of different species along with the location and where the photos were taken. Since August 2015 they have received submissions of over 41000 images contributed by over 1000 scientists. The 2 co-founders of this project are Ted Cheeseman and Ken Southerland and the data managers are Marilia Olio and Hayley Newell. The partner organizations of this platform are Cascadia Research Collective and Allied Whale [52]. An example of this dataset is shown in Fig 3.

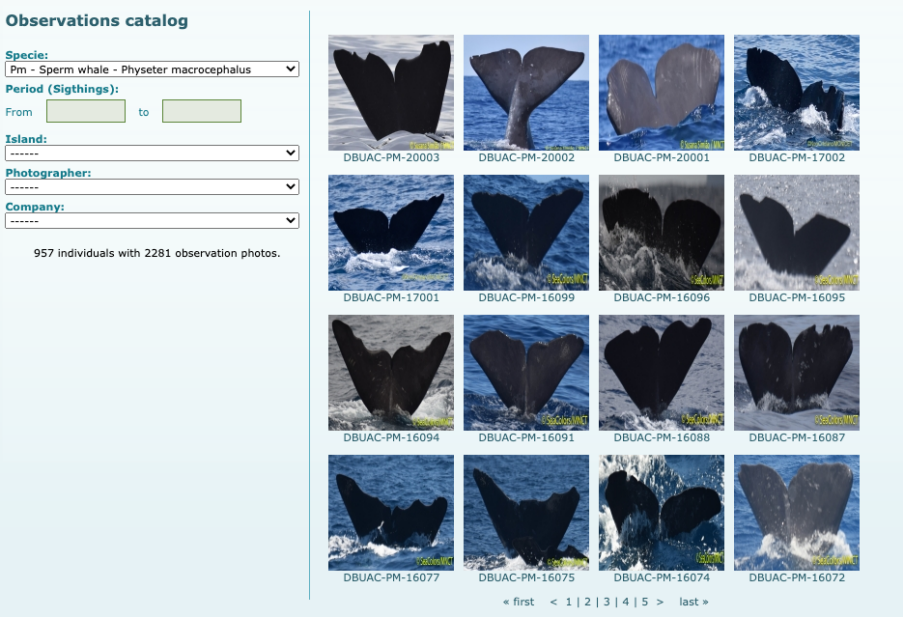


Fig. 2: Result of search on the MONICET catalog

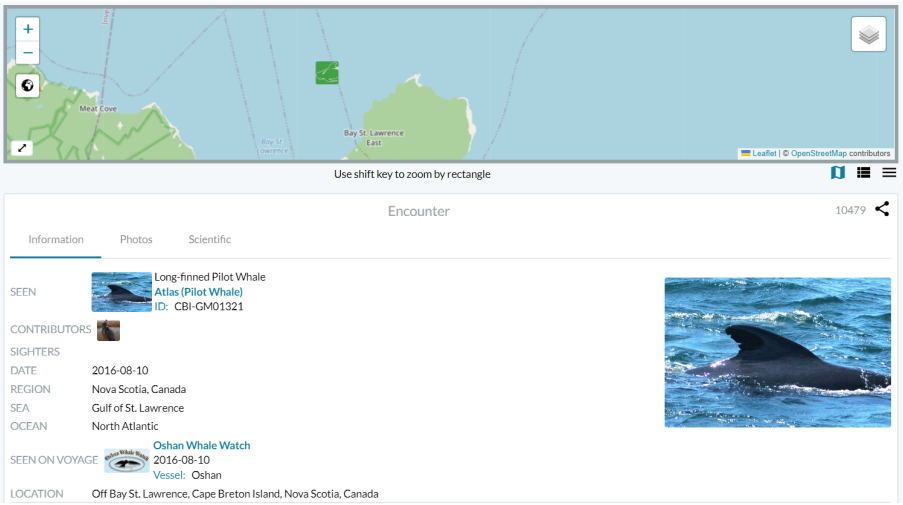


Fig. 3: Result of a pilot whale search in the Happywhale platform

OBIS-SEAMAP. OBIS-SEAMAP (Ocean Biodiversity Information System Spatial Ecological Analysis of Megavertebrate Populations) is the world data center for marine mammals, sea birds, and sea turtles. The system includes 4911450 observation records from about 1130 datasets and a total of 507 species, gathering data since 1935 [53]. Also, this system has been developed by the National Oceanographic Partnership Program and the Alfred P. Sloan Foundation since 2002. An example of a search is shown in Fig 4.

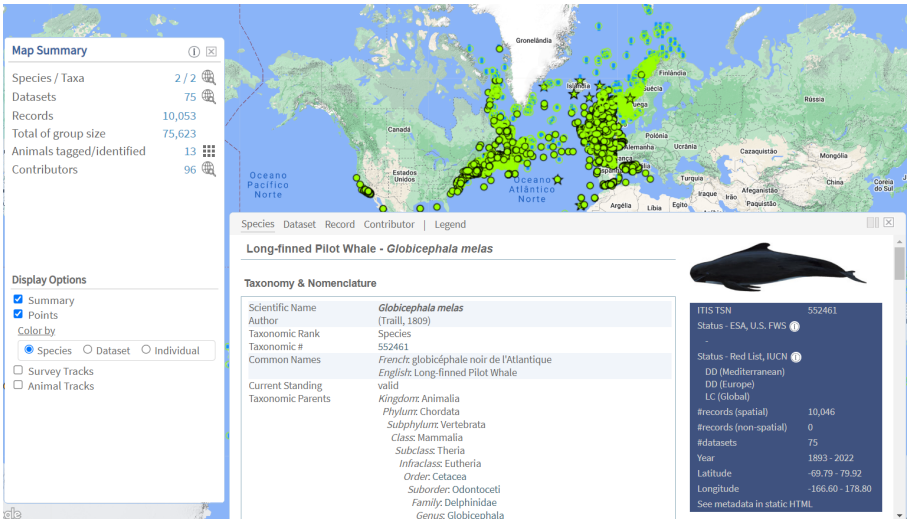


Fig. 4: Result of a search on the OBIS-SEAMAP platform

Flukebook. This is a software that uses photographs of flukes, dorsal fins, and scars to distinguish between individual animals. Users can submit images of their encounters having to give also the location and the date of the photographs. With this software, it's possible to identify individuals of species like humpback whales, sperm whales, orcas, bottlenose dolphins, spotted dolphins, right whales, fin whales, and humpback dolphins. The software allows users to follow an individual whale, find out where she goes and who has seen her lately [37]. In this software it's possible to realize a report but it's not possible to search for other reports because you need to have an account for that and there is no way to do a registration on the web page, so for that, it's not possible to validate this specific platform.

Summary. All of the proposed architectures in the literature will be used to further benchmark the obtained models, outlining which architecture works adequately in discriminating marine mammal individuals. As it's possible to see there's a lack of datasets and the existing ones although very useful for typical computer vision problems, remain unusable for the identification of such mammals, and therefore a novel annotated and labeled dataset will be generated, based on the dorsal fins, to be used in this work (Figure 5).

3 System

Next, the dissertation explores the identification of marine mammal individuals and the usage of dorsal fins throughout two experiments. The former explores the role of data augmentation to amplify the existing limited amount of individual specimens. The latter benchmark the state-of-the-art neural network architectures when addressing such challenges. Also, a third experiment was performed using the same neural network architectures of the previous experiment to classify a dataset into some categories in order to reduce the huge number of classes that previously existed. This work is based on transfer learning. Basically, transfer learning is a machine learning method that takes a model developed for a task and uses it as starting point for a model with a different task in its hands. In other words, the knowledge of an already trained machine learning model is applied to a different but related problem that doesn't have much data. So instead of starting the learning process from 0, it starts from patterns learned from solving a related task.

3.1 Experiment 1 - The Role of Augmented Dataset

To tackle the challenge of identifying marine mammal individuals dissertation describes the procedure for training and validating the model based on image classification.

Model. The used model was based on AlexNet architecture and contained a total of 1740 neurons, and 12 layers. In table 3 it is possible to see a summary of the model architecture and identify some parameters of the AlexNet architecture model. First presents the number of convolutional and pooling layers. Here it's needed to define the inputshape in the first convolutional layer only. The second layer consists of the convolutional and pooling layers with a ReLU activation function. The third layer is a convolutional and pooling layer. The fifth layer is a Dropout layer to avoid over-fitting with a 30% rate. The sixth layer flattens the last feature map into a vector of features. Seventh adds the first fully connected layer. Eight includes another dropout layer with a 40% rate Ninth is the output layer is a fully connected layer with 10 nodes and softmax activation to give probabilities to the 10 classes.

Dataset. The used dataset was obtained from MARE Insular team, containing 856 individual specimens. Each individual specimen had its own ID name (e.g. $GMa0909$) meaning that the species belonged to the Globicephala macrorhynchus (Gma) species – pilot whales. Furthermore, the dataset was boosted by performing the data augmentation which quadrupled the obtained imagery to 9288 images. Performed data augmentations were: vertical flip (making the image to be symmetrical around the vertical axis), hue saturation (increasing and decreasing the color saturation), Gaussian blur (performing filters for nearby pixels), and noise (increasing pixel saturation on image), etc. Data augmentation was performed using Supervisely's Data Transformation Language (DTL) and is depicted in figure 7. An example of the imagery before and after augmentation is seen in figure 6.

Fig. 5: An example of individual animals which are recognized and matched against the datasets by observing the shape of the dorsal fin. Each marine mammal has its own dorsal fin signature.

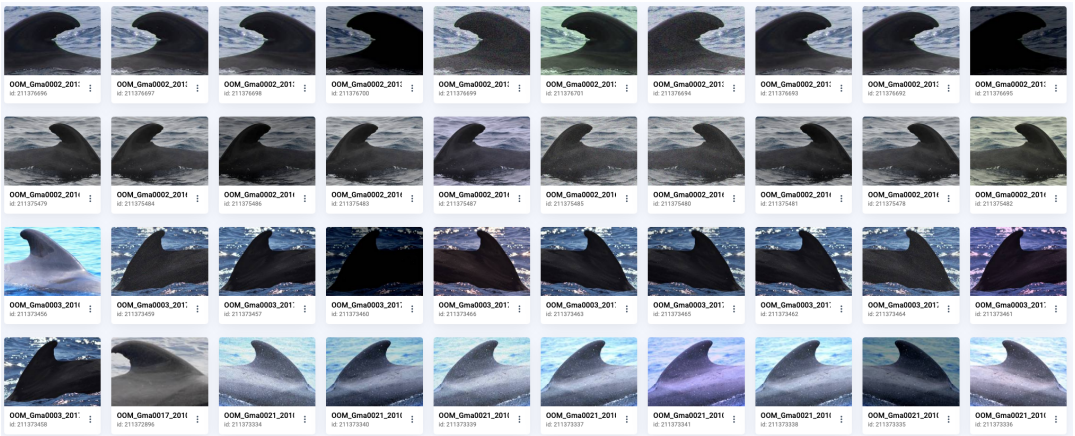


Fig. 6: An example of performed data augmentations on top of the existing dorsal fin dataset.

