Dolphin and Whale Identification Using Machine Learning

Justin Wong, Sitao Chen, Martin Lim
Spring 2022 W207 WBL 007

Brief Description/Title:

Marine researchers monitor dozens to hundreds of different species of dolphins and whales. Since these dolphins and whales live in the sea, it is very difficult to identify them (since they cannot be held still) other than taking pictures of them and comparing them to previous sightings. Similar to how individual people can be distinguished by their iris and fingerprints, dolphins and whales can be identified through unique features (such as the shape of their dorsal fin, tail, or body markings). Identifying the dolphins and whales in each picture is a tedious, manual process that requires careful examination of the human eye against many different pictures of similar creatures in order to identify the particular identity of the dolphin or whale in question. Happywhale is holding a competition to produce a machine-learning algorithm that can accurately identify these creatures using their distinguishing features in a short time. The automation and speed of this process can potentially save thousands of man-hours allocated for identification alone and reduce mistakes.

Team members:

Justin Wong , Sitao Chen , & Martin Lim

Problem Statement:

Input: Digital images of distinguishing features of a dolphin or whale (eg. dorsal fin, body markings, etc.)

Main: Identify the species of the dolphin or whale.

Sub: Identify the ID number of the dolphin or whale.

Objective:

To accurately identify the species of individual dolphins and whales from images of their unique, distinguishing features (eg. fins, tails, body markings, etc.) using Supervised and Unsupervised machine learning techniques. If possible, try to identify the ID.

Approach / Methodology:

1. A sample set of training images was created for data ingestion for model development.
    a. The entire available training image pool consisted of 51033 images, with 26 unique species and 15587 individual ids.
    b. We subsetted the dataset to train our models using the first subset of images consisting of 6750 images, composed of 5400 (4 images per individual) and a random sample of 1350 individuals that are in the first pool for a total of five images per individual representing 21 different species.
2. Transformations were applied on the images during the training process due to memory constraints on Kaggle and in order to standardize the input.

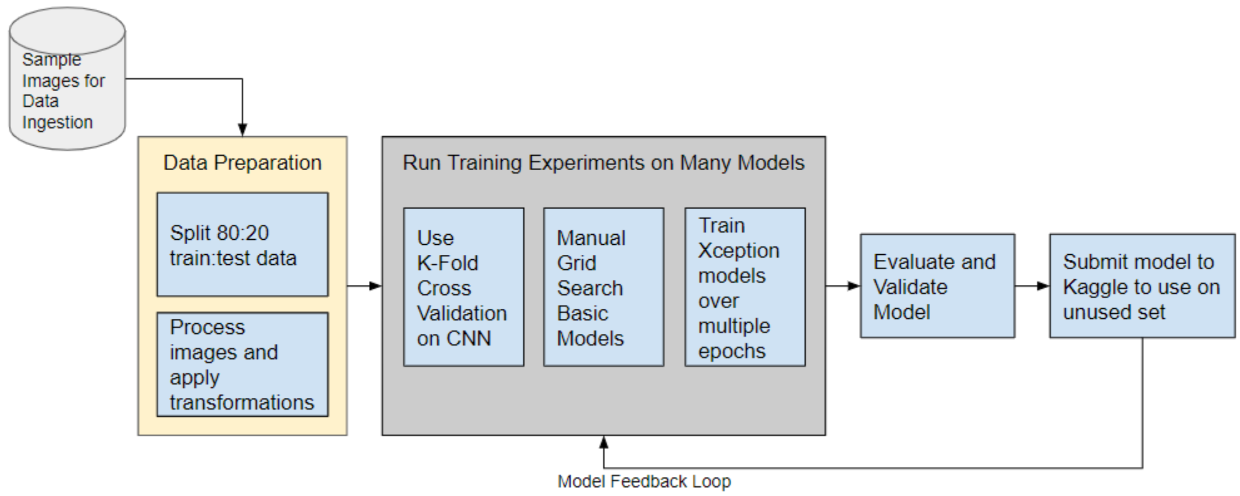
a. For statistical models (all models except the Xception and Convolutional Neural Net), all image pixels were changed to grayscale to standardize the input pixels for more efficient training of the model.
b. Since the images came in varying aspect ratios, all images were converted to a 256 x 256 format to make the input uniform.
c. For the deep learning models, the images were resized to 256 x 256 images, but the RGB pixels were maintained. The training image directory was modified so that images were in folders named for their label, which enabled training on disk memory instead of RAM memory.

