If There's a Whale There's a Way
ML Midterm Project Report

Maren Rieker
212723@mds.hertie-school.org

Dinah Rabe
d.rabe@mpp.hertie-school.org

Reed Garvin
r.garvin@mds.hertie-school.org

Victor Möslein
212963@mds.hertie-school.org

Abstract

For years, man-made climate change has been one of the main concerns of different scientific disciplines due to its multiple and complex effects on nature and societies. Ecosystems are being altered or completely destroyed. One highly impacted, important and vulnerable ecosystem is the marine ecosystem. Marine mammals are indicators of the ocean's ecosystem. Their protection and conservation is crucial for the balance and therefore health of these ecosystems. In order to successfully carry out meaningful conservation efforts, it is crucial to understand trends, identify animals, as well as monitor and track individuals and populations. Currently, the majority of research institutions still relies on time-intensive and sometimes inaccurate manual matching of photographs by the human eye. We aim to contribute to this important task by automating whale and dolphin photo-ID to significantly reduce image identification times and resources. Therefore, we are investigating the performance of different Machine Learning and Deep Learning algorithms for image classification. We plan to assess their performance in relation to accuracy and training speed. We expect the Machine Learning Models to perform well for the identification of species, but to reach their limit if aiming for the prediction of individuals. A Deep Learning model will be implemented, which can accurately predict an individual animal.

1. Proposed Method

The research process consists of the following steps [1]: First, the images are cropped to their important parts and to quadratic size. Next, the images are segmented to remove the unnecessary background with *Tracer*. Two ML models are then implemented as a baseline that are capable of predicting the correct species. These are tuned in terms of speed and classification accuracy. As the final part of our project, a Deep Learning algorithm capable of predicting individual animals is implemented.

As the goal of the Kaggle competition is the prediction of individual animals out of a pool of 18,000 individuals from 26 species, we decided to divide the project into two stages. In Stage 1, only the species are predicted (therefore 26 classes), to be able to apply ML algorithms. For Stage 2, in line with common practice, we implement a Deep Learning model able to predict on the level of individuals (18,000 classes).

Based on this setup the research process consists of the following st The selection of the classifiers is based on the following criteria: the type of data, the size of the dataset, and therefore the speed-accuracy trade-off.

1.1. Data Preprocessing

All data preprocessing was performed separately from the modelling, as this is a one time operation. The different preprocessing steps will be summarized into one Pipeline at the end of the project. In the following, the different steps of the preprocessing Pipeline are described.

1.1.1 Cropping

To focus on the actual modelling part, we rely at the moment on a pre-cropped data set provided by a fellow Kaggle-competitor [2]. We aim to implement our own model for cropping the images until the end of the project. Cropping all images to the same size is necessary for the next step of pre-processing the data: image segmentation.

[1] GitHub Link: https://github.com/Whale-way/happy-whale

[2] https://www.kaggle.com/datasets/phalanx/whale2-cropped-dataset

1.1.2 Segmentation

For the segmentation of the images, in other words to extract the animals from the background, we use *Tracer* [4]. *Tracer* is a Deep Learning model that works with already cropped images and a specified pixel size. Because our pictures have the size 512x512 after cropping, we employed the TRACER-efficient-5 model, as specified in the ReadMe.

The model works on the basis of pre-trained weights from efficientnet B7.[3] From there, *Tracer* derives which elements in the training image are part of the object and which are part of the background and are going to be coloured white. In order to make this decision, the program sets thresholds in the variation in the pixels.

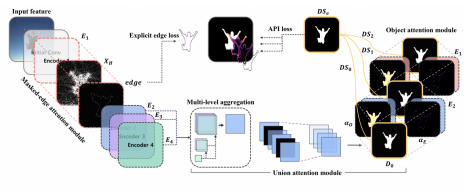

Figure 1. Overview of *Tracer* Architecture. [4]

Tracer repeatedly attempts (Epochs) to find the edge of the object. After the first attempts at finding the edge, this attempt is removed and compared to the final attempt at mass edged attention module, creating the segmented image. This comparison attempts for any loss that may have occured to be accounted for, therefore producing a more accurate segmented image.

