Whale & Dolphin Species Detection Using Hybrid Efficient Net Architecture

Harini Chiruvella[1], I. Adithya Sriraj[2], Dr. Y. Sreenivasulu [3]

[1, 2, 3]Department of E.C.E, Sreenidhi Institute of Science and Technology, Hyderabad, India

Abstract: We can identify individuals by their fingerprints and faces through facial recognition, but can we do the same for animals? In reality, scientists use the shape and markings on marine animals' tails, dorsal fins, heads, and other body parts to manually monitor them. Photo ID, or identification by natural markings, is a valuable tool for the study of marine mammals. It permits analyses of population status and trends as well as the tracking of specific animals through time. We streamline the procedure for photographing whales and dolphins, which speeds up image recognition by 99 percent for researchers. The majority of research organizations currently rely on labor-intensive and occasionally erroneous manual matching by the human eye. Manual matching, which entails looking at images to compare one person to another, finding matches, and identifying new people, takes thousands of hours. When there are several photographs to manage, the first way is tiring and has a restricted scope and audience. In order to get highly discriminative features for species detection, this research offers a novel and effective approach by feeding images into a Deep Convolutional Neural Network Architecture utilizing an Additive Angular Margin Loss. By determining the largest occurring class among the available features, we apply K-Nearest Neighbors to infer the model by computing the closest distances to feature embedding of distinct classes.
Keywords: feature embedding, additive angular margin loss, and convolutional neural network.

I. INTRODUCTION

The cornerstone for solving image classification or computer vision issues is a convolutional neural network. Higher resolution images present a challenge for conventional convolutional neural networks since they require a deeper network with a correspondingly larger number of layers, which leads to a more recent issue called Vanishing Gradient. Stochastic Gradient Descent is the algorithm used by neural networks. The procedure, which is iterative, moves down the slope of a function starting at a random position until it reaches the function's most advantageous point. The algorithm features particular hyper parameters that may be tuned and altered to enhance the network's functionality and the method by which it finds the global minima. We employ the transfer learning technique in this paper, where the model has already been trained and will be put to use. Designing suitable loss functions that can improve the discriminative power is one of the primary issues in feature learning utilizing Deep Convolutional Neural Networks (DCNNs) for large-scale recognition applications. To assess the quality of any network, the loss function is applied on top of the DCNN. With the help of an angular margin to identify features unique to many species, we also hope to increase model accuracy in this research without sacrificing image resolution. By utilizing K-Nearest Neighbors, a supervised machine learning technique that predicts the output by taking into account the "k" nearest feature embedding, we also want to make comprehending and drawing conclusions about the predictions easier.

II. DRAWBACKS OF EXISTING SYSTEM

A deep learning architecture is the convolutional neural network (CNN). CNNs just make the prediction to let us know whether or not the necessary component is there. Determining a loss function is crucial for a neural network to increase the potency of discriminative features, which is one of the key disadvantages of large-scale CNN. The performance of the convolutional neural network tends to saturate toward mean accuracy after a given depth in the layers, which is more depth and has no effect on performance. However, because deep structures require a lot of processing power, large-scale image recognition applications would be prohibitively expensive. Thus, additional CNN model components including loss functions, non-linearities, optimizers, etc., have been the focus of study in recent years. Traditionally, picture recognition tasks have employed the softmax loss function. Deep image recognition suffers under significant intraclass appearance fluctuations because the softmax method does not explicitly optimize the feature embedding to ensure better similarity for intraclass samples.

III. PROPOSED METHODOLOGY

In this study, we employ a transfer learning approach whereby the model is pre-trained and the initial weights and biases are generated at random using a probability distribution. For our current issue statement, we then feed the data into our neural network design. The input data is accepted by the hidden layer, which then applies an activation function and transmits the activations to the following layer. Up to the anticipated outcome, we disseminate the results. Forward propagation is the term for this. By comparing the projected outcomes to the ground truth labels after producing the predictions, we determine loss. Then, after computing the gradients of the loss with respect to the weights and biases determined for the network, we make the necessary adjustments by deducting a tiny amount proportionate to the gradient. A single epoch is defined as the duration of the complete procedure, which is applied to the full training set. Our prediction score should grow or our error should decrease as we increase the number of epochs since the model is more used to learning the labels. Due to the increased resolution of the pictures we work with, a deeper network is required to extract fine-grain features, which leads to poor training performance and the disappearing gradient issue. Therefore, we suggest in this research a new and improved version of the current way to scale up the network's performance using compound scaling, which scales the image's depth, breadth, and resolution all at once. With a set of predefined scaling coefficients, this technique scales the network's breadth, depth, and resolution equally. This approach of compound scaling is employed. With smaller models and quicker training times, the current state-of-the-art convolutional neural network for image classification applications has revolutionized feature extraction.
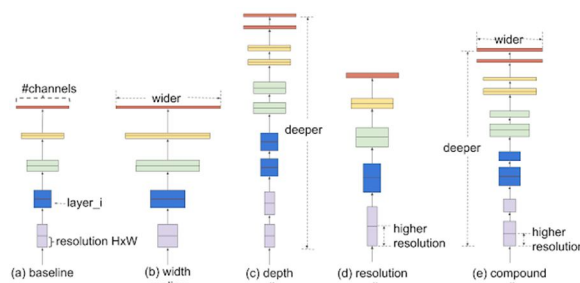