3. A number of classifiers, clustering methods, and deep learning models were used to predict both the species and the ID number of the individual images from the training sample:
    a. K-Nearest Neighbors classification
    b. Bernoulli Naive-Bayes classification
    c. Gaussian Naive-Bayes classification
    d. Decision Tree classification
    e. Random Forest classification
    f. Logistic Regression classification
    g. Basic Neural Network classification
    h. K-Means clustering
    i. Gaussian Mixed Model clustering
    j. Convolutional Neural Network
    k. Xception Deep Learning Model
    l. Hybrid Two-Layer classification (for ID only) - double Xception model

    K-Fold Cross Validation was used to ensure the models developed are not overfitted.

4. The different models were evaluated by accuracy and F1 score

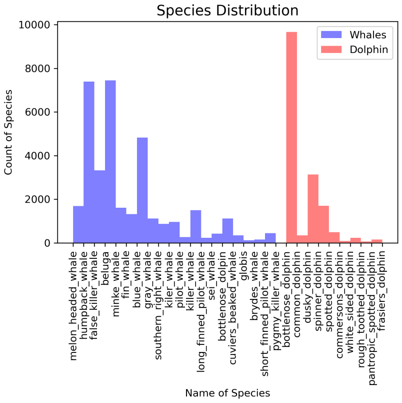
Block Diagram:

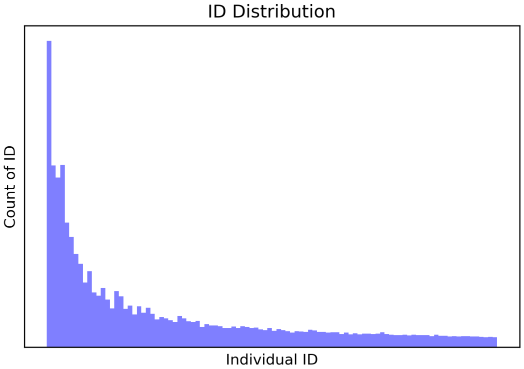

Sample Images for Data Ingestion
Data Preparation
Split 80:20 train:test data
Process images and apply transformations
Run Training Experiments on Many Models
Use K-Fold Cross Validation on CNN
Manual Grid Search Basic Models
Train Xception models over multiple epochs
Evaluate and Validate Model
Submit model to Kaggle to use on unused set
Model Feedback Loop

Images:

The input data are sample JPEG images from the training set that capture certain body parts with distinguishing features of each individual marine mammal (eg. dorsal fin shape, body markings). Most of the images are of the top side of the marine mammal.

The training data provided has varying levels of representations across different species and individual dolphins and whales. Certain species and individuals are well-represented while others have far less data points in comparison.





Datasets:

The data set used for this project comes from the Kaggle HappyWhale competition. The goal of the competition is to produce the best machine learning model that can quickly and accurately predict the ID number of an individual dolphin or whale based on the image of its distinguishing features. The data set contains a large pool of 51033 labeled

training images and 27956 unlabeled test images.
https://www.kaggle.com/c/happy-whale-and-dolphin/data

What is considered success?

We intended to achieve over 50% accuracy in predicting the species. However, if we achieved a less promising precision, we will not consider it a failure because we still would have gone through the experience of taking a dataset from kaggle, processing the images, and training a model until eventual submission.

In terms of predicting individual ID, since there are 15587 unique individuals in the dataset, any model that performs better than 1/15587 would already be better than random guessing, which would be considered a success.

Given the performance of basic methods in predicting the individual ID of the marine mammals, we intended to outperform them in accuracy using hybrid or more complex techniques tailored to image recognition.

