Sea Animals Classification: Traditional Machine Learning vs Deep Learning

Kshitij Yerande
Concordia University
Montreal, Canada
Student ID: 40194579

Siddhartha Jha
Concordia University
Montreal, Canada
Student ID: 40201472

Github Link: https://github.com/Kshitij13579/Sea-Animals-Classification

***Abstract*—Digital imagery and video have been widely used in many undersea and oversea applications. Automated labeling of sea animals is essential for labeling and gathering of information. In this paper we compare approaches to classify sea animals into 9 categories Corals, Crabs, Penguin, Sea Urchins, Seahorse, Seal, Sharks, Starfish, Turtle Tortoise using traditional machine learning approaches and deep convolution neural network. Traditional machine learning involves using of SIFT and HOG features with linear classifier and deep learning involves use of convolution neural network. Transfer learning and shallow convolution neural network is presented in the deep learning approach. Both the traditional machine learning approach and deep learning approaches are compared and the metric as evaluated and interpreted to identify the suitability of solution approaches or classifying sea animals.**

***Index Terms*—image classification, transfer learning, CNN, support vector machines, histogram of gradients, SIFT, ResNet-50**

I. Introduction

Most life forms began their evolution in aquatic environments. About 90% of the world's living space is provided by the oceans in terms of volume. Fish, which are only found in water, are the first known vertebrates. Some of these transformed into amphibians, which dwell both on land and in water for parts of the day. A few subgroups of one group of amphibians, which also included sea turtles, seals, manatees, and whales, developed into reptiles and mammals. Some underwater habitats are supported by plant life that grows in the water, such as kelp and other algae. The base of the ocean food chain is made up primarily of phytoplankton, which are important primary producers.

Sea animal classification is a fundamental task in marine biology, conservation, and fisheries management. The process involves identifying and categorizing different types of marine creatures based on their physical characteristics, behavior, and habitat. This task is vital for understanding the diversity of life in the oceans, monitoring and managing marine ecosystems, and ensuring the sustainability of fisheries.

Traditionally, sea animal classification has been approached using machine learning techniques. These methods rely on feature engineering, where domain experts manually extract relevant features from the raw data, and then train a model to classify the animals based on these features. Machine learning algorithms, such as decision trees, logistic regression, and support vector machines, have been used in this context.

However, traditional machine learning techniques have limitations in handling complex data and extracting useful features. For example, in the case of marine animals, the data can be heterogeneous and multidimensional, and it can be challenging to identify relevant features that capture the essential aspects of the animals' characteristics. Additionally, feature engineering can be time-consuming and requires expertise in the specific domain.

In recent years, deep learning techniques have gained popularity in the field of sea animal classification. Deep learning methods can automatically learn useful features from raw data, reducing the need for manual feature engineering. This has led to significant improvements in classification accuracy and robustness.

Convolutional neural networks (CNNs), for example, are particularly well-suited for image-based classification tasks, such as identifying different types of fish or coral species. These networks consist of multiple layers of convolutional filters that learn to extract relevant features from images. Recurrent neural networks (RNNs), on the other hand, are useful for sequence-based classification tasks, such as identifying the species of a whale based on its vocalizations.

Deep learning approaches have been used in various contexts in marine biology, such as:

- Fish species identification: Fish species identification is a critical task for understanding fish biodiversity, monitoring fish populations, and managing fisheries. Deep learning approaches have been used for this task, achieving high accuracy rates. For example, a recent study used a CNN to automatically identify fish species from underwater video, achieving an accuracy of over 90
- Coral classification: Coral reefs are one of the most diverse ecosystems in the world, and coral classification is a challenging task due to the high diversity and complexity of the corals. Deep learning approaches have been used for this task, achieving high accuracy rates. For example, a recent study used a CNN to classify coral species based on their images, achieving an accuracy of over 90
- Marine mammal vocalization classification: Marine mammals, such as whales and dolphins, use vocalizations to

communicate with each other, and these vocalizations can be used to identify the species of the animals. Deep learning approaches have been used for this task, achieving high accuracy rates. For example, a recent study used an RNN to classify the vocalizations of different whale species, achieving an accuracy of over 90%.

Deep learning approaches have also been used for other tasks in marine biology, such as object detection, image segmentation, and anomaly detection. Despite the benefits of deep learning approaches for sea animal classification, there are some challenges and limitations that need to be addressed. One of the main challenges is the need for large amounts of labeled data. Deep learning models require a significant amount of labeled data to train effectively, and this can be difficult to obtain in some cases. In addition, deep learning approaches can be computationally intensive and require significant resources, such as powerful GPUs and large amounts of memory. Finally, deep learning models can be difficult to interpret and explain compared to traditional machine learning models.

