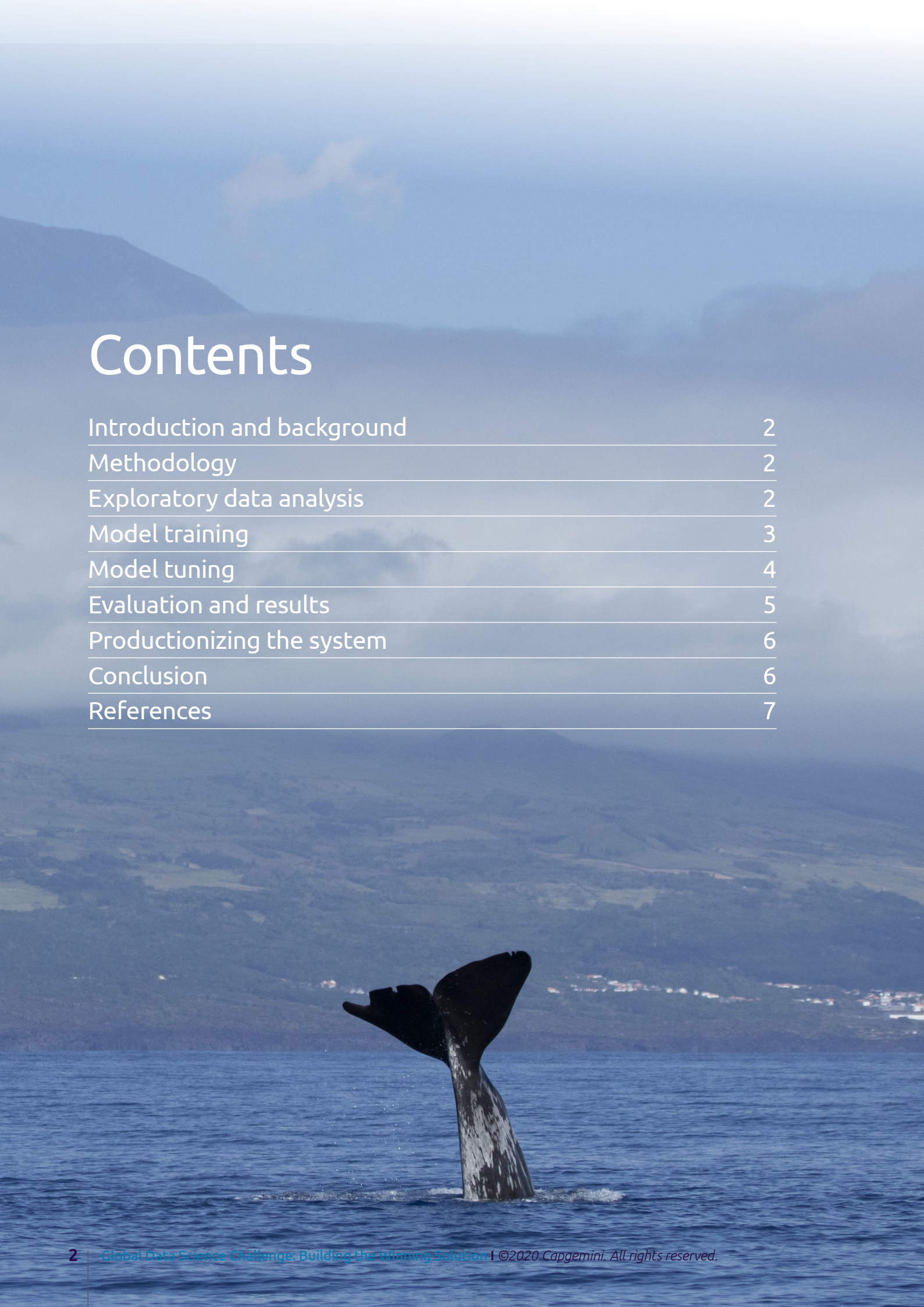
Global Data Science Challenge for a sustainable future

Building the winning solution - Using AI to help identify and track sperm whale population

Contents



 I ©2020 Capgemini. All rights reserved.