Experiments

The development phase was done in four (4) phases. The first phase involved developing and comparing the basic classifiers to gauge their performance in predicting the individual dolphin or whale species and ID. The basic classifiers were trained on a sample train and validation set. The second phase involved developing a CNN more tailored to image recognition while the third phase focused on using a Xception deep learning model to improve on the performance of the CNN. These methods were trained on a larger portion of the training dataset.

1. Basic Classifiers

Several classification and clustering methods were compared to determine how they accurately predicted species and ID number. This phase was performed on a local Jupyter notebook. Each classifier was developed using a sample pool of 6730 images from the test set. The sample pool contained 5 images per individual and represented 21 different species of dolphins and whales. 80% (5400 images) of the sample pool was used to train the models, while 20% (1350 images) were used for testing so that each individual in the test set had four other images in the training subset.

The training and test images were processed, resized, and transformed into a digitized form that could be fed into the models. Each image had 65536 (256 by 256 picture) features where each feature was the value of each of the pixels. The training and test features were further standardized for optimized training. The species names and individual IDs were differentiated and created into separate vectors because they would be used to train the models for different cases.

When training the models to predict species, the 65536 pixel features would be input as the feature data while the species names were treated as the labels. When training the models to predict ID number, the same pixel data was used as the feature data, while the ID vector was treated as the labels. Certain parameters were varied for each model to determine which parameters performed the best per model. The parameters that were varies are listed below:

Two hybrid two-layer models were used to predict the ID of an individual marine mammal. The logic behind the model was to group the training data by species and train a separate model with data from only a certain species so that the variation per species is captured. When predicting an individual ID from a test image, the model would first determine the species by using the Species Classifier (first layer). When the species is determined, the ID will be predicted by using the model trained with the species data (second layer). The output ID would represent the output of the entire model.

Method	**Parameter Varied**	**Input**
k-Nearest Neighbors	Value of k	20 Principal Components
Bernoulli Naive-Bayes	Alpha	20 Principal Components
Gaussian Naive-Bayes	Variable Smoothing	20 Principal Components
Decision Tree	Maximum Depth	20 Principal Components
Random Forest	Maximum Depth	20 Principal Components
Logistic Regression	L2 Strength	20 Principal Components
Basic Neural Network	No variation	All 65536 pixels
K-Mean Clustering	No variation	20 Principal Components
Gaussian Mixed Model	No variation	20 Principal Components
Two-layer Random Forest Classifier	No variation	20 Principal Components
Neural Network to Random Forest Classifier	No variation	20 Principal Components for the Random Forest Classifier All 65536 pixels for the Basic Neural Network

During this phase, it was discovered that all of the basic methods (except the basic neural network) took a very long time to train since all 65536 pixels were treated as individual features. Some models took a few minutes to train, while multiple hours were not sufficient to train others. Only the basic neural network was able to efficiently utilize all the pixel data to predict species and ID. In order to reduce the training time of the other models, Principal Component Analysis was used to reduce the pixel data from 65536 pixels into 20 principal components. The principal components were fed into the models that could not efficiently handle all the pixel data. The table above shows which techniques were used, which parameters were varied, and which input was used to train and test each model. In the later phases, neural networks were used so that all of the variation could be captured and utilized.

In addition to the time constraints of using many features, the RAM memory allocation was not enough to process all of the training data at once (the sample pool of 6750 images alone was more than 8GB so using 38000 images would require far more than 32GB of RAM). In order to circumvent this issue, the models in the later phases would load images from the storage (disk) space rather than having them available in RAM memory. This was a slower process, but it enabled the utilization of more of the training data points.

2. Basic Convolutional Neural Network, Xception Model, and Hybrid Model (Double Xception)

To build on the performance of the basic Neural Network, a Convolutional Neural Network was developed with additional layers to better predict species and ID. In addition to the dense layers used in the basic Neural Network, a Convolutional 2D layer was added and followed by a MaxPooling layer and a Flatten layer. This allowed the data to be accessed in a 2D format without having to digitize the images into 65536-pixel rows in a dataframe that would consume more memory.

