Humpback Whale Identification Challenge
Can you identify a whale by the picture of its fluke?

Introduction:

Humpback Whale Identification Challenge, Can you identify a whale by the picture of its fluke? Is a challenge on Kaggle to identify individual whale. Whale identification is a crucial task for conservation efforts, an accurate identification of individual whales can help monitor their population, movements, and health. To aid whale conservation efforts, scientists use photo surveillance systems. The primary objective of humpback whale identification challenge is to indentify individual whales based on the shape and markings on their flukes. These features serve as the whale's "fingerprint," enabling scientists to distinguish between different whales[1].

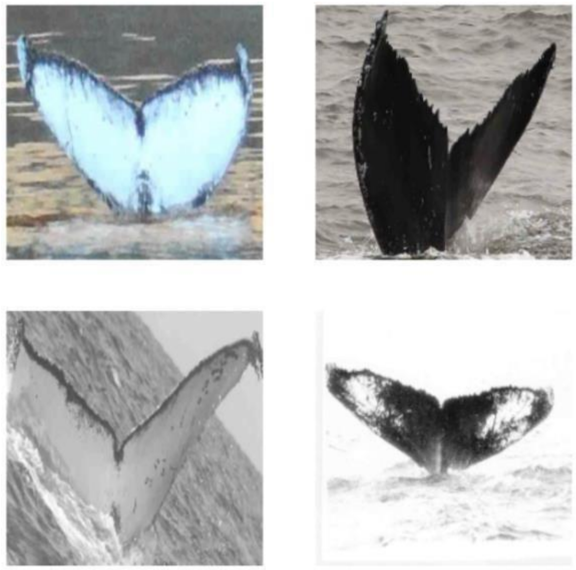
The dataset was available to download on Kaggle and contains 9850 images for training and 16000 images for testing[2], since the labels for test set are not given to download, I took 20% of training dataset for testing the model.

The model uses ResNet18 architecture with $l2$ regularization and was trained with the application of data augmentation techniques.

Data description

The training set consists of 9,850 images with varying sizes. Using images of different sizes is inconsistent for machine learning training, as most models require fixed input dimensions. To address this issue, all images were resized to a standard size of 224x224 pixels, which is a common input size for many deep learning models.

Different sized images



Reshaped Images

The dataset contains over 4,250 unique classes, each representing a distinct whale. Only classes with more than three images were considered for training. This ensures a balanced dataset and avoid classes with too few samples, This threshold was chosen because it allows the model to learn more effectively from multiple examples, reducing the risk of overfitting to a single image or over-representing rare examples in the training process. The resulting dataset has a total of 996 classes resulting in a reduced dataset of 4,752 images.

The new dataset is subjected to data augmentation techniques such as Average and Motion Blur, Random Addition, Random Multiplication, Random Scale, Translation, Shear, and Rotation.

Approach

1. Average and Motion Blur:
Average using 3x3 kernel. In average blur, the kernel calculates the average value of the region and replaces the central element. Motion blur simulates the distortion on the image due to abrupt motion when taking a photo. For this type of blur, a random kernel size is drawn from uniform distribution U(3,5)[1], which causes an almost imperceptible visual effect due to the high resolution of the images.

Blurred Images

2. Random Addition:

Random addition adds a random value to all pixel values. In the implementation, the random value is chosen uniformly from the interval [-10, 10] per image with a probability of 0.5 and added to every pixel for each channel[1].