Fig.1 Compound Scaling of Efficient Net

Although there are numerous classes in our problem, there are relatively few samples per class. Now, to get around this, we compare the feature vectors and photos using a similarity function. The primary goal of the similarity function is to increase the distance between feature vectors corresponding to different species while learning data embedding/feature vectors corresponding to photos of the same species. Arc Face - Additive Angular Margin Loss is the similarity function, also known as the loss function, that we deal with. It suggests an additive angular margin penalty to improve the discriminative power of the model between inter-classes in comparison to its predecessors, which are Centre Loss, Softmax Loss, and Sphere Face..
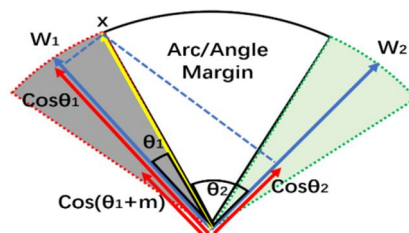

Fig.2 Arc Face Loss depiction of two different classes

The class separations in Arc Face is more stringent and has better performance over the large-scale datasets for image recognition.
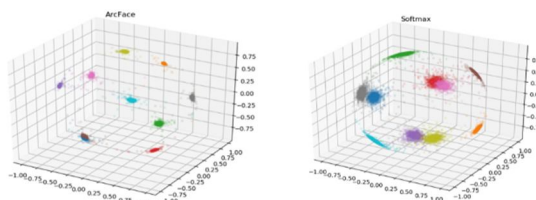

Fig.3 Comparision between Sotmax Loss function and Arc Face Loss functions

IV.   PROBLEM STATEMENT

Using the forms, characteristics, and patterns of their dorsal fins, backs, heads, and flanks, whales and dolphins may be recognised in a collection of photos. Some animals and individuals have very distinctive characteristics, whereas others have significantly less.
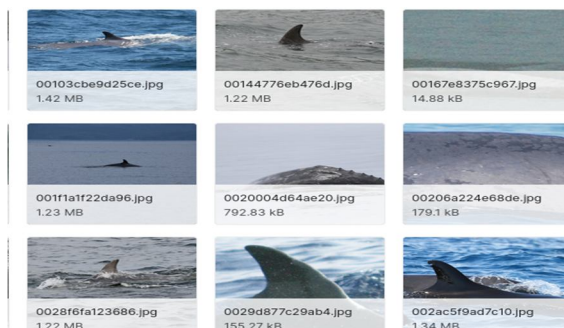

Fig.4 Train and Test images of our dataset

The dataset includes photos of more than 15000 distinct marine animals, representing 30 different species, that were gathered from 28 different research institutions. To help the model recognise, we give it a collection of training and test photos. By using a picture as input, the aim is to accurately identify and forecast the sort of person. Additionally, a CSV file with three columns titled "picture," "species," and "individual id" is provided.

Fig.4 Train.csv file

V.   DESIGN SYSTEM

In this research, the model is trained using weights and biases; photos are fed into the hidden layers for feature extraction using EfficientNet; and ArcFace is used to compare embedded features. Traditionally, picture recognition has employed the softmax loss function.

The libraries we used are:
- Tensorflow
- Scikit-learn
- Matplotlib
- NumPy
- Pandas
- Efficient Net