In addition to the optimizations on the model itself, the training method was modified to load the images directly from the directory to train the model. Instead of using a local Juypter notebook that would require all the data to be loaded into RAM (and rely on the hardware of our personal machines), the notebook was implemented using the Kaggle website to utilize the Kaggle resources and allow the models to run online. An ImageGenerator was used to pull the images from the public competition directory and train the model more efficiently. The model was trained using a sample dataset with 6750 images like in the first phase

Because of memory constraints, the following approach was implemented to address memory and time constraints:

A. Sample / Select images to be used for training in the challenge's `train.csv`
   a. This train.csv includes the jpg `image` name, `species`, and `individual_id`
   b. Input images consists of:
      i. Image height
      ii. Image width
      iii. Colored values (3 values for RGB, or 1 value for Grayscale)
   c. We sampled the data in a particular way because we were limited by the number of images for any individual whale or dolphin.
      i. To achieve a balanced dataset, we wanted roughly equal representation of all species and individuals for that particular species.

ii. This means we want

1. At least `MAX_IMAGES_PER_INDIVIDUAL_THRESHOLD` images for each individual, so that we have multiple angles of the same individual.
2. At most `MAX_INDIVIDUALS_PER_SPECIES` individuals for any particular species, so that each species would be included an equal number of times.
3. At most `MAX_SAMPLE_DATA_SIZE` images in the entire training set due to memory and computational constraints.

iii. To make sure the model is not overfit, we use the VALIDATION_RATIO to determine the size of the validation data test size. By default, we set `VALIDATION_RATIO=.20`, which means 20% of the sampled data is used as a validation test set.

B. Set up the working directory such that folder names correspond to species class and contents are the images of that species.

a. This means the kaggle working directory looks like:

```
/kaggle/working/
         | beluga/
         | blue_whale/
                        | <img1>.jpg
                        | <img2>.jpg
                        | <img3>.jpg
         .
         .
         .
         | spinner_dolphin/
         | spotted_dolphin/
```

b. This is used to train a LSTM CNN model later on so that the model training reads from files instead of in memory.

C. Develop different models for 4 competing experiments:

a. Baseline Decision Tree

i. Results 59.6% on cross validation datasets

b. CNN model.

i. Create an ImageDataGenerator to create a train iterator using flow_from_directory and use fit_generator to fit the CNN using a generator instead of raw data.

1. This is needed because the training images consists of 60GB, which is too large for memory.

ii. Basic Sequential CNN

1. Results 27% on validation dataset

iii. Cascading Sequential CNN

1. Results 25.94% on validation dataset

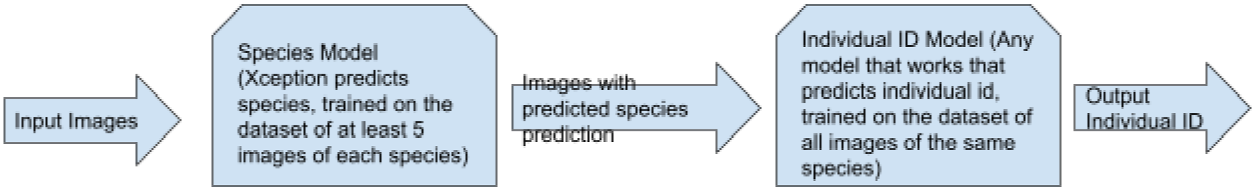
iv. Attempt on cascading sequential on 40k images failed bc out of memory

c. Xception model. (Based on the Xception research paper)