Introduction and background

Since 2006, sperm whales have been listed as endangered on the International Union for Conservation of Nature (IUCN) Red List of Threatened Species. Their plight is due to extensive hunting from the 18th to the 20th century and other more modern threats such as naval sonar and "ghost" (abandoned) fishing gear. Scientists now dedicate their lives to monitoring the whales' movements, observing their social structures, and ensuring the protection of the species, which is not possible without knowledge of how they behave down to an individual level.

To track and protect the sperm whale, researchers identify individuals via images of their flukes, which can be used in an equivalent way to human fingerprints due to their uniqueness from whale to whale. However, this is a very time-consuming and difficult task for the researchers. Support from software applications has been limited, making it a challenge to efficiently keep track of and update a catalog of more than 2,200 individuals over several decades. As Lisa Steiner, a marine biologist specializing in sperm whales, explains:

"I have to manually assist the program to pick out the contour for each half of the fluke. If the photos are good, this process doesn't take very long; however, if there isn't a lot of contrast between the fluke and background, or there is a lot of glare on the edge of the fluke, I have to follow the contour manually."

In the 2020 Global Data Challenge, an internal Capgemini-wide competition, entrants were asked to solve this problem for Lisa Steiner with the use of new machine learning (ML) techniques. An impressive 693 teams took the challenge, utilizing advanced image recognition and identification to create a new system for identifying whales.

This article presents the approach used by the competition's winning team, and describes how they successfully created an ML model that uses many of the same identifiers as scientists such as Steiner to match images of the many whale individuals.

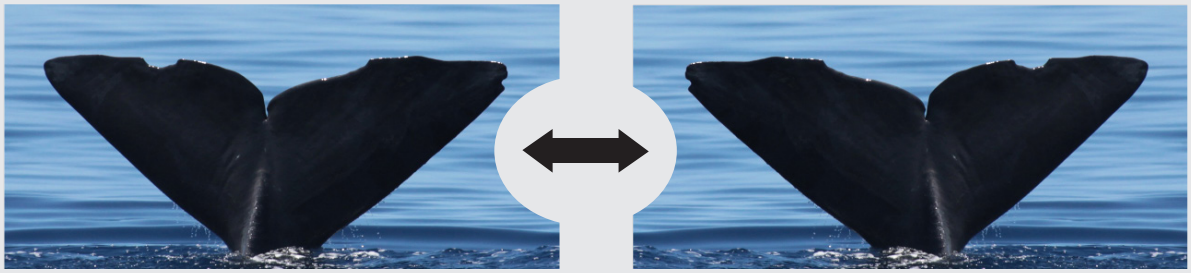
Methodology

The winning team carried out three main steps that are commonly used in ML projects, as well as for data science competitions such as this, so the same approach can be applied to other similar problems. The three steps are exploratory data analysis, model training, and model tuning. Let's look at each in turn.

Exploratory data analysis and data pre-processing

Exploratory data analysis (EDA) proved to be one of the winning aspects of the project, where one of the first things investigated in the EDA step was the class distribution of the images in the dataset. Around 75% of all the whale individuals had only a single image, 12.5% had just two, and the remaining 12.5% had more than two. The fact that most cases had only one image posed a major challenge for training the ML identification model.

A strategy that can be implemented for whale flukes like these is to flip the images horizontally. Due to the flukes not being symmetrical, this generates a completely new set of imaginary whale individuals from the real whales, increasing the size of the dataset. The figure below shows an example: The left-hand image has a marking (notch) only on the left side of the fluke, while the right-hand image has the same marking on both sides. The ML model can be trained on these images as if they are of completely different whales.



The EDA also revealed that nearly all the images in the training and test datasets had been cropped. Just a few were original images, completely unaltered. Training ML models on a dataset with inconsistencies such as this can be much more difficult, as important feature differences are lost in the resultant noise. To achieve a consistent crop for all images, 800 images were labeled for the training of an ML model used for cropping images, and the cropping model was fine-tuned for sperm whale flukes. This resulted in a dataset with less noise from differences in image crops – and also from splashes from waves and other features that are irrelevant for identifying the whales.