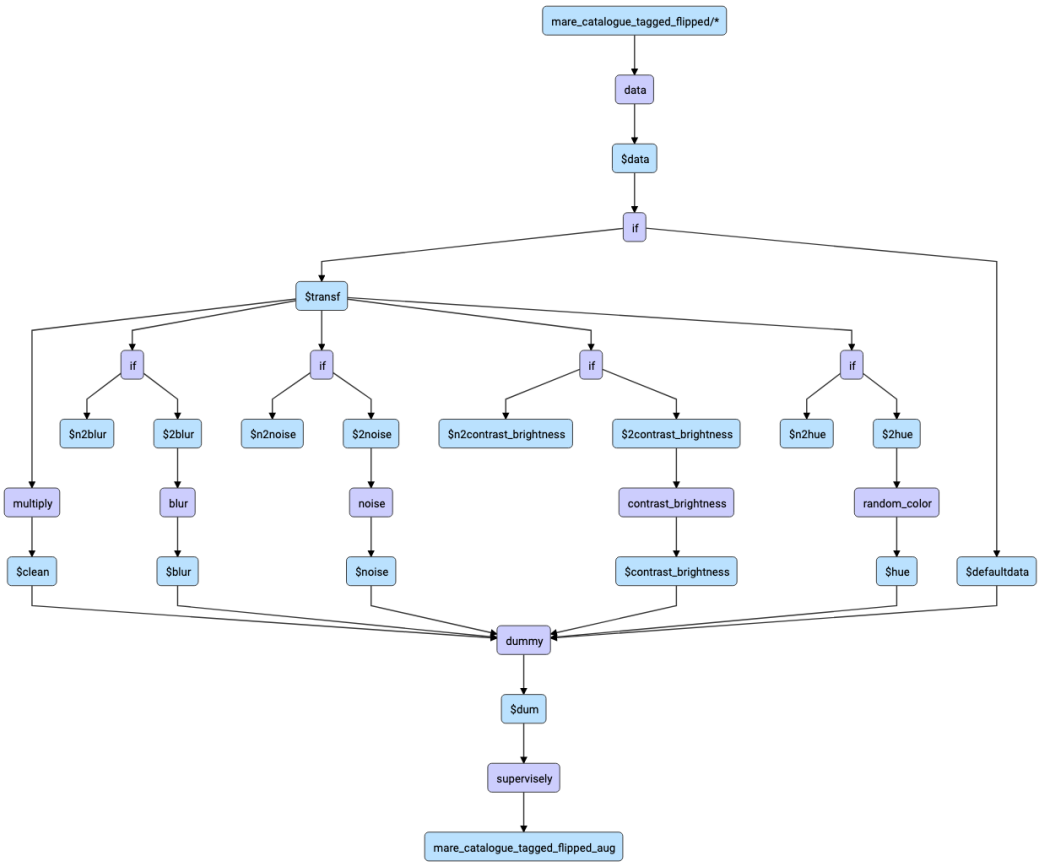


Fig. 7: Data Transformation Language (DTL) diagram showing the augmentation procedure as well as the split between the training, validation, and testing samples used during the first experiment.

Results. After 100 obtained epochs, using AlexNet and when discriminating marine mammal individuals, obtained training accuracy score was $0,9986$, while the validation accuracy score was $0,9784$ (Figure 8). Furthermore, a cross-entropy training loss score of 0,0059 was obtained, including the validation loss of 274,7850 (Figure 8). In this experiment all unique samples were used for splitting, resulting in portions for training and testing datasets during training and the validation dataset later during validation (model inference). Important to note that the dataset used for validation was also synthetic-based, due to the low amount of training imagery. Obtained results suggest that the model may discriminate marine mammal individuals, however, fails to generalize on the validation datasets.

Fig. 8: Results obtained during training and validation.

In Figure 9 it is possible to observe the confusion matrix of AlexNet individuals classification. The diagonal present on the image shows that the model is being able to predict correctly the most part of the images. This diagonal may be explained by the role of the synthetic dataset, as augmentations were performed on the dataset, increasing the small amount of data. Future work of the dissertation will perform data augmentation without color adjustments, focusing mostly on the slight angle rotations. Conversely, more training epochs and tweaking of hyperparameters will be performed, analyzing thoroughly the sensitivity, and specificity per each class.

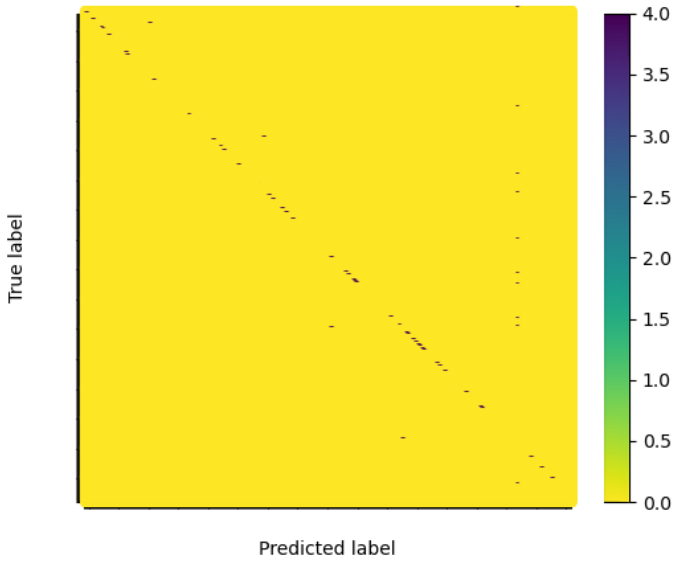


Fig. 9: Matrix resulting of testing on the classification of different individuals using AlexNet model.

3.2 Experiment 2 - Benchmarking State of the Art Neural Network Architectures to Discriminate Marine Mammal Individuals

Since Experiment 1 showcases the reasonable loss score another experiment was performed training the individual's dataset for a series of different models, like AlexNet, DenseNet121, LeNet, MobileNetV2, ResNet50, and InceptionResNetV2. The approach used these models with data augmentation for both training and validating sets.

Models. The used models for this experiment were based on both traditional shallow and deep-learning neural networks such as AlexNet, LeNet, ResNet, DenseNet, MobileNet, and InceptionResNet.

AlexNet - It's possible to see on table 3 the AlexNet model used on the experience, this one it's the same that is described in section 3.1.

LeNet - One of the models used is based on the LeNet architecture, this one contains a total of 10 layers, in table 4 it's possible to see a summary of the model architecture and identify some parameters of the LeNet architecture model. The first layer is a convolution layer with 128 filters, a ReLU activation function, and stride 1, in the second layer it was decided to have a max pooling layer with a pooling size of 2 and stride 2 instead of an average pooling layer, the difference will be that instead of calculating the average value for each patch on the feature map it will calculate the maximum. The third layer is a Dropout layer with a rate of 10% to help prevent overfitting. Then there's another convolution layer, a max pooling layer, another Dropout layer, and after a flatten layer that flattens the feature map into a vector of features. And finally, there are 2 dense layers, the first one with 100 neurons while in the last one, the number of units corresponds to the number of classes, the activation function for the output layer is a softmax activation function.

DenseNet121 - In table 7 is possible to see a summary of this model architecture used for the experiment. DenseNet121 architecture contains 1 convolution layer size 7×7, 58 convolution layers size 3×3, 61 convolution layers size 1×1, 4 average pooling layers, and 1 fully connected layer. This variant was used because it has 121 layers, fewer layers than the others variants of DenseNet.

MobileNetV2 - The MobileNet model that was used in this experiment was the MobileNetV2, which is illustrated in the table 8. It's composed of 53 convolution layers and 1 average pooling layer. This model has 2 main components, the inverted residual block, and the bottleneck residual block. Bottleneck residual block utilizes 1×1 convolutions to create a bottleneck and its use reduces the number of parameters and matrix multiplications. The inverted residual block is a type of residual block that uses an inverted structure for efficiency reasons. This inverted residual block has a narrow-wide-narrow structure with a number of channels. There are 2 types of convolution layers in this architecture, convolution layers size 1×1 and depthwise convolution layers size 3×3.

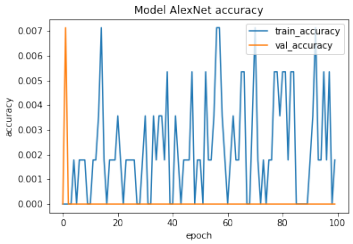
ResNet50 - The variant of ResNet used in this experiment it's the ResNet50 which has 48 convolution layers along with 1 max pooling layer and 1 average pooling layer as it's presented in table 6, having a total of 50 layers. This variant was chosen because it's a pre-trained deep learning model and provides a good starting point since the features learned on the old task are useful for the new task.

InceptionResNetV2 - InceptionResNetV2 is the variant used in this experiment and its architecture is shown in table 5. This one is a convolutional neural network that was trained on more than a million images from the ImageNet, this network is 164 layers deep and it's able to classify images into 1000 object categories.

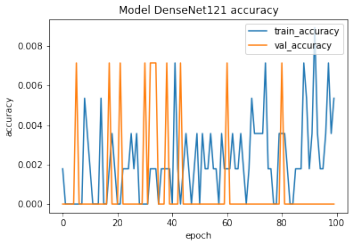
Dataset. The dataset used in this part was also obtained from MARE Insular team containing the 856 individual specimens (classes). Initially, the dataset has a total of 1138 images, where each individual has its own ID (e.g. $GMa0909$) meaning that the species belonged to the Globicephala macrorhynchus (Gma) species – pilot whales. In order to have more data and consequently have a more robust model, some augmentations were done, like for example vertical flip (making the image to be symmetrical around the vertical axis), range rotation of 20 (that means to randomly rotate the images between 0 and 20 degrees), width_shift_range (shifts the image to the left or right), height_shift_range (shifts the image vertically, up or down) and horizontal flip (making the image to be symmetrical around the horizontal axis). After the augmentation, the dataset reached a total of 70831 images, for this data augmentation it was used the class ImageDataGenerator from Keras. Also, 20% of the dataset will be used only for validation, and the other 80% will be used for training.