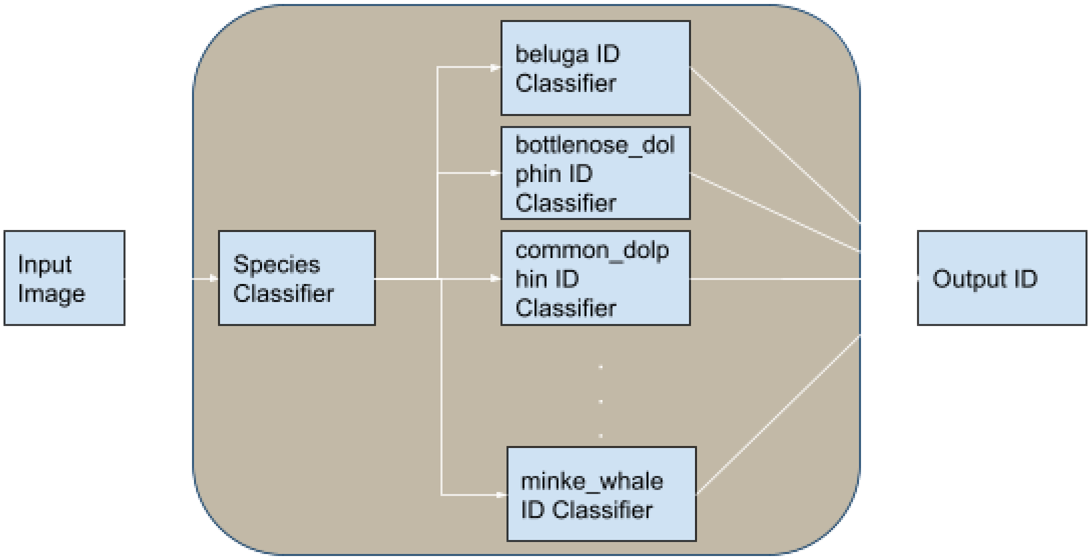
i. We used an existing Xception deep neural net architecture and trained it on our sampled dataset.

- ii. The third phase involved using a more specialized model called an Xception Deep learning model to identify species and ID. The package is a little more “black-boxed” compared to the other Neural Networks, but it had optimizations for easier training. The training process was the same as the Convolutional Neural Network, but was restricted by timeout constraints.
- iii. Training Runs:
    1. Results trained on 6750 trained on species
    2. Xception on 60k trained on individual ids.
    3. Xception on 10k trained on individual ids on 50 epochs.
    4. Xception trained on 38k on all individual ids.
    5. Xception trained on 37k on all individual ids but only 10 epochs.
    6. Xception trained on 38k on all individual ids 10 epochs.
    7. Xception trained on 37k on all species 10 epochs.
    8. ** Xception trained on 6750 on all species 50 epochs. **
        - a. Dataset generated from Output.
        - b. This was the model we ultimately used for the hybrid model

- d. Hybrid Model. Double Xception.
    - i. The final phase of the experiments involved a two-layer hybrid Xception model. This model followed the structure of the two hybrid models from Phase 1 (the Two-Layer Random Forest and Neural Network to Random Forest models).
        1. The first Xception model is trained to predict the species of the test image.
        2. Then the model will call one of N-individual Xception models (trained with images from a certain species) to predict the individual ID of the test image given that it is classified as a member of the species.
        3. The overall output of the model will be an ID number.


Input Images
Species Model (Xception predicts species, trained on the dataset of at least 5 images of each species)
Images with predicted species prediction
Individual ID Model (Any model that works that predicts individual id, trained on the dataset of all images of the same species)
Output Individual ID

- ii. This hybrid model was only developed as a “Proof-of-Concept”. This Two-Layer Hybrid Xception model was not fully completed due to time constraints. The species classification model was trained, but only one second-layer individual Xception model (for Beluga Whales) was trained to predict the ID number of the test image classified as a Beluga Whale. This experiment was also performed on Kaggle to maximize the training dataset.


Input Image
Species Classifier
beluga ID Classifier
bottlenose_dolphin ID Classifier
common_dolphin ID Classifier
minke_whale ID Classifier
Output ID

iii. Training Runs
1. P1 from CNN and the Xception model that predicts species.
2. P2 trains Xception model on individuals
    a. Beluga model on 37k images
    b. Beluga model on 10k images
    c. Beluga only
3. P3 combined uses the P1 model to predict the species, then P2 to predict the individual id of beluga whales.
    a. Phase 1 species applied on entire training set.
    b. Phase 2 species predicted followed by individual id predicted

D. Comparison between the models
  a. Single Xception model trained on individual ids
  b. Hybrid model with Xception model trained on species and another Xception model trained on individual id for each given species