While cropping removed a lot of noise from the dataset, there was also a significant number of images with features that made them difficult to use for identification. Low-resolution images, those where only half the fluke was visible, and those taken from the wrong side of the fluke from usual are examples of images that were judged to only introduce more noise to the dataset. During the EDA process, several identifiers for these types of images were established, meaning that they could be removed efficiently to improve the training set.

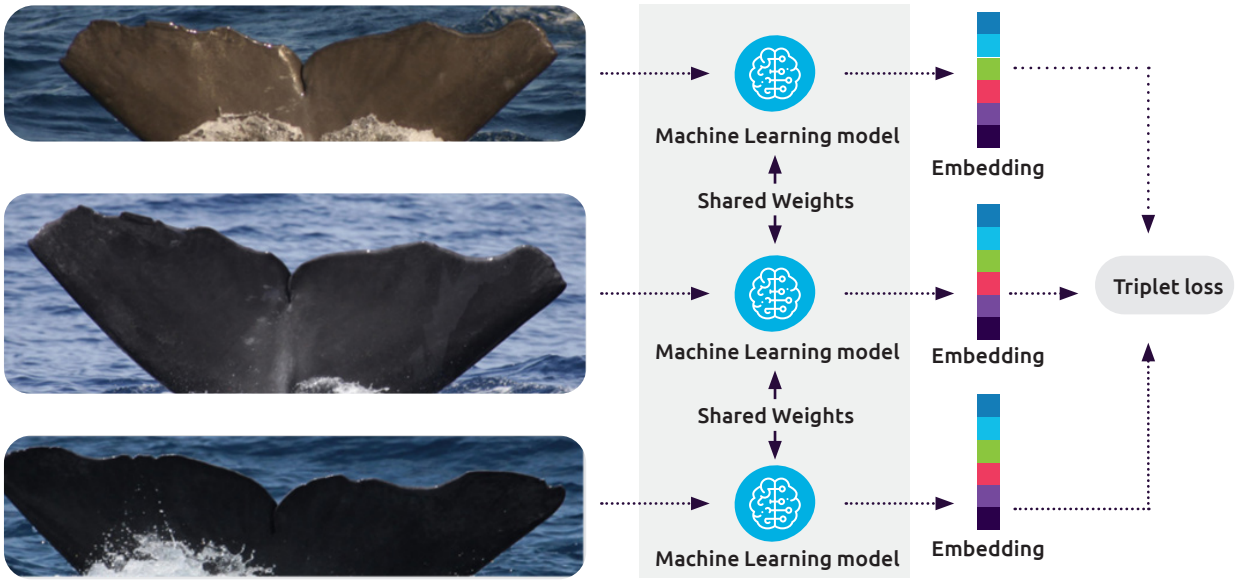
Model training

Two common methods used for identification with ML are using Siamese networks[1], and approaching the problem as a multi-class classification problem. During the EDA it was found to be a large class imbalance, with most individuals having very few images and some having many. This can result easily result in a poor and inaccurate ML model. Because of this, the team decided to use both classification and Siamese networks to take account of special considerations that can counteract the challenges of imbalanced datasets.

When dealing with a classification problem, the model is trained to predict which class, an image from the dataset, belongs to. ArcFace[2] and Focal Loss[3], as well as a regular softmax function, are the loss functions used in the classification part of this model. The first two loss functions specialize in problems with extreme class imbalances, as was found in the EDA, and proved to be very influential in the training of the models. However, for inference, the model also needs to be able to make predictions on completely new individuals not seen before. For one-shot learning problems like this, Siamese networks are more suitable.