Results. The results of obtained 6 models based on state-of-the-art architectures in identifying marine mammal individuals are depicted in Figure 10. From obtained results, it is possible to see that the model fails to detect the marine mammal individuals. In the accuracy plot, there's a performance learning curve and in the loss plot, there's the optimization learning curve. The graphs 10a, 10b, 10c, 10d, 10e and 10f represent the accuracy of AlexNet, DenseNet121, InceptionResNetV2, LeNet, MobileNetV2 and ResNet50 models. The results show that the training and validation accuracy on these models is practically 0 if not 0. This happens since the dataset is very limited and the number of classes is large, having in total 1138 images and 856 classes.

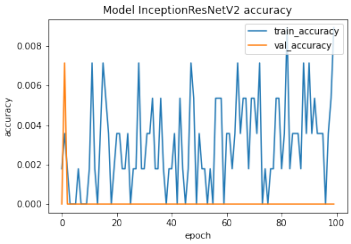
Such provides a small ratio of 1.3 images per class. In order to improve these results, it requires more images in the dataset. The graphs 10g, 10h, 10i, 10j and 10k show the losses of AlexNet, DenseNet121, InceptionResNetV2, LeNet and MobileNetV2 models, it is possible to observe that while the plot of validation loss continues to increase with training, the training loss is decreasing with time. This means that the model is overfitting, in other words, the more specialized the model becomes in the training data, the less well it is able to generalize to new data, resulting in an increase in generalization error. Graph 10l is the ResNet50 model loss, it's possible to see that while the training loss decreases with time, the validation loss increases and decreases between epochs 0 and 40 and then increases till the end. It's another case of overfitting but with the difference that in this case the validation loss is more unstable. The figures 10m, 10n, 10o, 10p, 10q and 10r represent the matrices of AlexNet, DenseNet121, InceptionResNetV2, LeNet, MobileNetV2 and ResNet50 models respectively. Results show that the model failed to predict the classes of the images, instead of having a vertical line indicating that the model predicted always the same class for all the images it should have a diagonal line, which means that this class has something that makes the model always recognize him on every image.



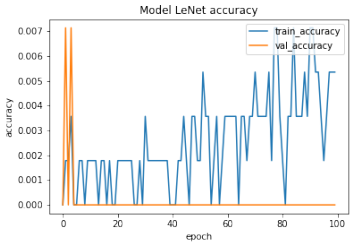
(a) AlexNet



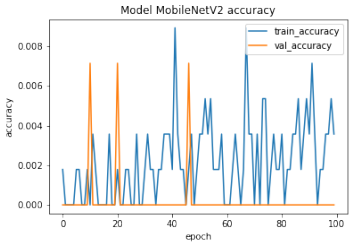
(b) DenseNet121



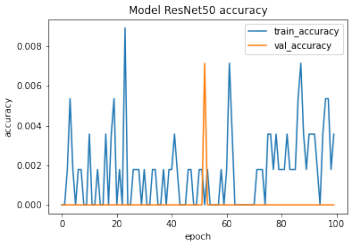
(c) InceptionResNetV2



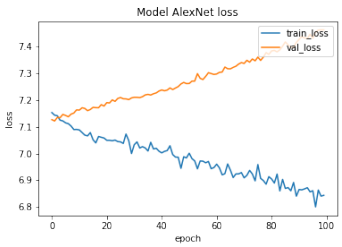
(d) LeNet



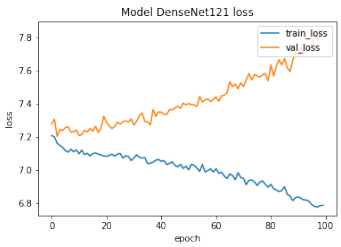
(e) MobileNetV2



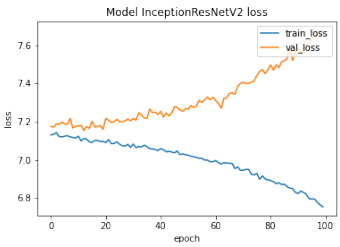
(f) ResNet50



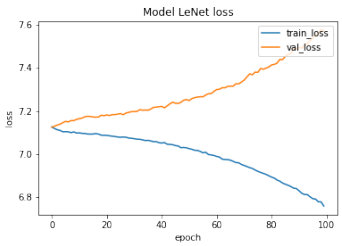
(g) AlexNet



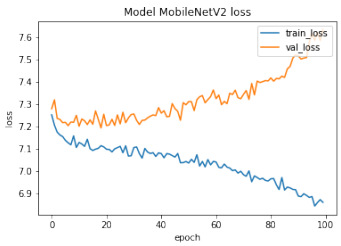
(h) DenseNet121



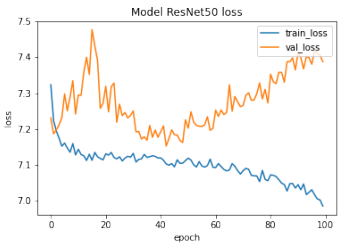
(i) InceptionResNetV2



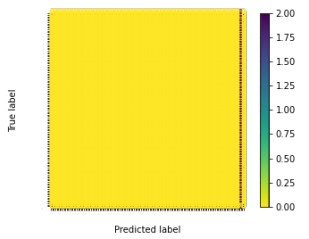
(j) LeNet



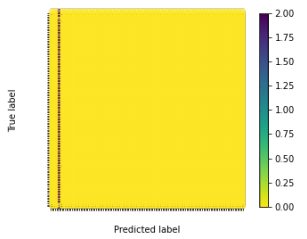
(k) MobileNetV2



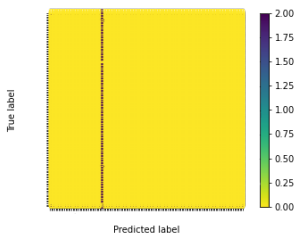
(l) ResNet50



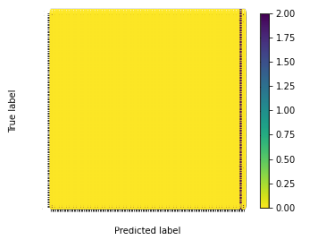
(m) AlexNet (I)



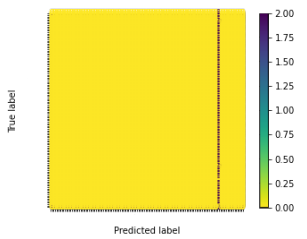
(n) DenseNet121 (I)



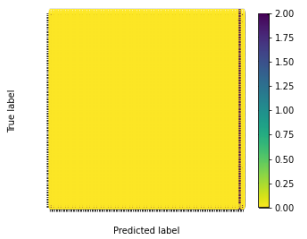
(o) InceptionResNetV2 (I)



(p) LeNet (I)



(q) MobileNetV2 (I)



(r) ResNet50 (I)

3.3 Experiment 3 - Benchmarking State of the Art Neural Networks to Discriminate other Features

In this experiment it was decided instead of classifying the individuals, to classify the dataset in the following categories, roundness, wideness, nicks, and scratches using the same models that were utilized in the previous experiment reducing the huge number of classes that previously existed. For this to be possible it was necessary to do manual labeling of each dataset.

Models. Same experimental setup that was used in the previous experiment, but for different purposes.

Dataset.

Roundness. The dataset used is the same that was used previously but this time for roundness. The roundness category is a binary classification, the dorsal fin can be rounded or pointed. There is a total of 624 images of class rounded and 514 images of class pointed.

Wideness. Same dataset here but for wideness. The wideness category is also binary, being the 2 existing classes wide and narrow. This dataset has 774 wide dorsal fins and 364 narrow dorsal fins.

Nicks. The same previous dataset but now for nicks. On the nicks category exists 7 classes depending on the number of nicks present in the dorsal fin, the class 1N represents the dorsal fins with 1 nick, 2N represents the ones with 2 nicks, 3N represents 3, 4N is 4 and +4N is the ones with 5 or more nicks. Class WN represents the dorsal fins without nicks and class MAD represents all the dorsal fins with malformations on the anterior side of the dorsal fin. In class 1N there is a total of 137 images, in class 2N 215 images, in class 3N has 209 images, on 4N there are 137 images and +4N has a total of 197 images. WN possesses a total of 8 images and class MAD has a total of 235 images.

Scratches. Again the same dataset but for scratches. Here, there are 39 classes and the class is defined based on the number of scratches that are able to be identified on the dorsal fin.

Classification. In the next section the image in figure 11 will be analyzed and understood how it's classified depending on the category in question.

Fig. 11: Image chosen to explain what are the procedures to classify the image in each different category.

Roundness. Looking at the roundness category, it's possible to see that the image in figure 12 belongs to the class rounded. The dorsal fin is classified as rounded because when drawing a red circle inside the fin it's possible to see that it fits perfectly. Otherwise, the dorsal fin it's considered pointed.

Fig. 12: Rounded dorsal fin.

Wideness. On the wideness category, observing the figure 13 it's possible to see that the dorsal fin is wide at the bottom and the length of the axis x is bigger than the length of axis y. Therefore this dorsal fin is classified as wide, otherwise, if the length of axis y was bigger than the length of axis x the dorsal fin would be considered to belong to the class narrow.

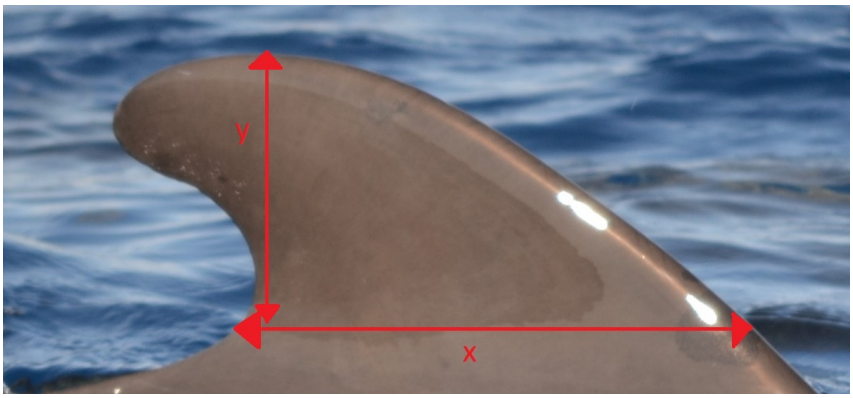


Fig. 13: Wide dorsal fin.

Scratches. Observing the image shown in figure 14, it's possible to identify 7 red boxes. Each box has one or more scratches inside, and in total there are 10 scratches on this image, so the image is classified as scratch 10.