E. Evaluate metrics using accuracy and f1-score.
  a. Evaluation of Xception model → Accuracy 83% on all images predicting species.
  b. Evaluation of Hybrid → Accuracy 58% on all beluga labels/predictions, predicting individual ids

Results

After conducting the different experiments to predict species and ID number, the classifiers performed much better at predicting species rather than ID number (no model among the basic classifiers scored 3% accuracy or better). This was likely due to the far larger proportions of sample data points per species (21 species) compared to individual IDs (1350 individuals) among the 6750 sample images. In terms of species, the basic classifiers generally performed at the 50% range in terms of accuracy where Random Forest (best species accuracy of 55.111%), K-Nearest Neighbors, Neural Network, and Logistic Regression beat the 50% threshold while K-Nearest Neighbors (best species F1 score of 0.5104) and Random Forest exceeded a F1 score of 0.5. These classifiers alone seemed to reach the mark for success for species prediction. It is noteworthy that the clustering methods performed poorly when predicting species and there were too many individuals to properly use them to predict individuals. The Two-Layer Random Forest Hybrid Model and the Neural Network to Random Forest classifier did not perform better than their single-model counterparts in terms of ID prediction accuracy and F1 score. The Two-Layer Random Forest scored 1.966% accuracy and 0.0144 F1 score while the Neural Network to Random Forest classifier performed even worse, scoring 1.259% accuracy and 0.0076 F1 score.

The Two-Layer Hybrid Xception model was not fully completed due to time constraints. Only the individual species model for beluga whales was trained and used to predict the ID number of a beluga whale. The 58% accuracy figure was produced when correctly predicting the ID number of all beluga whales from a sample test set. However, the opposite trend seemed to occur when moving onto more complex models geared toward facial recognition. Both the Convolutional Neural Network and the Xception models predicted the ID's better than the basic classifiers (27.38% and 96.30% accuracy respectively), but the Convolutional Neural Network did not perform as well when predicting species (25.931%). It seems that the more complex models performed better for predicting specific individuals more than generalizing to a larger group (species).

Species Prediction

Method	Best Accuracy (%)	Best Parameter for Accuracy	Best F1 Score	Best Parameter for F1 Score
K-Nearest Neighbors	54.593	K = 10	0.5104	K = 10
Bernoulli Naive-Bayes	42.296	Alpha = 10	0.3813	Alpha = 2
Decision Tree	44.296	Max depth = 5	0.4235	Max depth = 10
Random Forest	55.111	Max depth = 50	0.5090	Max depth = 50
Gaussian Naive-Bayes	46.0	Var smoothing = 0.01	0.458	Var smoothing = 0.01
Logistic Regression	50.667	L2 = 1.0	0.4564	L2 = 1.0
K-Means	0.889	N/A	0.0064	N/A
Gaussian Mixture Model	4.0	N/A	0.0417	N/A
Neural Network	51.185	N/A	0.4565	N/A

ID Prediction

Method	Best Accuracy (%)	Best Parameter for Accuracy	Best F1 Score	Best Parameter for F1 Score
K-Nearest Neighbors	2.889	K = 1	0.0224	K = 1
Bernoulli Naive-Bayes	1.037	Alpha = 1e-10	0.0057	Alpha = 1e-10
Decision Tree	1.185	No max depth	0.0091	No max depth
Random Forest	2.667	Max depth = 1000	0.0186	Max depth = 1000
Gaussian Naive-Bayes	1.926	Var smoothing = 0.01	0.0120	Var smoothing = 0.01
Logistic Regression	1.407	L2 = 0.1	0.0068	L2 = 0.5
Neural Network	0.074	N/A	~0	N/A
Two-Layer Random Forest	1.926	N/A	0.0144	N/A
Neural Network to Random Forest	1.259	N/A	0.0076	N/A