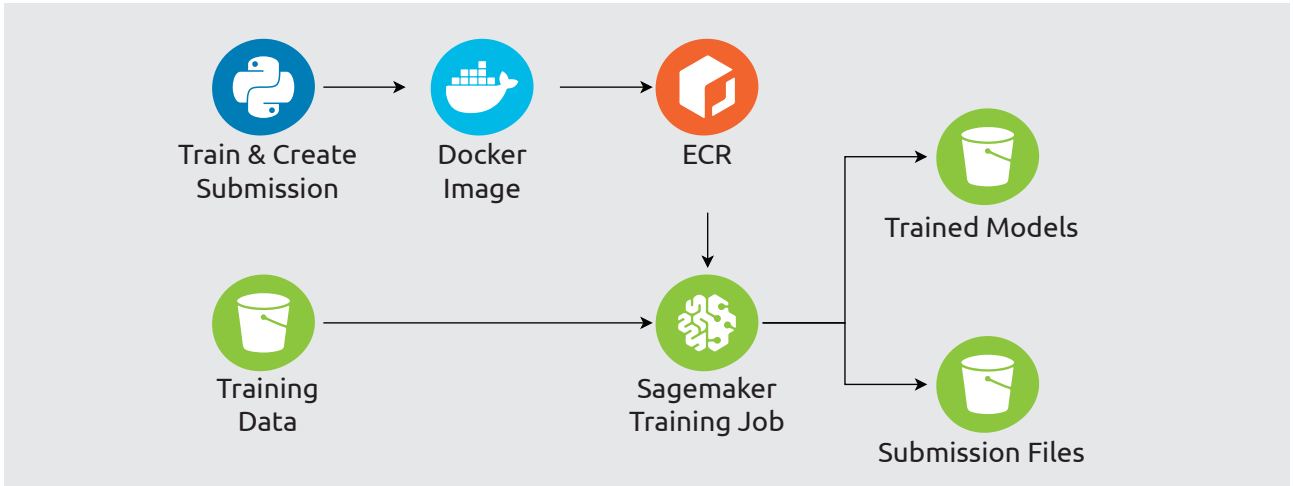
Siamese networks are trained by comparing the feature representations of images, in this case using a method called triplet loss[4]. Here, the feature representation of an image is compared to the feature representation of two other images; one of the same individual and one of some other individual in the training set. During training, the similarity between the feature representation from the image with the same identity is then maximized, while the similarity to the other individual is minimized. To address the issue of class imbalance, hard example mining[5] is also applied; here, the model is purposely given images that are hard to separate, to make it better at differentiating individuals from each other.

 ©2020 Capgemini. All rights reserved.



During training, the Siamese and classification approaches were combined by using the output from the last layer in the network before the class prediction as the feature representation, and updating the weights with the losses from both. When selecting a backbone architecture to be used by both the Siamese and the classification approaches, several different model types were tested. In the end, ResNet-101[6] trained using ImageNet[7] model weights gave the best results. As mentioned earlier this approach can be applied to similar problems, and by using transfer learning from ImageNet model weights, a similar training setup and training procedure can be used for other problems and datasets. The two main reasons behind this is that models trained for the ImageNet dataset are considered a very general starting point for image recognition models due to the 1000 classes in the dataset and that, except for mirroring the flukes, no whale specific preprocessing was used to train this model.

To achieve good results from model training, the validation strategy must also be selected carefully. In this project, the class imbalance and number of images in the dataset had to be accounted for when creating the validation dataset. By selecting random images relating to those whale individuals with more than two images, a balance between the number of individuals with more than one image in the training dataset and variation in the validation dataset was found. Finding the right balance here is important to give the model enough data and opportunities to learn similarities between images of individuals, while also ensuring a robust model through good validation. The size of the validation set, however, was one of the parameters tuned in the final methodology step, discussed in the next section.



For model training, the Amazon Web Services (AWS) platform was utilized. With Amazon SageMaker being a central component in developing and testing the ML model. The figure above gives an overview of the architecture used to train ML models using SageMaker. The Docker images were created with the training code and pushed to Elastic Container Registry (ECR) where they could be accessed by SageMaker. Amazon Simple Storage Service (Amazon S3) buckets were used to store the training data together with the resulting trained models and submission files for the competition. Initial training and experimentation on model architectures and training procedures were done on p3.2xlarge and p2.xlarge instances, optimized for deep learning with a single NVIDIA V100 and K80 GPUs.