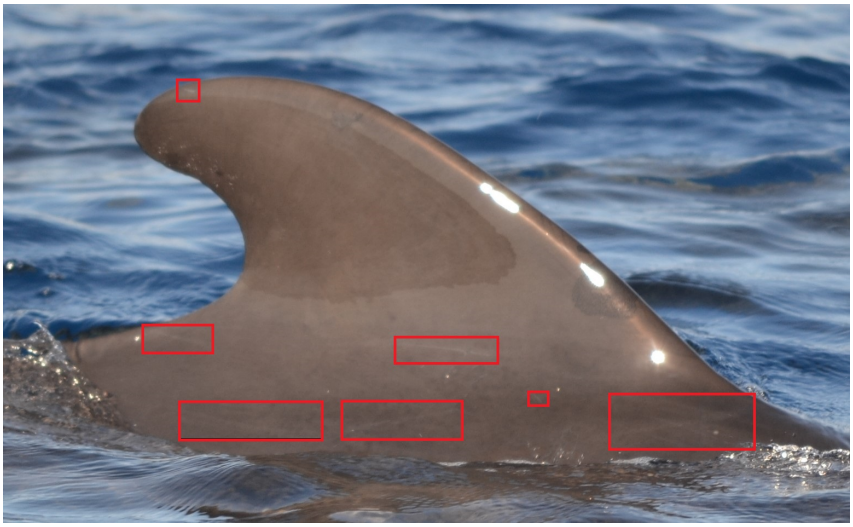


Fig. 14: Whale with 10 scratches present on the image.

Nicks. In the image of figure 15, it's possible to watch that this dorsal fin has only 1 nick, so this image belongs to class 1N. Each time it is possible to identify a small cut in the dorsal fin like the one inside the red box of the image, it counts as one more nick in the classification of the category.

Fig. 15: Dorsal fin with 1 nick.

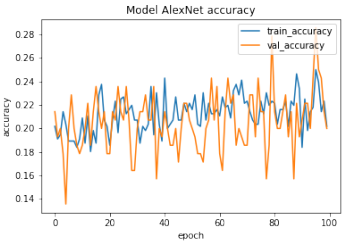
Results.

Nicks. On the nicks category are obtained results, having the 7 classes indicating the needed effort of the models not as complex as the previous experiment.
The graph 16a shows the accuracy obtained by the AlexNet model. In the validation accuracy, there is a variation between 13% and 28%(Figure 16a). In the training accuracy, the variation is between 18% and 25%(Figure 16a). The training and validation do not diverge too much, meaning the model can generalize well but not improve over the epochs. The graph 16b shows the accuracy of DenseNet121 model, it is possible to see the training accuracy increasing from 16% to 84%(Figure 16b), which is a sign that the model is learning. But validation accuracy stays more and less the same, ranging between 13% and 28%(Figure 16b). This might be a sign of overfitting, showing that the model doesn't generalize.
The graph 16c represents the accuracy of InceptionResNetV2 model, practically the same thing happens here as in the previous graph, meaning this model also fails to generalize. The training accuracy increased, ranging from 18% to a maximum of 66%(Figure 16c) and on the other hand, the validation accuracy stayed more and less the same, ranging between 11% and 32%(Figure 16c).
In the graph 16d it's possible to see the accuracy of LeNet model, this situation is similar to the one in graph 16a with the model showing that it can generalize well but with the difference that has an improvement over the epochs.
The accuracy of MobileNetV2 model is shown in graph 16e, here it's the same case as on graphs 16b and 16c with the training accuracy increasing over the epochs but reaching a maximum of 86% and the validation accuracy ranging between 11% and 25%(Figure 16e). The accuracy graph of ResNet50 model 16f is the same case as well the previous ones but with the training accuracy reaching a maximum of 92%. In the graph 16g there's the AlexNet loss, in this case, it was identified as a good fit, the goal of the learning algorithms, this one is identified by a training and validation loss that decreases to a point of stability with a very small gap between them, which is the case in this situation. The loss of the DenseNet121 model shown in graph 16h allows seeing that

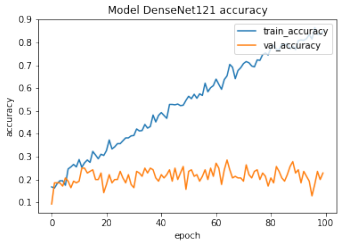
while training loss decreases the validation loss increases, meaning that the model is overfitting. What is kind of what happened on some graphs of experiment 2, so basically the model is not able to generalize to new data. This model in the end reaches a training loss of 44%(Figure 16h). This situation also happens in the loss graph of InceptionResNetV2 16i, in the MobileNetV2 loss graph 16k, and also in the ResNet50 loss graph 16l but in this case, the model reaches a training loss of 23%.

The LeNet loss graph 16j shows the decrease of the training and validation loss. Here it is possible to have an unrepresentative train dataset, meaning that the training dataset does not provide enough information to learn the problem. This situation occurs when the training dataset has too few examples compared to the validation dataset. It is possible to identify this situation when there is a large gap between the curves of validation and training loss and both show improvement over time. Figure 16m represents the matrix of the AlexNet model, showing that this model did not have a very good performance as it classifies correctly 23 images, from which 22 belong to class 2N and 1 belongs to class 3N. The DenseNet121 matrix 16n shows better results in terms of recognizing different classes. It predicted correctly 3 images of class MAD, 2 images of class 1N, 8 images of 2N, 2 images of class 3N, 4 images of 4N, and 6 images of class +4N. The InceptionResNetV2 matrix 16o predicted correctly 9 images of class MAD, 1 image of class 2N, 5 images of class 3N, 2 images of class 4N, and 5 images of class +4N.

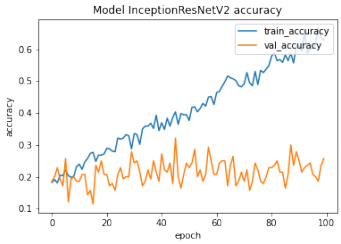
The figures 16p, 16q and 16r represent the matrices of LeNet, MobileNetV2 and ResNet50 models. The LeNet model solely predicted correctly 2 classes, class 2N with 18 images and 3N with 5 images, while the other 2 (MobileNetV2 and ResNet50 models) have more classes predicted correctly.



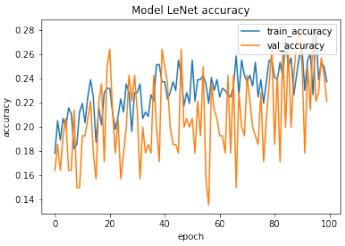
(a) AlexNet



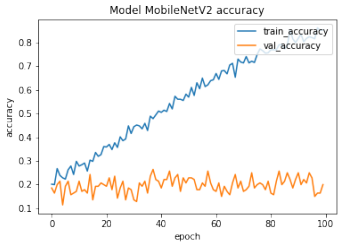
(b) DenseNet121



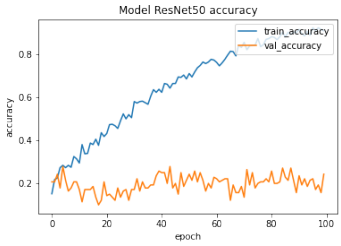
(c) InceptionResNetV2



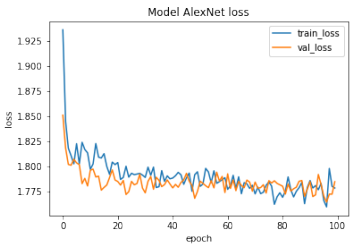
(d) LeNet



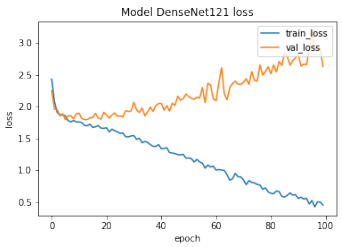
(e) MobileNetV2



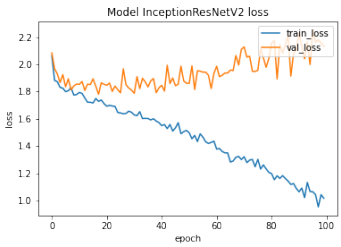
(f) ResNet50



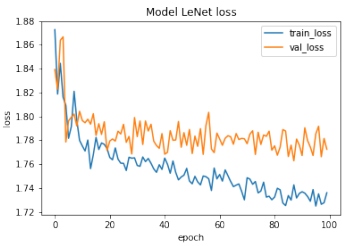
(g) AlexNet



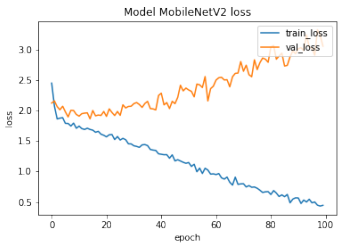
(h) DenseNet121



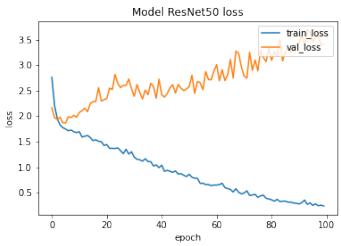
(i) InceptionResNetV2



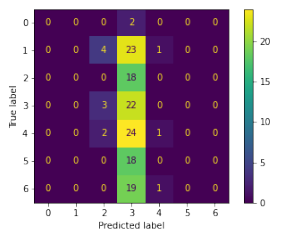
(j) LeNet



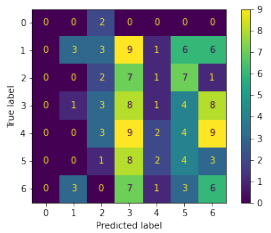
(k) MobileNetV2



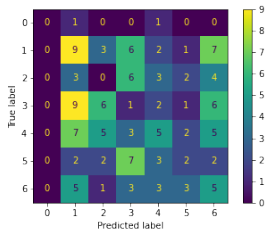
(l) ResNet50



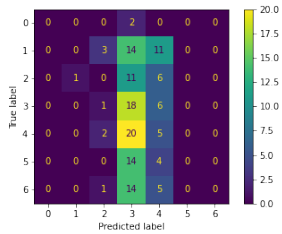
(m) AlexNet (N)



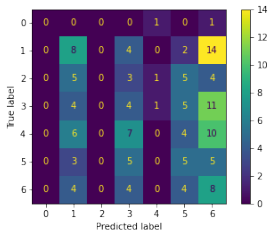
(n) DenseNet121 (N)



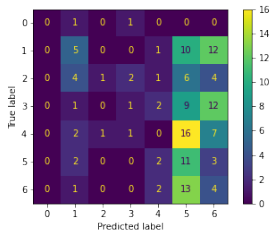
(o) InceptionResNetV2 (N)



(p) LeNet (N)



(q) MobileNetV2 (N)



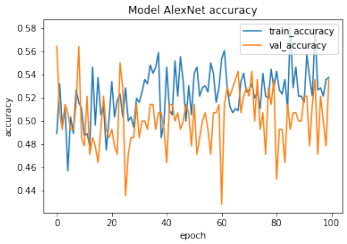
(r) ResNet50 (N)

Roundness. On the roundness category, the number of classes is 2 expecting the effort by the models to be less challenging compared to the previous experiment. The graph 17a shows the accuracy obtained by the AlexNet model. It is possible to see that in the training accuracy the variation is between 47% and 58%. The training and validation do not diverge too much from what is good, meaning the model can generalize but is not perfect and there's also a slight improvement in the training accuracy over the epochs. In the accuracy graph of DenseNet121 17b it's possible to see the training accuracy increasing from 50% to 93%. While the validation accuracy only ranges between 48% and 64%. Looking at the learning curves of both accuracy it can be said that the model fails to generalize. In the accuracy graph of InceptionResNetV2 17c occurs the same as the last one but the gap here is smaller. The training accuracy goes from 51% to a maximum of 79%, while validation accuracy oscillates between 44% and 62%.