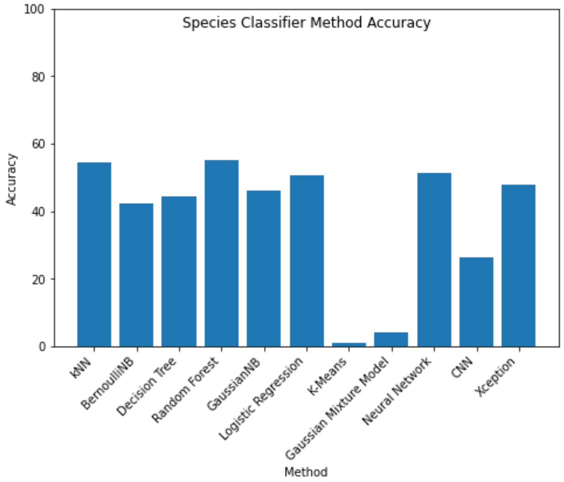
Convolutional Neural Network	25.931	N/A
Hybrid Xception*	58.29%	N/A

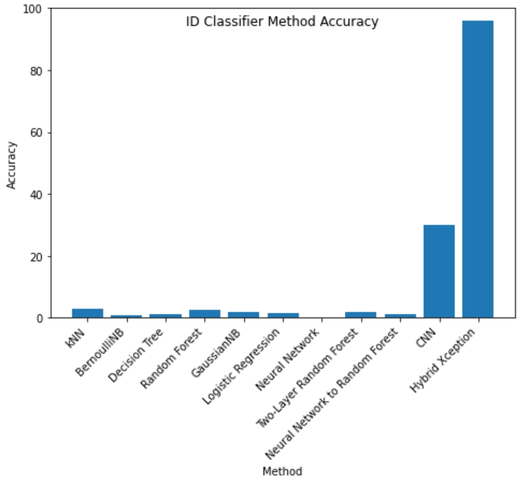
Convolutional Neural Network	27.38	N/A
Xception	83.01	N/A

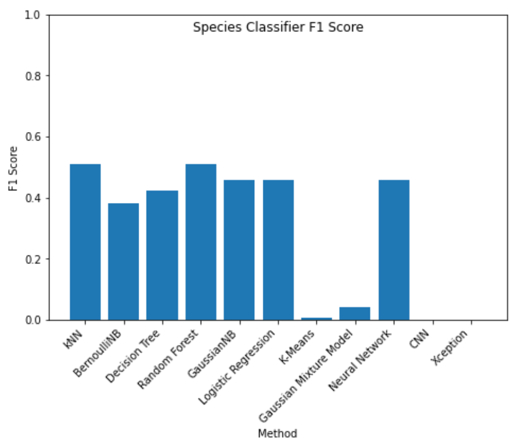
NOTE: The hybrid Xception model predicting individual_id was calculated using the dataset that consisted of true beluga species or predicted beluga species.

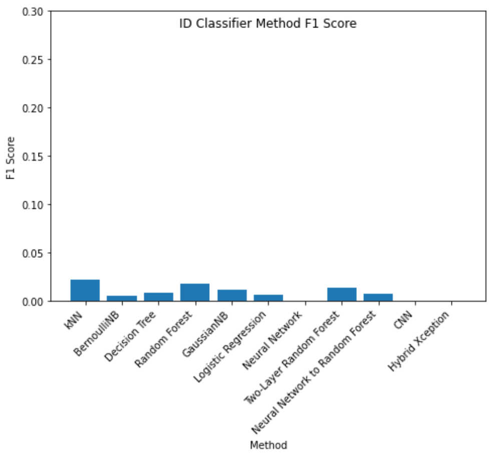
Tests/Graphs/Discussions

Among the basic classifiers Random Forest (an improved version of decision tree classifiers) performed the best when predicting species and second best when predicting individual ID. It was a noteworthy observation that the K-Nearest Neighbors performed, despite being known as a “lazy-learner”, very well compared to other models. It had the best accuracy and F1 when predicting individual ID and the second best scores when predicting species. Another noteworthy observation was that a simple neural network performed well in predicting species (third in both accuracy and F1) but performed the worst in predicting individual ID. This made it necessary to innovate the way it processed the images in order to boost its performance. Both Naive-Bayes classifiers performed slightly worse than the top classifiers. Both basic hybrid models applied the logic of segregating species data to predict ID, but performed slightly worse than their simpler counterparts. After trying more complex models, the Convolutional Neural Network and the Xception models performed well in predicting ID but did not perform as well in predicting species compared to the basic classifiers.