Model tuning

In the final step of the methodology, the hyperparameters of the model are fine-tuned to achieve the best possible performance. Model architectures, types of augmentations, and image resolutions are among the hyperparameters tuned to achieve the optimal results. However, meaningful changes and a good plan for experimentation were important in order not to overfit the model to the public leaderboards, where the teams were able to test their model on a part of the test dataset, validate their results, and see standings compared to other teams.

This process requires more computing power than the other steps, as a lot of models have to be trained to find the optimal set of hyperparameters Here the use of the AWS platform came into its own, as several models could be trained quickly and in parallel, allowing for a large amount of experimentation. Using p3.8xlarge instances with SageMaker optimized for deep learning, training of a model took approximately three hours, with some variations caused by different image resolutions and early convergence of some sets of hyperparameters. The p3.8xlarge instances use NVIDIA Tesla V100 GPUs with 64GB of memory to offer accelerated computing, which allowed the team to speed up the model tuning phase after initial testing on smaller p3.2xlarge and p2.xlarge instances.

The team did a total of 50 iterations of this training process during the model tuning phase before selecting the predictions from their best model for submission. Due to the flexibility of using cloud computing the team was also able to run several of the model tuning experiments in parallel, greatly shortening the time used on model tuning. The conv1 and conv2_x layers shown in the table below (which is from the ResNet101[6] model) were frozen during the entirety of the training procedure. This resulted in a total of 46,862,525 trainable parameters in the model after replacing the final softmax layer with custom classification layers.

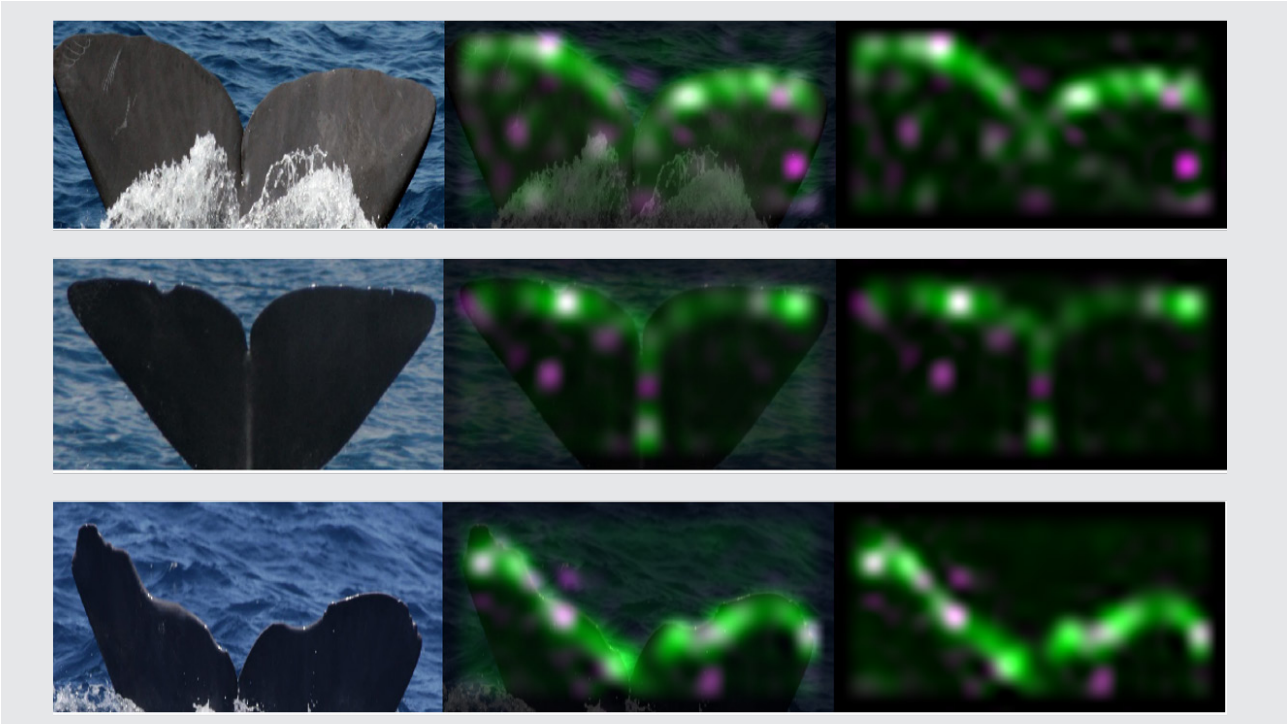
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