Random Addition

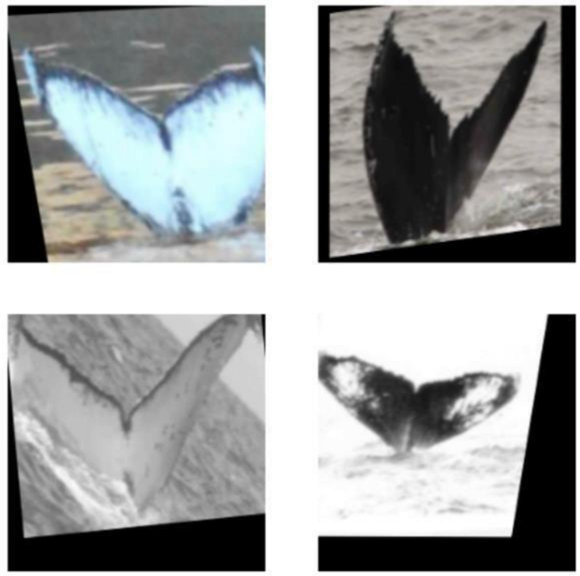
3. Random Multiplication:
In Random multiplication, image tensor is scaled by the sampled value. the scaling factor is drawn from the uniform distribution U(0.9,1.1) and has a 0.5 probability of having different values sampled for each channel[1].

Random multiplication

4. Random Scale, Translation, Shear, and Rotation:
Scaling, translation, shearing, and rotation are all affine transformations, and they are all applied together in the implementation for better generalization and computational efficiency. Scaling refers to the change of the original image spatial dimensions by some factor, followed by a filling operation when the transformation shrinks the image to preserve the tensor dimensions. Independent scaling factors on the spatial dimensions are used, both drawn from the distribution U(0.9,1.1)[1].

Translation is based on a random percentage of the image dimensions. For both width and height, a random value is drawn from the distribution U(-0.05,0.05). Pixel values undefined due to the translation are zeroed, while the pixels outside the image are ignored.

The last two affine transformations, shearing and rotation, have their angles drawn uniformly from the interval [-10, 10] (in degrees), and have the empty tensor entries zeroed instead of replicated[1].



Random Scale, Translation, Shear, and Rotation

The model used for this problem is ResNet18 architecture, which is a deep convolutional neural network (CNN) that is widely used for image classification tasks. ResNet18 is a variation of the Residual Network (ResNet) family and consists of 18 layers, including convolutional and fully connected layers.

The model is first pre-trained on the ImageNet dataset (with the weights set to IMAGENET1K_V2), which provides a good initialization for the weights and biases of the convolutional layers. This technique is known as transfer learning and helps improve the performance of the model.

The original fully connected layer in the pre-trained model is replaced with a new fully connected layer, with the number of output nodes set to the number of unique whale ids (996). This new layer is initialized randomly and will be trained to adapt to the specific whale identification task.

The loss function used for training the model is the Cross-Entropy Loss, which is commonly used in multi-class classification problems. The equation for cross-entropy loss is:

$$L(y, y') = -\sum[y * \log(y')]$$

where y is the true label (one-hot encoded vector) and y' is the predicted probability

distribution over the classes.
The optimization algorithm used is Adam (Adaptive Moment Estimation), with a learning rate of 0.001 and weight decay of 1e-3($l2$ Regularization).

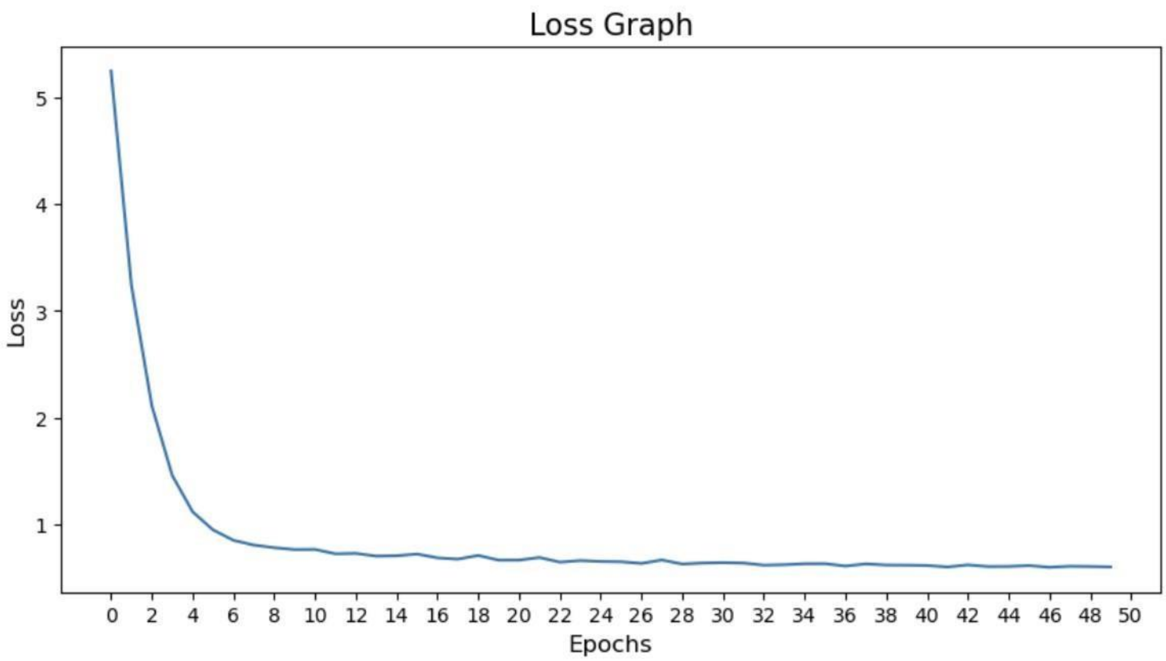
The DataLoader object is configured with 64 images per batch.