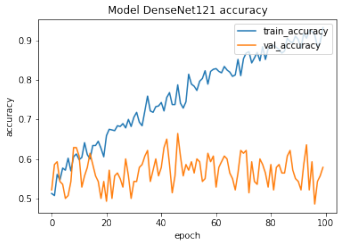
LeNet accuracy graph 17d shows that there is a little improvement in training. This does not happen on the validation sets, as there is also a gap between them as it's possible to identify, indicating issues for models to generalize. The graphs 17e and 17f represent the accuracy of MobileNetV2 and ResNet50 and they are very similar because in both the training increases and the validation doesn't. The ResNet50 training even reaches 94% accuracy, but the 2 models fail to generalize like the others. The graph 17g, which represents AlexNet loss shows that training and validation tend to have close values as the training reduces and validation stabilizes, which means a good fit, and the model is able to generalize well. On the other hand, in the DenseNet loss graph 17h, is possible to identify the increase of the validation loss and the decrease of training loss reaching a value of 18%. This means the model is overfitting and is not able to generalize to new data.

The situation on the loss graph of InceptionResNetV2 17i is more and less the same as the previous one, the validation increases while the training decreases indicating the models also fail to generalize but the gap between training and validation is smaller.

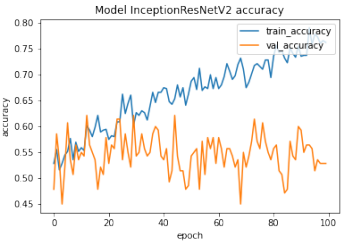
The LeNet loss graph 17j on other hand shows that initially the training and validation loss decrease but then they both stabilize, this is a case of unrepresentative training dataset. Has was already mentioned, this means that the training dataset does not provide sufficient information to learn the problem compared with the validation dataset used. The graphs of MobileNetV2 17k and ResNet50 17l both indicate that they are overfitting and fail to generalize like some previous examples in this experiment. The MobileNetV2 model was able to reach a training loss of 24% and the ResNet50 model reached 13%. The figures 17m, 17n, 17o, 17p, 17q and 17r represent the matrices of each different model of this experiment. The DenseNet121 model was the one that predicted correctly more images, predicting a total of 91 images, 79 images of class rounded and 12 images of class pointed. The 2 models that predicted fewer images correctly were the MobileNetV2 and the ResNet50 models, both predicting 77 images correctly each. MobileNet predicted 69 images of class rounded and 8 images of class pointed while the ResNet model predicted 58 images of class rounded and 19 images of class pointed.



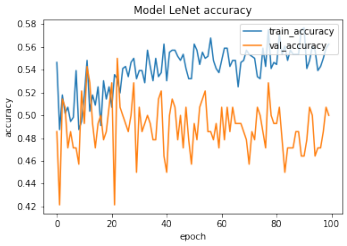
(a) AlexNet



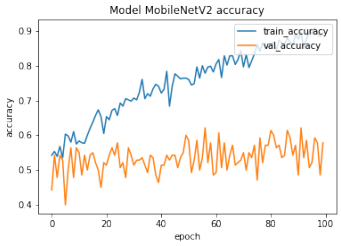
(b) DenseNet121



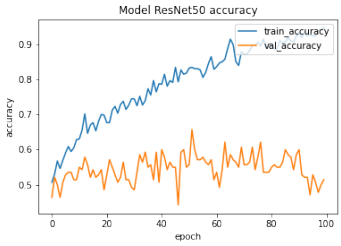
(c) InceptionResNetV2



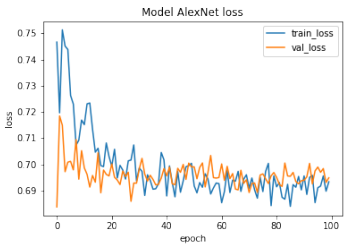
(d) LeNet



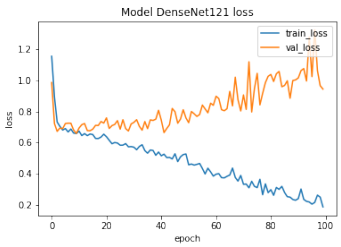
(e) MobileNetV2



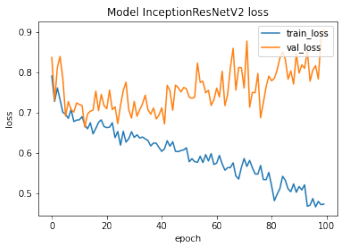
(f) ResNet50



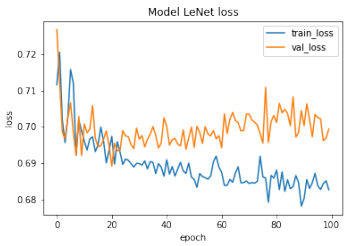
(g) AlexNet



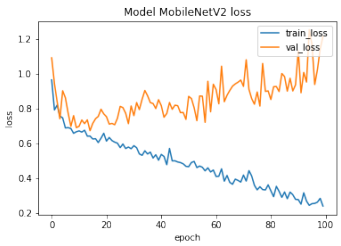
(h) DenseNet121



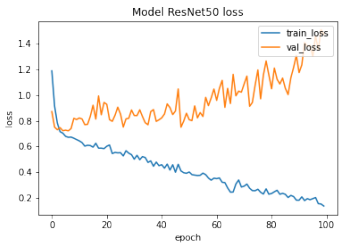
(i) InceptionResNetV2



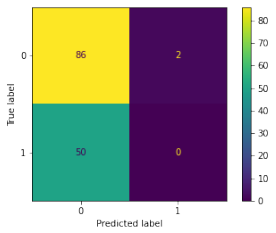
(j) LeNet



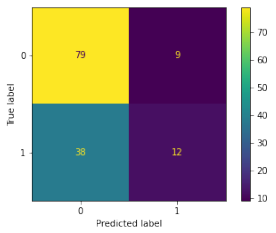
(k) MobileNetV2



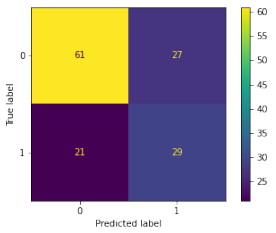
(l) ResNet50



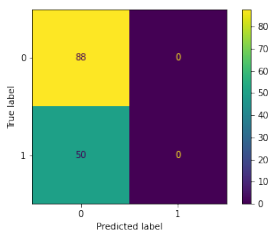
(m) AlexNet (R)



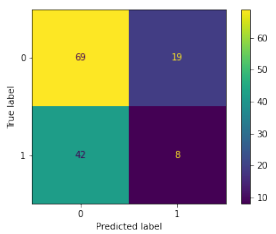
(n) DenseNet121 (R)



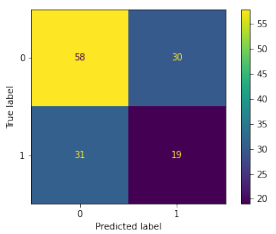
(o) InceptionResNetV2 (R)



(p) LeNet (R)



(q) MobileNetV2 (R)

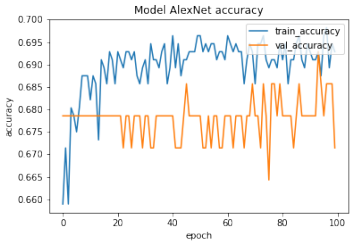


(r) ResNet50 (R)

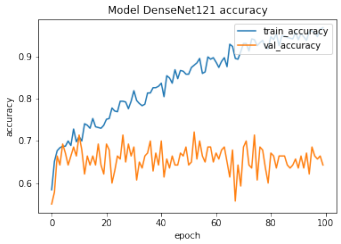
Wideness. On the wideness category the obtained results are below, here the number of classes is 2, like the previous one. The graph of AlexNet accuracy 18a shows an improvement in training but that doesn't happen on the validation side, it's possible to identify a gap between them, which indicates the model doesn't generalize well. In the DenseNet121 accuracy graph 18b it's possible to see the training accuracy increasing and reaching 96%. The validation accuracy just ranged between 55% and 72%. Looking at the gap between training and validation it's possible to say that this model fails to generalize, this also happens in the MobileNetV2 accuracy graph 18e, in the ResNet50 accuracy graph 18f and also in the InceptionResNetV2 graph 18c despite the gap between validation and training being lower as it's possible to see below. At one point ResNet model is even able to reach a training accuracy of 97%.

The LeNet accuracy graph 18d shows that training and validation do not diverge too much, which means the model can generalize well but not perfectly and there's a slight improvement of the training accuracy while the validation accuracy constantly increases and decreases keeping within a certain range of values. Graphs 18h, 18i, 18k and 18l who represent the losses of DenseNet121, InceptionResNetV2, MobileNetV2, and ResNet50 show the decreasing of training loss and the increase of validation loss, what means that they are overfitting and also fail to generalize. The graphs of AlexNet 18g and LeNet 18j losses are kind of similar, on both of them, there's an unrepresentative training dataset, which means as previously said the training dataset has too few examples as compared to the validation dataset. The AlexNet validation's loss ranges between 62% and 65%, and the training loss between 60% and 66%. The validation loss of Lenet ranges between 61% and 67%, while the training loss ranges between 58% and 64%.

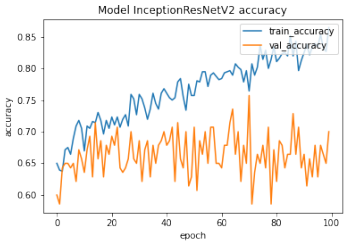
Like the previous example, the graphs 18m, 18n, 18o, 18p, 18q and 18r represent the matrices of each different model. In this category the model that predicted correctly more images was the InceptionResNetV2 model, more precisely 98 images, 85 belonging to the class wide and the others 13 of class narrow. The second model with more images predicted correctly was DenseNet121 with 93 images, 70 class wide and the remaining class narrow. The 2 models with fewer images predicted correctly were the AlexNet and LeNet, curiously in the 2 cases, all 88 images belong to the wide class.