 ©2020 Capgemini. All rights reserved.

Evaluation and results

In the competition, teams were asked to find the 20 images that were most similar to each image in the test dataset and then sort them by similarity. The scoring was designed so that more points were gained by finding a match on the image ranked as the most similar by the model, but some points were also awarded for having matches in the top 3 and top 20 most similar images. While it doesn't cover the entirety of the scoring function top-n score can be used to evaluate the results. With this scoring function, a prediction is considered correct if one of the "n" most similar images is of the correct whale. For top-1 this resulted in 88%, for top-3 91%, and 94% for top-20. The winning team also had a 3% better score than the team in second place on the private leaderboard, where results are not revealed until the end of the competition. The concrete plan for tuning and experimenting with model hyperparameters also proved beneficial in helping the winning team gain a position from the public leaderboard, where the models could be tested against a subset of the test set, to the leaderboard containing the full test set and into first place.

The raw results are not the only interesting part when analyzing the results of an ML model. Interpretability is also an important topic in both the understanding of results and the acceptance of the model. The latter is vital because ML models are often viewed as "black box" models with no way of knowing how the model arrived at a conclusion. By analyzing the outputs of the final convolutional layer of the ResNet-101 model used as a backbone network after training, we can see where the model focuses its attention and how it recognizes different individuals.

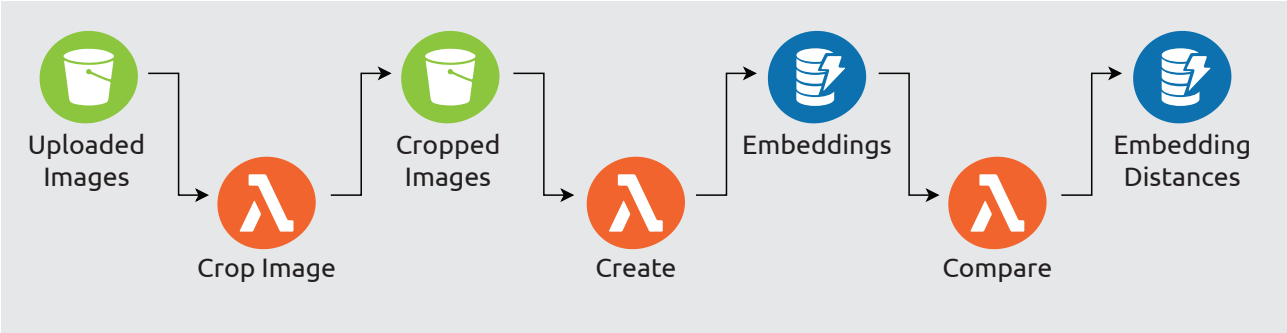


From a visual analysis, it is clear that it focuses on the trailing edge of the fluke, much as the human researchers do. It also activates on the smallest of markings on the fluke itself – the first and second rows of images in the figure below are good examples of this. For future usage of the model, this suggests that the model has generalized well and will continue to perform satisfactorily when new whale individuals are photographed. Using the insights gained from this analysis also provides a lot of new information that can be used in new development and to improve on the model. Especially in verifying that the model has focused on the areas that differentiate the flukes, and is not affected by unimportant features of the images such as wave splashes and other obstructing objects.