Constraints

- The sample dataset for model development:
  - 6750 data points
  - 80%/20% train/test split
  - 21 species
  - 1350 individuals (5 images each)
- Training using the full dataset / computing resources was lengthy:
  - 38,000 training images was the limit due to the time constraints on the Kaggle platform
  - Our notebooks running on Kaggle timed out because the notebooks took over 43,200.90 seconds to run
- No labels for the test set (Kaggle submission required) so model development was performed on the training test set only
- All images except a few were RGB so gray-scaling was required for all images; this removed some variation in the data

Standards

- Python Version: 3
- Python Packages used:
  - Standard Python Libraries
  - Numpy
  - Pandas
  - Matplotlib
  - Jupyter Notebooks (local training)
  - Kaggle (larger models)
  - Keras
  - SK Learn
  - Tensorflow
  - Keras
  - SciPy
  - H5py
  - Xception

- Jupyter Notebooks (local training)
- Kaggle (for larger-scale training)
- Disk Storage used: 68+ GB
- RAM Memory used: 32 GB

Comparison

The Xception DNN and CNN models performed better than the basic classifiers in predicting ID because they were fine-tuned and made complex enough to detect the unique individual features rather than generalize onto a larger group of species. This was probably the reason why both models did not perform as well in predicting the species.

The Random Forest classifier may have performed the best among the basic classifiers (in terms of accurately predicting species) because it may have had a good blend of feature importance as well as improving on the performance of a single decision tree classifier. Since multiple decision trees are made, and that the depth was not as deep, overfitting may have been better mitigated and helped the model generalize better to the test data.

K-Nearest Neighbors may have performed better than expected because the memory/distance-based learning may have grouped similar species/individuals near each other. Similar species tended to have a similar color/shade and body proportions, so the values of the major principal components may have been closer together between individuals of the same species.

The drawback for the Naive-bayes algorithms may have been that the features were not completely independent of each other. The texture, shape, and color of the overall body parts of a dolphin or whale tend to be similar since they are of the same body of a certain species. This limitation may have put a ceiling on the effectiveness of these algorithms because they did not capture how the features are related to each other.

Limitations of Study

- Due to the disproportionate distribution of species and individual mammals, individual ID number identification was difficult because certain individuals and species did not have as many data points to train the model to detect them. This forced a tradeoff of not utilizing all the data points in order to prevent the model from predicting the most frequent species or individual.
- Due to timeout constraints on Kaggle’s testing environment (for the CNN and Xception models), the final models were able to train with only 38,000 images out of 51003.
- The dolphins and whales aged over time, or may have endured injuries that caused their appearance (and distinguishing features) changes slightly so these changes are unaccounted for when using training images of the same individual from different dates in time

Future Work

To improve the accuracy and reliability of the model, the following improvements are suggested for future work

- Implement ensemble methods such as bagging/boosting to improve the performance of the decision tree algorithms (for both standalone and hybrid models)
- Implement a Siamese model to harness its capability to distinguish unique human facial features to detect uniquely-shaped dolphin or whale features
- Implement bounding boxes to crop out surrounding objects that cause unnecessary variation and a mask to remove the pixel variation in the images due to background features (sea, sky, landmarks, foreign objects) so that the additional variation caused by features unrelated to the marine mammals will not be captured or used to predict species and/or individuals
- Automatically flip/rotate the image so that all pictures will have the individuals face the same way (eg. to the left, to the right, etc.) because will reduce unnecessary variation due to orientation
- Increase the contrast in images to improve neural network models

Github Repository

- https://github.com/landmund/w207-spring2022-final-project-wong-chen-lim

References

- https://www.kaggle.com/competitions/happy-whale-and-dolphin/overview
- https://www.gograph.com/vector-clip-art/whale.html
- https://ai.stackexchange.com/questions/6274/how-can-i-deal-with-images-of-variable-dimensions-when-doing-image-segmentation
- https://arxiv.org/pdf/1610.02357.pdf