We employed the TensorFlow processing unit (TPU) to process our large-scale image recognition challenge in order to reduce the time-to-accuracy for training this complicated neural network model. As a part of exploratory data analysis, we turn our photos into TensorFlow records and add enhancement to them. In order to improve the image, adjustments are made to the saturation, contrast, and hue as well as the amount of noise and blur present in the image. Then, in order to extract the minuscule and fine-grained information from the photos, we feed these images via an Efficient Net architecture. The Efficient Net model scales up the performance using a compound scaling method with parameters that correspond to the depth, width, and resolution of the picture, as well as a tunable parameter Ø. Now we train a controllable parameter Now, we train the model using a very slow learning rate and use the neural network Stochastic Gradient Descent technique to look for the local minima of the function.

```
Layer (type)                     Output Shape             Param #     Connected to
==================================================================================
inp1 (InputLayer)                [(None, 512, 512, 3)     0

efficientnet-b5 (Functional)     (None, None, None, 2     28513520    inp1[0][0]

global_average_pooling2d (Globa  (None, 2048)             0           efficientnet-b5[0][0]

dropout (Dropout)                (None, 2048)             0           global_average_pooling2d[0][0]

dense (Dense)                    (None, 512)              1049088     dropout[0][0]

inp2 (InputLayer)                [(None,)]                0

head/arcface (ArcMarginProduct)  (None, 15587)            7980544     dense[0][0]
                                                                      inp2[0][0]

softmax (Softmax)                (None, 15587)            0           head/arcface[0][0]
==================================================================================
Total params: 37,543,152
Trainable params: 37,370,416
Non-trainable params: 172,736
```

This is a description of the model we built, and it is clear from it that the input layer consists of an RGB picture with a resolution of 512 by 512 and three colour channels. Efficient Net architecture is used in the second layer of our model to extract fine characteristics from the supplied photos. Additionally, we employ the regularization method of dropout to avoid overfitting when training our neural network. Just before the output layer, the loss function is constructed as ArcFace to determine whether the feature similarities vector space. This loss function is highly useful for our situation since it has excellent feature discrimination ability and employs an angular additive marginal distance to prevent the overlapping of classes that are very similar to one another. Due to the correspondence of the geodesic distance, the function may produce fairly accurate decision boundaries between the classes. Arc Face increases the compactness within a class and the disparity between classes.

VI.   RESULT AND CONCLUSION

It's time to assess the correctness of the testing data after our model has been trained, assembled, and fitted to the training set of data. We employ K-Nearest Neighbors, a supervised learning approach, to infer the neural network. When the KNN function's n neighbors parameter is set to 50, the picture is split into feature space, and KNN employs the 50 closest neighbours to the new data point to categorize the image by presenting the class with the highest frequency among the closest neighbours. By doing this, photos of the same species will have a comparable feature space and any nearby data point may be easily recognized as belonging to that image. In this research, we systematically investigate CNN scaling and show that a good strategy to improve accuracy and efficiency is by carefully balancing network width, depth, and resolution. To solve this problem, we offer a straightforward and incredibly efficient compound scaling technique that enables us to efficiently scale up a baseline CNN to any target resource limitations in a more principled manner while keeping model effectiveness. In order to prevent label overlap, we also utilize the Arc Face loss function to increase the classes' ability to discriminate. Using a straightforward approach that predicts the class by taking the largest number of nearby characteristics around the class, we also utilized a very basic method to infer and understand how the model is doing. new data point.

REFERENCES

[1] Mingxing Tan, Quoc V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. [Online]. Available: https://arxiv.org/abs/1905.11946

[2] Mingxing Tan, Quoc V. Le, EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling. [Online]. Available: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

[3] Arjun Sarkar, Understanding EfficientNet – The most powerful CNN architecture. [Online]. Available: https://medium.com/mlearning-ai/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad

[4] Daniela Gingold, Face Recognition and ArcFace: Additive Angular Margin Loss for Deep Face Recognition. [Online]. Available: https://medium.com/analytics-vidhya/face-recognition-and-arcface-additive-angular-margin-loss-for-deep-face-recognition-44abc56916c

[5] Sik-Ho-Tsang, [Paper]EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (Image Classification). [Online]. Available: https://sh-tsang.medium.com/efficientnet-rethinking-model-scaling-for-convolutional-neural-networks-image-classification-ef67b0f14a4d

[6] Keras Applications. [Online]. Available: https://keras.io/api/applications/efficientnet/

[7] Aakash Agrawal, The Why and the How of Deep Metric Learning. [Online]. Available: https://towardsdatascience.com/the-why-and-the-how-of-deep-metric-learning-e70e16e199c0

[8] K-Nearest Neighbors. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[9] Abhishek Thakur, Approaching (Almost) Any Machine Learning Problem Book

[10] Prakhar Ganesh, From LeNet to EfficientNet: The evolution of CNNs. [Online]. Available: https://towardsdatascience.com/from-lenet-to-efficientnet-the-evolution-of-cnns-3a57eb34672f

[11] Charudatta Manwatkar, How to Choose a Loss Function For Face Recognition. [Online]. Available: https://neptune.ai/blog/how-to-choose-loss-function-for-face-recognition

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)