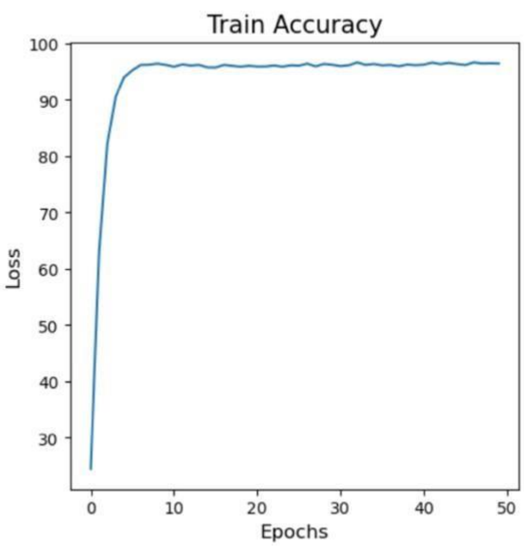
During training, the parameters of the initial layers are frozen, which means their weights won't be updated. This is done to preserve the pre-trained features learned from the ImageNet dataset. The parameters in the final residual block (layer4) and the new fully connected layer are set to be trainable. This allows the model to learn more specific features related to the whale identification problem.

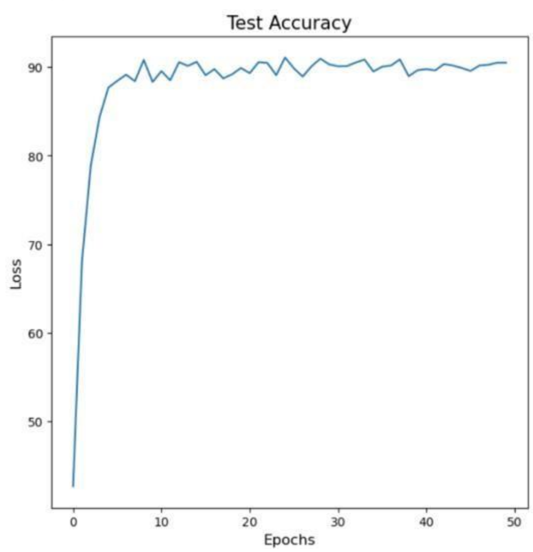
Results

The model is trained for 50 epochs with $l2$ regularization which helps prevent overfitting by encouraging the model to learn smaller, more generalizable weight values.

Train Accuracy	Test Accuracy
96.45%	90.49%







References:

1. Henrique da Fonseca Sim~oes jo~ao meidanis april 29, 2021. (n.d.). https://www.ic.unicamp.br/~meidanis/PUB/IC/2019-Simoes/HWIC.pdf
2. https://www.kaggle.com/competitions/whale-categorization-playground/data