Figure 2. Example of segmented images. Source: Own illustration

1.1.3 Cleaning

As mentioned above, the last step of the data wrangling is performed in a separate Jupyter Notebook [4]. This includes (1) the cleaning of the provided .CSV-file that includes the individual-IDs (labels for stage 2), the species (labels for stage 1) and the filename, as we discovered misspellings in the species. Additionally, (2) images have to be turned into machine-interpretable data. For this, the package NumPy is used, which can extract all numerical information stored in the color scheme (RGB in our case) by just applying the array method to a picture opened and stored in a variable. For later purposes, a function to resize the picture to different pixel sizes is implemented (i.e., the Deep Learning model uses a 224x224 resolution). The last step is to save the transformed datasets, so that the other classification modules can access the numerical representation of the images without re-running the whole preprocessing every time.

1.2. Baseline ML Models

As described above, we use supervised Machine Learning models to predict species of whales and dolphins. The over 50,000 images can be associated to 26 species. As our data is labeled according to these species, all chosen algorithms are classifiers. In line with common research practice it makes sense to compare different classifiers as, in light of the "No Free Lunch Theorem", one might perform better for one criterion and worse for others. We decided for Logistic Regression and Random Forest for the their good performance regarding the accuracy-speed trade-off [5].

1.2.1 Principal Component Analysis

Reducing the dimensionality is first of all necessary in order to accelerate the training of our models.

As the images we are working with have an already reduced resolution of 224×224 pixels and are colored using the RGB scale, the number of features is $224 \times 224 \times 3 = 150,528$ per image. Considering that the training set alone contains $51,033$ images (see Data section for details), this results in a number of features far too high for the conventional ML models implemented here to work within a reasonable amount of time. Reducing the number of pixels from the original 512×512 was the first approach used to reduce the number of features. The second approach implemented is Principal Component Analysis (PCA) [2].

Reducing the number of pixels further was not a promising path forward for this project. The reason for that is that our images contain very important details that need to be preserved in order to classify the animal into the correct species. The images only depict the dorsal fins or lateral views of the marine mammals, which can look startlingly similar to each other across the different species. This is even more the case as the classification model has to decide between 26 different species of dolphins and whales, the idiosyncrasies between which can be extremely delicate [5].

[3] https://keras.io/api/applications/efficientnet/

[4] GitHub Link: https://github.com/Whale-way/happy-whale/blob/Classification/Data_Preprocessing.ipynb

[5] Kaggle discussion post elaborating on different species' char-

PCA is able to explain a maximum amount of variance of the original images. This therefore allows us to preserve as much information of the original images as possible [5]. The subspace this unsupervised clustering method finds is defined by the dominant eigenvectors of the original data's covariance matrix. The resulting subspace is thus of much lower dimension, meaning that the overall number of features is decreased significantly.

Additionally, we do not expect to lose too much necessary information by using PCA. As shown in Figure 2, the segmented images contain a large amount of white pixels, which can be dropped without loosing any information. Furthermore, noise and unnecessary details, which do not facilitate the classification, may be dropped. Furthermore, the loss in explainability is not seen as a negative effect in our case, as it is not necessary to visualize the numerical representation of the images after applying PCA.

Analysing PCA for dimensionality reduction in Hyperspectral Image Classification, Rodarmel and Shan confirm the advantages of using this method [9]. They found the classification using the data resulting from PCA to yield the same class patterns as when using the entire data set. The very accurate classification results after using PCA are also to be expected when applied to RGB images, as in our case. The significant data reduction and still excellent classification performance clearly speak in favor of using this method for our project.

1.2.2 Softmax Regression

As a first model, we are implementing a Softmax Regression using the Scikit Learn Multiclass Logistic Regression module. Softmax Regression, also known as Multi-class - or Multinomial Logistic Regression and Maximum-Entropy Classifier, is based on the two-class Logistic Regression and generalized to work on multiple mutually exclusive classes [8].