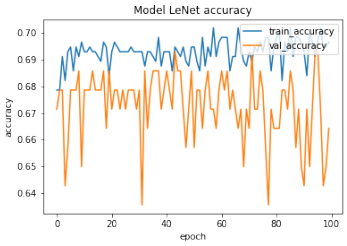
(a) AlexNet



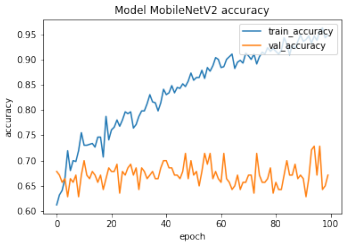
(b) DenseNet121



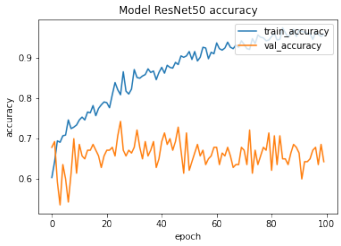
(c) InceptionResNetV2



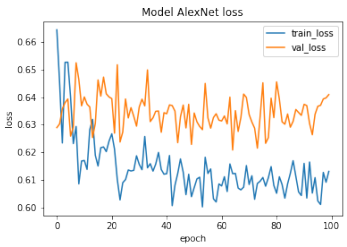
(d) LeNet



(e) MobileNetV2

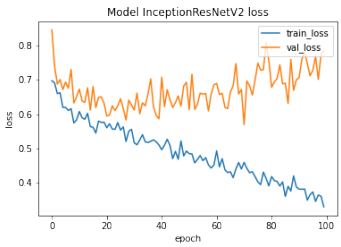


(f) ResNet50

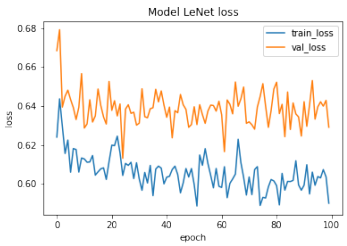


(g) AlexNet

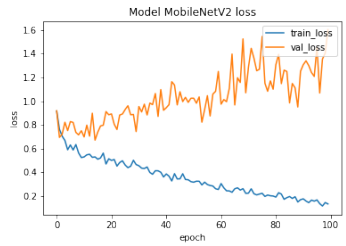
(h) DenseNet121



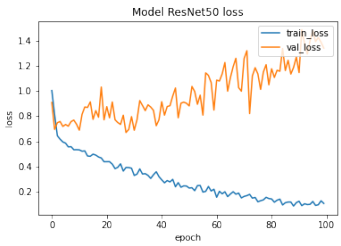
(i) InceptionResNetV2



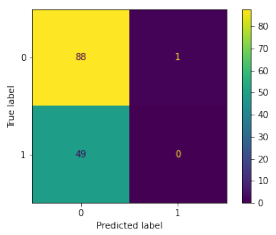
(j) LeNet



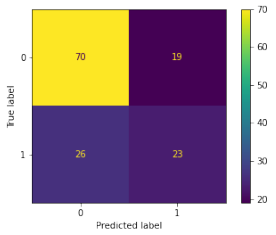
(k) MobileNetV2



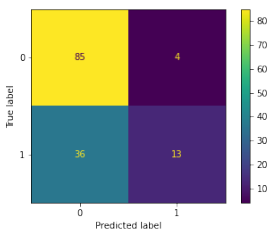
(l) ResNet50



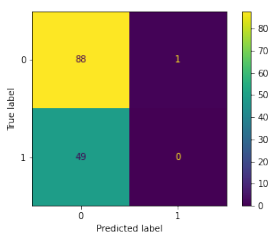
(m) AlexNet (W)



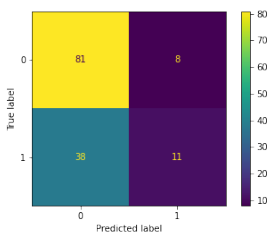
(n) DenseNet121 (W)



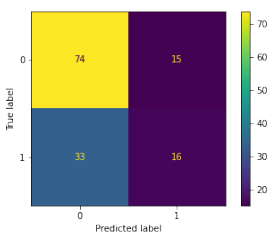
(o) InceptionResNetV2 (W)



(p) LeNet (W)



(q) MobileNetV2 (W)



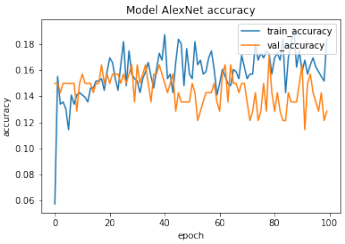
(r) ResNet50 (W)

Scratches. On the scratches category it was obtained the results below, here the number of classes is 39, a higher number than the previous categories, which should make the task more difficult. Starting with graph 19a, it's possible to see that practically there isn't a gap between the training and validation accuracy of the AlexNet model. Meaning that this model is generalizing well, although the training accuracy only reaches a maximum of 19% and the validation accuracy a maximum of 17%, values that demonstrate a poor performance of the model.

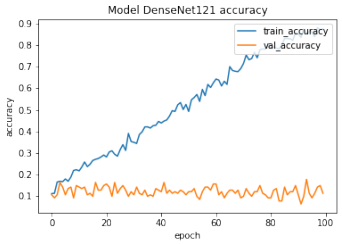
The graphs 19b, 19c, 19e and 19f that correspond to the accuracy models of DenseNet121, InceptionResNetV2, MobileNetV2, and ResNet50 are very similar regarding the point that the training accuracy increases over the epochs while the validation accuracy stays in a certain range of values. The ResNet model is the one that reaches the higher value of training accuracy being 95%, the MobileNet model reaches a value of 92%, DenseNet reaches 88% and InceptionResNet only reaches 64%. The validation accuracy on these 4 models oscillates between 5% and 18% but not increasing like the training. The gap between training and validation tells that these models fail to generalize like it was seen in previous cases. The LeNet plot 19d also fails to generalize due to the gap between the 2 variables, but the training accuracy, in this case, doesn't improve much over the epochs only reaching a max of 20%. In the AlexNet loss plot 19g, is possible to see that training and validation loss go down together without a gap between them, which indicates that this one is a good fit although the values of both are still too high. This model also generalizes well but if the training is continued, it will likely lead to an overfit.

The DenseNet 19h, the InceptionResNet 19i, the MobileNet 19k, and the ResNet 19l loss plots, what these 4 graphics have in common is the fact that the training loss decreases as it should over the epochs while the validation loss increases slightly. As was seen in previous examples this indicates that the model is overfitting and isn't able to generalize to new data. Of these 4, the ResNet model had the best performance as it reached a training loss of 20%, the MobileNet model reached 30% and the DenseNet121 reached 43%. The LeNet graph 19j in turn, shows a decrease in both training and validation with a gap between them, this is a sign of an unrepresentative training dataset so it's known that the training dataset does not provide sufficient information to learn the problem, compared with the validation dataset used.

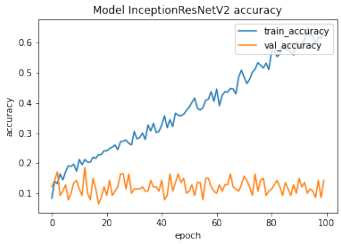
The remaining figures represent the matrices of the different models, looking to figure 19m it's possible to observe the matrix of the AlexNet model. Here the model predicted correctly 6 images of class 2 (meaning images with 2 scratches) and 5 images of class 4 (images with 4 scratches). But the rest of the predictions are all wrong, instead of the 39 existing classes, in this case, there are just 23 classes because in this model only the 23 classes present in the figure were predicted. Each matrix has a different number of classes, for example, the DenseNet matrix 19n has 24 predicted classes. The model predicted correctly 8 images of class 0 (without scratches), 6 images of class 1 (with 1 scratch), 1 image of class 2 (with 2 scratches), 1 image of class 3 (with 3 scratches), 5 images of class 4 (with 4 scratches), 2 images of class 5 (with 5 scratches), 1 image of class 6 (with 6 scratches) and 2 images of class 10 (with 10 scratches). The rest of the predictions in this model don't correspond to the true label. MobileNetV2 matrix 19q for example has more classes predicted than the other, 26 classes to be more precise. The others matrices have between 23 and 24 of the number of classes.



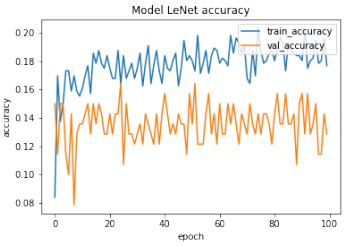
(a) AlexNet



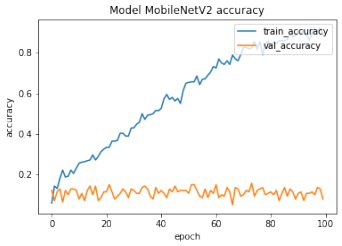
(b) DenseNet121



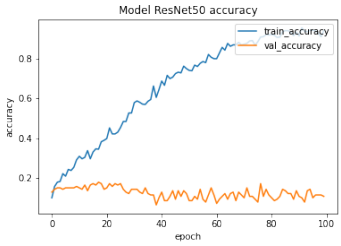
(c) InceptionResNetV2



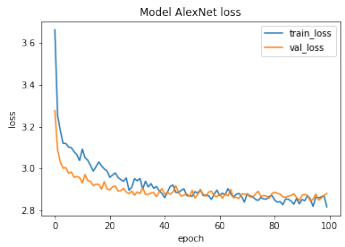
(d) LeNet



(e) MobileNetV2

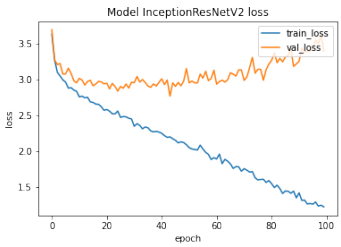


(f) ResNet50

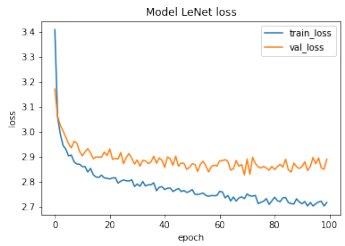


(g) AlexNet

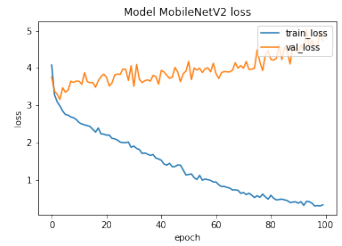
(h) DenseNet121



(i) InceptionResNetV2

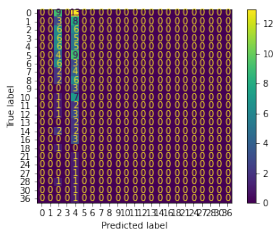


(j) LeNet

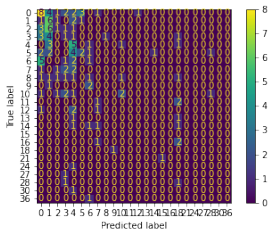


(k) MobileNetV2

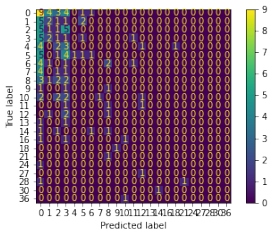
(l) ResNet50



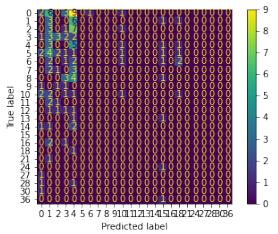
(m) AlexNet (S)