Productionizing the system

After completion of the competition, a system was implemented to replace the legacy systems used by whale researchers. For use in the field, the requirements in terms of computation and usage differ from what is required during training. Thus a new approach was used to connect the ML models with a web application built using AWS Amplify, a framework on AWS for quickly building scalable web applications.

For the ML part of the system, AWS Lambda was used to deploy both the cropping and whale identification model for inference. Lambda gives a quick response following cold starts of the system and provides sufficient memory and computation to create feature representations of images. In the model's "Embeddings" database, precomputed feature representations of images are stored, while the "Embedding Distances" database stores the Euclidean distance between every pair of images. This allows for precomputed comparisons and fast queries for potentially matching individuals. From a cold start and with a database of almost 6,000 images, the feature representation is created and similar images are presented within 2.5 minutes. This provides users with adequate results, even at times when the system is used infrequently. As is the case for the mentioned researcher, Lisa Steiner, who then quickly can upload and match images when she gets back from studying the whales.



Conclusion

It is never going to be easy to identify whales by their flukes and keep track of individuals that may not have presented themselves to the camera for many years. However, by using modern ML techniques for image recognition, scientists can be assisted with the process so that they can focus on their goals of monitoring, understanding, and protecting sperm whales. Interestingly, the ML model has independently arrived at the same methods that have been used by researchers for years, and so will likely have generalized well. The hope is that it will be a valuable aid to scientists' work long into the future.

The proven three-step methodology used by the winning team in the Capgemini Global Data Science Competition proved a key factor for success; all three steps contributed to differentiating their solution from competing ones. The EDA step gave the team an early advantage in having a well-prepared dataset with limited noise compared to the original. In the model training step, the team used their knowledge of different approaches to the training of identification models, and combined them for the highest possible accuracy. And finally, in the model tuning step, the computational power of AWS was utilized in combination with a well-structured model tuning plan to avoid overfitting.

This three-step methodology also leads to the creation of an ML model and training procedure that can be applied to many similar use cases as well. The main building blocks in creating this was the extensive EDA and preprocessing steps that were used in combination with a model trained using the ImageNet weights, which are commonly used in widely varying image recognition tasks.

The operational system built after the competition to replace the legacy systems used by researchers greatly simplifies a tedious and labor-intensive task. The open platform enables other researchers to contribute images from their collections, track the movements of whales across the oceans, and ultimately gain a better understanding of the lives of sperm whales.

 ©2020 Capgemini. All rights reserved.

References

1 Siamese Neural Networks: An Overview
Davide Chicco
Artificial Neural Networks pp 73-94 (2021)

2. ArcFace: Additive Angular Margin Loss for Deep Face Recognition
Jiankang Deng, et al. (2018)
https://arxiv.org/pdf/1801.07698.pdf

3. Focal Loss for Dense Object Detection
Tsung-Yi Lin, et al. (2018)
https://arxiv.org/pdf/1708.02002.pdf

4. Deep Metric Learning with Hierarchical Triplet Loss
Weifeng Ge, et al. (2018)
https://openaccess.thecvf.com/content_ECCV_2018/papers/Ge_Deep_Metric_Learning_ECCV_2018_paper.pdf

5. Hard Example Mining with Auxiliary Embeddings
Evgeny Smirnov, et al. (2018)
https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w1/Smirnov_Hard_Example_Mining_CVPR_2018_paper.pdf

6. Deep Residual Learning for Image Recognition
Kaming He, et al. (2015)
https://arxiv.org/pdf/1512.03385.pdf

7. http://www.image-net.org/

About Capgemini

A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of almost 220,000 team members in more than 40 countries. The Group reported 2019 global revenues of EUR 14.1 billion.

Visit us at

www.capgemini.com

Learn more about us at:

https://www.capgemini.com/partner/amazon-web-services/

People matter, results count.

The information contained in this document is proprietary and confidential. It is for Capgemini internal use only. Copyright © 2020 Capgemini. All rights reserved.