Researchers often rely on Softmax as a conventional supervised ML method for its speed and for the good classification accuracy. Comparing different ML algorithms on image data of different animals, Faaeq et al. found Multiclass Logistic Regression to produce the best results on the original dataset, achieving a 0.98 classification accuracy [1]. At the same time, this method achieved this result in a reasonable amount of time, and much faster than other popular ML classification algorithms, such as Support Vector Machines (SVM).

After the application of dimensionality reduction algorithms, the researchers found K-Nearest Neighbors (kNN) to provide the most accurate results. However, as kNN does not work well with large datasets and with high dimensionalities, due to the need to calculate the distances between each data point, this method is not suitable for our project. Similarly, the researchers found SVM to produce results on-par with kNN in terms of accuracy. However, SVM, as expected, needs substantively more time than both kNN and Logistic Regression, making it unsuitable for the classification task of such large datasets as in our case.

As Softmax delivers similarly accurate results as kNN and SVM and needs significantly less time when working with large datasets, it is the appropriate method for us.

1.2.3 Random Forest

Implementing and training a Random Forest (RF) model is done in a next step, to be able to compare the two selected ML models. RF classifications are popular for achieving similarly good prediction results as kNN, SVM and Softmax, while being significantly faster than those, which is why we are choosing it as a second ML model [1]. Additionally to their good training speed, RF has a number of other advantages that make it useful for our classification task. Most importantly, RF is not sensitive to over-fitting, good at dealing with outliers in the training data and easy to parametrize, which enables trying out different hyperparameter configurations [3].

1.3. Deep Learning Model

The Deep Learning model we are employing and tuning is a combination of the features of a Transformer and convolutions [10]. This Mobile Visual Transformer uses a combined Convolutional Neural Network (CNN) and Visual Transformer(ViT) approach to provide for a light ViT model for image-recognition tasks, which even has the ability to be run on modern mobile phones.

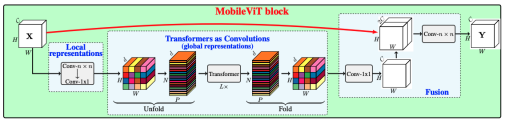

Figure 3. MobileViT Architecture [6]

We have chosen this model for two reasons: the cutting edge of Visual Transformers and how lightweight this model is. This ensures us high performance, great accuracy and a model that can efficiently, in terms of computational capacity, be carried out on the Hertie Server [6].

> "MobileViT has 2× fewer FLOPs and delivers 1.8 percent better accuracy than DeIT on the ImageNet-1K dataset"[6]

In terms of accuracy, this model exceeded other CNN and ViT models, including DeIT, the current standard for

acteristics: https://www.kaggle.com/code/kwentar/what-about-species/notebook

ViT, and more complex models such as MobileNet(V1-3). It has an average decrease in parameters by 2 million.

2. Experiments

Data: Our project is based on a Kaggle competition.[6] They provide a pre-split dataset, containing 27,956 images in the test set and 51,033 images in the training set of whales and dolphins. Additionally, a .CSV-file is provided that contains filename (of the corresponding image), individual-id and species. Images focus on dorsal fins and lateral body views. The raw images have different pixel resolutions.

Evaluation method: For the Deep Learning model we will evaluate our model in contrast to the results of the other Kaggle submissions. The Mean Average Precision (MAP@5) function used for the evaluation will allow us to submit five predictions of the individual photo-ID for each image, leading to a score of one if at least one of them is correct. We will also analyse the accuracy through a loss function.
The two ML models will be compared against each other in terms of prediction accuracy and speed. To check the overall accuracy we will use the cross validation score, which will give us a first idea how good the classifiers perform on different folds of the training set. This will be a first check for overfitting. Furthermore, the precision and recall metrics will be compared.

Experimental details: For establishing our baseline code, we used a sample dataset of 100 images, which we extracted at the beginning. We did this to check and develop our code in shorter time periods, as we need to use the Hertie Server for working with the whole dataset. We provide this dummy dataset so that our code can be run and checked without the server and downloading the entire dataset.