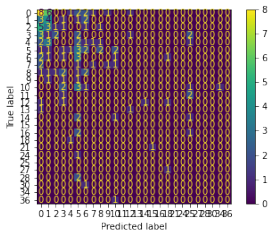
(n) DenseNet121 (S)



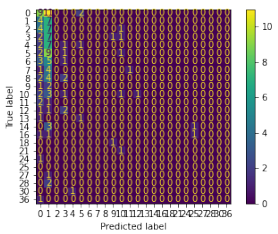
(o) InceptionResNetV2 (S)



(p) LeNet (S)



(q) MobileNetV2 (S)



(r) ResNet50 (S)

4 Discussion

4.1 Findings Interpretation

After analyzing the results of these experiments it is possible to say the following about each conducted experiment.

4.1.1 Experiment 1 - The Role of Augmented Dataset

This experiment shows that when a model needs to learn that many classes (more precisely 856), a lot of images per individual are needed to train the model (AlexNet), basically there's a need to do a lot of augmentations, and that will transform the dataset into a synthetic dataset, what is a challenge because the synthetic dataset will teach the model to recognize features that don't occur on the reality. The AlexNet model used in the experiment is very simple because it has a small number of hidden layers and neurons compared with some much more complex models like the MobileNet model used in experiment 2 for example. The shallow deep learning neural network is, therefore, capable to be used to identify the features from the dorsal fins. This has also been validated by a study where was developed an application that uses a series of neural networks to automatically locate dorsal fins in images and match them to an existing photograph catalog. In this study, they were capable to reach an accuracy of 88% and an accuracy of 94% in the 10 top-ranked matches, which are very good results in the process of image recognition [11]. Another study that also validates this experiment consisted of the development of a program named Finscan that compares the new photographic images with the ones previously collected on the database and does the matching based on the pattern of nicks and notches found on the trailing edge of dorsal fins of whales and dolphins. It's possible to say that this study was very successful because they manage to reach an accuracy of 75% within the first 4 suggested matches [54].

4.1.2 Experiment 2 - Benchmarking State of the Art Neural Network Architectures to Discriminate Marine Mammal Individuals

From experiment 2, here the augmentations realized are considered less synthetic because the operations performed are just flips and rotations, there are no augmentations like Gaussian blur, noise, or hue saturation like in the previous experiment. So fewer augmentations are performed but it seems that augmentations that play with the pixels of the images like blur, noise, and others contribute to increasing the training and validation accuracy and perform more robustly. The differences between the matrix in experiment 1 and the matrices in this one show that the performances of the models, in this case, are much lower than expected, probably due to the fact that there are too few images per class, which complicates the classification task. Looking at the matrices shows that each model predicted the same class for every image presented. So, in conclusion, the model predicted wrongly the images, it should be possible to see a diagonal case if the prediction were right, which isn't the case here. Also, it's possible to say that even more complex neural networks were not able to perform well, getting weak results like the more simple neural networks. It's possible to reiterate here that more input data (images) would probably improve the performance of the models because there would be more images per class, which is one of the great challenges in this experience.

4.1.3 Experiment 3 - Benchmarking State of the Art Neural Network Architectures to Discriminate Other Features

In experiment 3, it's possible to say that the fact that each category has fewer classes compared with experiment 1 and experiment 2, helps on the performance side since there are more images per class in this case. The augmentations done in this experiment are the same as the ones done in experiment 2, that is, flips, rotations, and shifting the images to the left or right and up or down. The performances of each category were definitely better than the previous 2 experiments. For example, although some models like AlexNet and LeNet in the roundness category only reach 60% in the training accuracy, these are much better results than any model of experiment 2 obtained. The combination of each output category will probably make a more robust model because by crossing predictions between the different categories it is possible to reduce the number of hypotheses turning the model more accurate in the best case.

4.2 Research Contributions

This study confirms that with data augmentation is possible to obtain even better results due to the fact that the model is feed-ed with more data, which is crucial to improve the performance of the models as other studies have already shown. It also confirms that deep neural networks nowadays are a very powerful tool that can be used in the recognition of individuals. And this is mainly due to the development of the neural networks that have been carried out, which are increasingly complex and with more layers like the Inception with the introduction of the inception modules or the ResNet model with 50 weight layers. It also brings up the possibility of combining different features like nicks, scratches, roundness, and wideness, which can reach interesting and better results than the previous approaches.

4.3 Research Challenges

One of the challenges is that there are few datasets available on the pilot whale specie, it is necessary to collect more images on the field which is complicated because experts have to take trips on boats and wait for the sighting of the species, in addition to having to capture the precise moment of the sighting, which normally only lasts a few seconds. Another challenge was the fact that the dorsal fins of whales change over time. For example, it's normal a dorsal fin to gain more scars or more bites due to clashes with other animals and that will make the identification process more complicated since the model will see features that didn't exist on the individual before which can lead to mistakes in the classification. To perform classification with so many classes on such complex models it's necessary to have very advanced and expensive hardware, a mid-range GPU cannot perform these classifications in such a short amount of time. So in order to realize this experiment it was necessary to use a powerful GPU like the NVIDIA RTX A6000 which possesses 48 GB of GPU memory and a power consumption of 300 watts.

4.4 Guidelines for Applying CNNs to identify Marine Mammal Individuals

Based on previous findings, the dissertation next provides the steps paving the way toward the usage CNNs when classifying marine mammal individuals.

1. **In-situ Dataset Enrichment.** When obtaining the photographs from the field, it is important to obtain as many as possible images from marine mammal individuals. For instance, when taking a photo of the dorsal fin, it is possible to take shutter photographs - e.g. 100 photographs of the same frame with minor differences. Such can enrich the dataset and allow the clear contrast and identification of features for the neural networks.

2. **Proper data augmentation.** Based on the experiments, it is important to select the types of augmentations which do not skew the nature of realistic images. That is, augmentations like vertical flip allow the species to be seen from both sides, whereas augmentation such as color hue contributes to the increase of color saturation, which is typically not found on the surface during photo-gathering and in photo-identification.

3. **Selection of specific features.** To tackle the challenge of recognizing marine mammal individuals, the dissertation proposes a focus on distinctive features, such as the number of nicks, scratches, and the roundness and wideness of the fin. This is doable for the specific type of taxa, e.g. pilot whales, and such clearly may not work well with other Odontoceti species such as sperm whales, as their dorsal fins are becoming underdeveloped.

4.5 Conclusion

The dissertation challenged the detection of marine mammal individuals which remains an open challenge. Convolutional neural network models were trained based on photo identification and dorsal fin features. The dissertation provided three experimental validations, where the first focused on the role of the synthetic dataset in model performance based on shallow deep-learning neural networks. The second one benchmarked the state-of-the-art convolutional neural networks, showing the challenges for such in identifying marine mammal individuals. The third showed the approach by identifying other features (nicks, roundness, wideness, and scratches), suggesting future experiments in combining their outputs for decision-making trees. All three experiments point to the need for more datasets from the field, which remains challenging to obtain, suggesting the resort to careful data augmentation. Such may further improve the model performance, and reduce the costs of acquiring the datasets. In short, it was possible to use CNNs to identify individuals of marine mammals, in a brief analysis, the results obtained in experiment 1 were quite satisfactory taking into account the number of classes, but in experiment 2 the results were beyond disappointing were the worst of the 3 experiences. While the results in experiment 3 were more encouraging as it was possible to observe that in the categories with fewer classes, there was a better performance by the models but in the categories with more classes, the performance was worse, and of course that the possibility of using a decision tree in future on this experiment it's interesting and may help the improvement of performances.

4.6 Future Work

For the next steps, the dissertation proposes the usage of trees to vote on obtained classes from different models (wideness, roundness, nicks, and scratches), selecting the individual with the most votes. The dissertation also proposes the usage of Generative Adversarial Networks (GANs) [55], using a generator and discriminator to perform data augmentation in order to have larger samples of imagery and thus training models with improved performance since the reduced number of images available is one of the greatest limitations in this study. In theory, such will also allow the generated images to be more reliable compared to the images obtained through traditional augmentation methods. Additional future work includes the creation and comparison of more shallow deep learning neural networks, and deeper analysis by comparing the applied filters in the obtained feature maps after each convolution layer.

References

[1] E. Hoyt, "Whale watching," in *Encyclopedia of marine mammals*. Elsevier, 2009, pp. 1223–1227.

[2] M. B. Orams, "Tourists getting close to whales, is it what whale-watching is all about?" *Tourism management*, vol. 21, no. 6, pp. 561–569, 2000.

[3] A. M. Cisneros-Montemayor, U. R. Sumaila, K. Kaschner, and D. Pauly, "The global potential for whale watching," *Marine Policy*, vol. 34, no. 6, pp. 1273–1278, 2010.

[4] C. Khan, D. Blount, J. Parham, J. Holmberg, P. Hamilton, C. Charlton, F. Christiansen, D. Johnston, W. Rayment, S. Dawson *et al.*, "Artificial intelligence for right whale photo identification: from data science competition to worldwide collaboration," *Mammalian Biology*, pp. 1–18, 2022.

[5] C. Gomez-Salazar, F. Trujillo, and H. Whitehead, "Photo-identification: A reliable and non-invasive tool for studying pink river dolphins (inia geoffrensis)," *Aquatic Mammals*, vol. 37, no. 4, pp. 472–485, 2011.

[6] B. Wiirsig and T. A. Jefferson, "Methods of photo-identification for small cetaceans," *to Estimate Population Parameters*, p. 43, 1974.

[7] A. Gilman, T. Dong, K. Hupman, K. Stockin, and M. Pawley, "Dolphin fin pose correction using icp in application to photo-identification," in *2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*. IEEE, 2013, pp. 388–393.

[8] R. Maglietta, R. Carlucci, C. Fanizza, and G. Dimauro, "Machine learning and image processing methods for cetacean photo identification: A systematic review," *IEEE Access*, 2022.

[9] F. Alves, S. Quérouil, A. Dinis, C. Nicolau, C. Ribeiro, L. Freitas, M. Kaufmann, and C. Fortuna, "Population structure of short-finned pilot whales in the oceanic archipelago of madeira based on photo-identification and genetic analyses: implications for conservation," *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 23, no. 5, pp. 758–776, 2013. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aqc.2332

[10] E. Guirado, S. Tabik, M. L. Rivas, D. Alcaraz-Segura, and F. Herrera, "Whale counting in satellite and aerial images with deep learning," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[11] J. W. Thompson, V. H. Zero, L. H. Schwacke, T. R. Speakman, B. M. Quigley, J. S. Morey, and T. L. McDonald, "finfindr: Automated recognition and identification of marine mammal dorsal fins using residual convolutional neural networks," *Marine Mammal Science*, vol. 38, no. 1, pp. 139–150, 2022.

[12] V. Renò, G. Losapio, F. Forenza, T. Politi, E. Stella, C. Fanizza, K. Hartman, R. Carlucci, G. Dimauro, and R. Maglietta, "Combined color semantics and deep learning for the automatic detection of dolphin dorsal fins," *Electronics*, vol. 9, no. 5, p. 758, 2020.

[13] R. Bogucki, M. Cygan, C. B. Khan, M. Klimek, J. K. Milczek, and M. Mucha, "Applying deep learning to right whale photo identification," *Conservation Biology*, vol. 33, no. 3, pp. 676–684, 2019.

[14] V. Renò, G. Dimauro, C. Fanizza, R. Carlucci, and R. Maglietta, "Computer vision and deep learning applied to the photo-identification of cetaceans," in *Measurement for the Sea*. Springer, 2022, pp. 291–308.

[15] A. Polzounov, I. Terpugova, D. Skiparis, and A. Mihai, "Right whale recognition using convolutional neural networks," *arXiv preprint arXiv:1604.05605*, 2016.

[16] H.-W. Hsu, Y.-C. Lee, J.-J. Ding, and R. Y. Chang, "Dolphin recognition with adaptive hybrid saliency detection for deep learning based on densenet recognition," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE, 2018, pp. 455–458.

[17] C. Bergler, A. Gebhard, J. R. Towers, L. Butyrev, G. J. Sutton, T. J. Shaw, A. Maier, and E. Nöth, "Fin-print a fully-automated multi-stage deep-learning-based framework for the individual recognition of killer whales," *Scientific reports*, vol. 11, no. 1, pp. 1–16, 2021.

[18] P. C. Gray, K. C. Bierlich, S. A. Mantell, A. S. Friedlaender, J. A. Goldbogen, and D. W. Johnston, "Drones and convolutional neural networks facilitate automated and accurate cetacean species identification and photogrammetry," *Methods in Ecology and Evolution*, vol. 10, no. 9, pp. 1490–1500, 2019.

[19] T. Cheeseman, K. Southerland, J. Park, M. Olio, K. Flynn, J. Calambokidis, L. Jones, C. Garrigue, A. Frisch Jordán, A. Howard *et al.*, "Advanced image recognition: A fully automated, high-accuracy photo-identification matching system for humpback whales," *Mammalian Biology*, pp. 1–15, 2021.

[20] M. Pawley, K. Hupman, K. Stockin, and A. Gilman, "Examining the viability of dorsal fin pigmentation for individual identification of poorly-marked delphinids," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[21] V. Renò, G. Dimauro, G. Labate, E. Stella, C. Fanizza, G. Cipriano, R. Carlucci, and R. Maglietta, "A sift-based software system for the photo-identification of the risso's dolphin," *Ecological informatics*, vol. 50, pp. 95–101, 2019.

[22] R. Maglietta, A. Bruno, V. Renò, G. Dimauro, E. Stella, C. Fanizza, S. Bellomo, G. Cipriano, A. Tursi, and R. Carlucci, "The promise of machine learning in the risso's dolphin grampus griseus photo-identification," in *2018 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*. IEEE, 2018, pp. 183–187.

[23] K. R. Debure and A. Russell, "Feature extraction for content-based image retrieval in darwin," in *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, 2001, p. 470.

[24] K. Urian, A. Gorgone, A. Read, B. Balmer, R. S. Wells, P. Berggren, J. Durban, T. Eguchi, W. Rayment, and P. S. Hammond, "Recommendations for photo-identification methods used in capture-recapture models with cetaceans," *Marine Mammal Science*, vol. 31, no. 1, pp. 298–321, 2015.

[25] A. V. Towner, M. A. Wcisel, R. R. Reisinger, D. Edwards, and O. J. Jewell, "Gauging the threat: the first population estimate for white sharks in south africa using photo identification and automated software," *PloS one*, vol. 8, no. 6, p. e66035, 2013.

[26] R. B. Tyson Moore, K. W. Urian, J. B. Allen, C. Cush, J. R. Parham, D. Blount, J. Holmberg, J. W. Thompson, and R. S. Wells, "Rise of the machines: Best practices and experimental evaluation of computer-assisted dorsal fin image matching systems for bottlenose dolphins," *Frontiers in Marine Science*, vol. 9, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmars.2022.849813

[27] G. Hillman, N. Kehtarnavaz, B. Wursig, B. Araabi, G. Gailey, D. Weller, S. Mandava, and H. Tagare, ""finscan", a computer system for photographic identification of marine animals," in *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] [Engineering in Medicine and Biology*, vol. 2, 2002, pp. 1065–1066 vol.2.

[28] C. Trotter, N. Wright, A. S. McGough, M. Sharpe, B. Cheney, M. A. Civil, R. T. Moore, J. Allen, and P. Berggren, "Towards automatic cetacean photo-identification: A framework for fine-grain, few-shot learning in marine ecology," *arXiv preprint arXiv:2212.03646*, 2022.

[29] J. W. Thompson, V. H. Zero, L. H. Schwacke, T. R. Speakman, B. M. Quigley, J. S. Morey, and T. L. McDonald, "finfindr: Computer-assisted recognition and identification of bottlenose dolphin photos in r," *BioRxiv*, p. 825661, 2019.

[30] H. J. Weideman, Z. M. Jablons, J. Holmberg, K. Flynn, J. Calambokidis, R. B. Tyson, J. B. Allen, R. S. Wells, K. Hupman, K. Urian, and C. V. Stewart, "Integral curvature representation and matching algorithms for identification of dolphins and whales," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[31] J. D. Adams, T. Speakman, E. Zolman, and L. H. Schwacke, "Automating image matching, cataloging, and analysis for photo-identification research," *Aquatic Mammals*, vol. 32, no. 3, p. 374, 2006.

[32] N. G. Blas, L. F. de Mingo López, A. A. Albert, and J. M. Llamas, "Image classification with convolutional neural networks using gulf of maine humpback whale catalog," *Electronics*, vol. 9, no. 5, p. 731, 2020.

[33] R. R. Reisinger and L. Karczmarski, "Population size estimate of indo-pacific bottlenose dolphins in the algoa bay region, south africa," *Marine Mammal Science*, vol. 26, no. 1, pp. 86–97, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1748-7692.2009.00324.x

[34] J. Holmberg, S. Gero, A. Blount, J. Parham *et al.*, "Comparison of three individual identification algorithms for sperm whales (physeter macrocephalus) after automated detection," *bioRxiv*, 2021.

[35] P. Santa Cruz das Ribeiras, "Nnpool in spir pipeline for risso's dolphins identification."

[36] R. Maglietta, R. Caccioppoli, E. Seller, S. Bellomo, F. C. Santacesaria, R. Colella, G. Cipriano, E. Stella, K. Hartman, C. Fanizza *et al.*, "Convolutional neural networks for risso's dolphins identification," *IEEE Access*, vol. 8, pp. 80 195–80 206, 2020.

[37] D. Blount, S. Gero, J. Van Oast, J. Parham, C. Kingen, B. Scheiner, T. Stere, M. Fisher, G. Minton, C. Khan *et al.*, "Flukebook: an open-source ai platform for cetacean photo identification," *Mammalian Biology*, pp. 1–19, 2022.

[38] D. Blount, G. Minton, C. Khan, J. Levenson, V. Dulau, S. Gero, J. Parham, and J. Holmberg, "Flukebook–continuing growth and technical advancement for cetacean photo identification and data archiving, including automated fin, fluke, and body matching. 13," *IWC*, 2020.

[39] M. Lettink and D. P. Armstrong, "An introduction to using mark-recapture analysis for monitoring threatened species," *Department of Conservation Technical Series A*, vol. 28, pp. 5–32, 2003.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

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

[45] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[46] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[48] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for non-parametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[49] F. Alves, A. Alessandrini, A. Servidio, A. S. Mendonça, K. L. Hartman, R. Prieto, S. Berrow, S. Magalhães, L. Steiner, R. Santos *et al.*, "Complex biogeographical patterns support an

ecological connectivity network of a large marine predator in the north-east atlantic," *Diversity and Distributions*, vol. 25, no. 2, pp. 269–284, 2019.

[50] F. Alves, S. Quérouil, A. Dinis, C. Nicolau, C. Ribeiro, L. Freitas, M. Kaufmann, and C. Fortuna, "Population structure of short-finned pilot whales in the oceanic archipelago of madeira based on photo-identification and genetic analyses: implications for conservation," *Aquatic Conservation: Marine and freshwater ecosystems*, vol. 23, no. 5, pp. 758–776, 2013.

[51] J. M. Azevedo, M. Fernandez, M. Cravinho, and C. Coutinho, "Citizen science and corporate responsibility: the case of the monicet platform."

[52] T. Cheeseman, T. Johnson, K. Southerland, and N. Muldavin, "Happywhale: Globalizing marine mammal photo identification via a citizen science web platform," *Happywhale, Santa Cruz, CA, USA, Rep. SC/67b/PH/02*, 2017.

[53] P. N. HalPiN, A. J. Read, E. Fujioka, B. D. Best, B. Donnelly, L. J. Hazen, C. Kot, K. Urian, E. LaBrecque, A. Dimatteo *et al.*, "Obis-seamap: The world data center for marine mammal, sea bird, and sea turtle distributions," *Oceanography*, vol. 22, no. 2, pp. 104–115, 2009.

[54] G. Hillman, B. Wursig, G. Gailey, N. Kehtarnavaz, A. Drobyshevsky, B. Araabi, H. Tagare, and D. Weller, "Computer-assisted photo-identification of individual marine vertebrates: a multi-species system," *Aquatic Mammals*, vol. 29, no. 1, pp. 117–123, 2003.

[55] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.