In the next steps, the focus is on optimizing the hyperparameters of the Softmax and RF models and running them on the Hertie Server on the whole dataset. Specifically, for tuning the Softmax model, the effect of normalizing the data in the preprocessing step using Scikit Learn's MinMaxScaler will be examined, as well as choosing different penalty sizes and different types of solvers for the optimization of the Logistic Regression. Additionally, the effect of using one-vs-rest (OvR) versus using cross-entropy loss for the training algorithm will be analysed. Tuning the hyperparameters of the RF classifier will include different ranges for the number of estimators (i.e., trees) and the maximum amount of features used in the model.

To test the different configurations of the models, we will use Scikit Learn's GridSearchCV. This allows us to define a grid of the different parameters and its different values. Using cross-validation, we will then be able to examine which hyperparameter configuration leads to the best models being fitted.

Furthermore, we will experiment with the amount of variance we can keep for the PCA to reduce the number of features enough while still achieving accurate predictions.

For the Deep Learning Model, after running our model we will be able to understand the model's weight decay and training error and thus be able to improve our model. We can change the hyperparameters, such as the size of the the patches and the expansion factor of the model, to improve our results. [7]

Results: The image segmentation was a first major step in our project. The implemented model succeeded in spotting the important parts of the images and extract them from the background, which consists only of water and sky. This is the foundation for our upcoming work in implementing models that predict the correct species or individual animal from the fin or other part of the animal. In a first implementation on a sample of 100 images of the original training set, PCA was able to reduce the number of features per image from the aforementioned $150,528$ to 56, while keeping 95% of the original variance. First results of the baseline classification using Softmax Regression on the sample data set were then computed in a few seconds.

Comment on quantitative results: The result of the PCA are much better than expected. This means that using PCA allows us to run the ML models on the full data set without risking exorbitant training times but still getting very accurate species predictions. It furthermore allows us to experiment with different configurations of those ML models on the entire data set, which was unexpected given the large amount of images.

3. Future work

As outlined, we aim to train the most accurate Softmax and RF models and make a comparison between their performances regarding speed and accuracy.

Regarding the Deep Learning model, it has not been possible for us to run it with our dataset, because the infrastructure at Hertie School is not yet updated for it. We are in contact with the Data Science Lab over this and also on the look for alternative online servers that will allow us to run the model using Tensorflow 2.6.0 or higher. Implementing the Deep Learning model is the big end goal for our project, even if we will most likely not be able to make the Kaggle deadline, which is already on April 19th. It will still be informative for us to compare the accuracy we reached to this of other competitors.

[6]The competition can be found here: `https://www.kaggle.com/c/happy-whale-and-dolphin`

References

[1] A. Faaeq, H. Gürüler, and M. Peker. Image classification using manifold learning based non-linear dimensionality reduction. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. Ieee, 2018.

[2] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019.

[3] N. Horning et al. Random forests: An algorithm for image classification and generation of continuous fields data sets. In *Proceedings of the International Conference on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences, Osaka, Japan*, volume 911, 2010.

[4] M. S. Lee, W. Shin, and S. W. Han. Tracer: Extreme attention guided salient object tracing network. *arXiv preprint arXiv:2112.07380*, 2021.

[5] H. Li. Which machine learning algorithm should I use? `https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/`, 2020. [Online; accessed 08-April-2022].

[6] S. Mehta and M. Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. 2021.

[7] S. Paul. Keras documentation: Mobilevit: A mobile-friendly transformer-based model for image classification. `https://keras.io/examples/vision/mobilevit/`, 2021. [Online; accessed 07-April-2022].

[8] S. Raschka. What is softmax regression and how is it related to logistic regression. `https://www.kdnuggets.com/2016/07/softmax-regression-related-logistic-regression.html`, 2016. [Online; accessed 09-April-2022].

[9] C. Rodarmel and J. Shan. Principal component analysis for hyperspectral image classification. *Surveying and Land Information Science*, 62(2):115–122, 2002.

[10] P. Sayak. Keras-io/mobile-vit-xxs - hugging face. `https://huggingface.co/keras-io/mobile-vit-xxs`, 2021. [Online; accessed 09-April-2022].