In conclusion, sea animal classification is an essential task in marine biology, conservation, and fisheries management. Traditional machine learning techniques have been used in this context, but they have limitations in handling complex data.

II. Related Work

There is a significant amount of related work in the field of sea animal classification using machine learning and deep learning techniques. The paper [6] to identify fish species from underwater video describes the use of a deep learning approach to automatically identify fish species from underwater videos, achieving high accuracy and demonstrating the potential for using deep learning in marine research. Applications of Machine Learning in Chemical and Biological Oceanography [10] where ML successfully classified the mammals using their acoustics, detecting endangered mammalian and fish species in a specific environment. Most importantly, using environmental data, the ML proved to be an effective method for predicting hypoxic conditions and harmful algal bloom events, an essential measurement in terms of environmental monitoring. Automatic annotation of coral reefs using deep learning [13] aims to automate the analysis of large available AUV imagery by developing advanced deep learning tools for rapid and large-scale automatic annotation of marine coral species. A Parallel Classification Model for Marine Mammal Sounds Based on Multi-Dimensional Feature Extraction and Data Augmentation [16] describes the design of a classification model based on a multi-channel parallel structure, which can process multi-dimensional acoustic features extracted from audio samples, and fuse the prediction results of different channels through a trainable fully connected layer. It uses transfer learning to obtain faster convergence speed, and introduces data augmentation to improve the classification accuracy. These topics provide a good starting point for further exploration and understanding of the current state of the art in sea animal classification using machine learning and deep learning techniques.

III. Proposed Methodology

A. Dataset

The data set contains different images of marine animals. Some images were taken from pixabay.com and requires no license or attribution when used. Other images were taken from flickr.com where attribution to the original authors will be required when used commercially. Currently, there are 19 different classes available and may be extended further in the future. The images are resized to either (300px, n) or (n,300px) where n is a pixel size less than 300px [1]. In this paper we are using 9 classes of the data set approximately 6000 images of classes Corals, Crabs, Penguin, Sea Urchins, Seahorse, Seal, Sharks, Starfish and Turtle Tortoise.

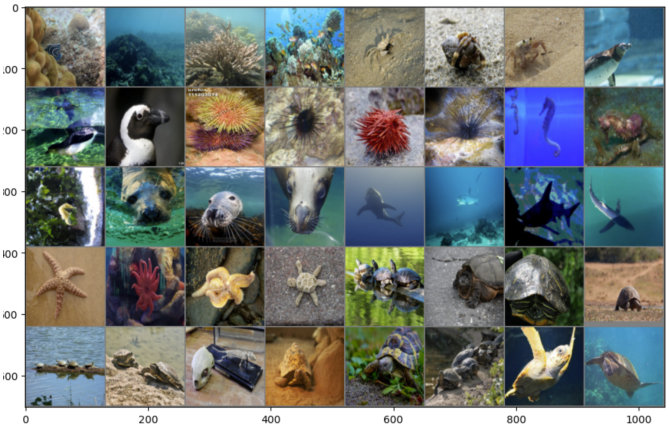

Fig. 1. Data

B. Preprocessing

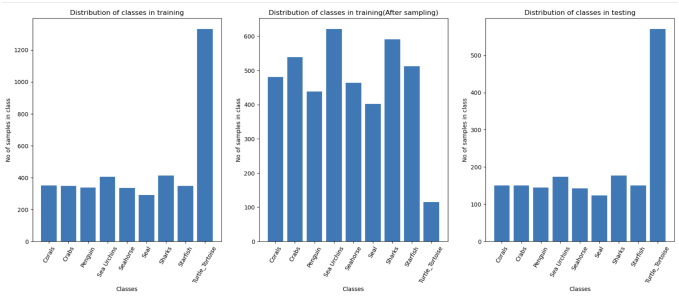

Fig. 2. Data Sampling

All the images are resized to 128x128 from 300x300 and normalized. The images are

1) Sampling: Most machine learning algorithms assume that the data is evenly distributed within classes. In the case of class imbalance problems, the extensive issue is that the algorithm will be more biased towards predicting the majority class. The algorithm will not have enough data to learn the patterns present in the minority class.

Most machine learning algorithms are not very useful with biased class data. But, we can modify the current training algorithm to take into account the skewed distribution of the classes. This can be achieved by giving different weights to both the majority and minority classes. The difference in weights will influence the classification of the classes during the training phase. The whole purpose is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class [2].

In this paper weighted Random Sampler from PyTorch [3] is used to balance the data set. From the initial distribution in the first graph of figure 2 we can see that the tortoise class has many records as compared to other classes Therefore we need to balance the data set so that the models do not over train on one of the classes. Steps of sampling are given below.

1) Total number of unique classes(N) and there counts(c_i) is obtained.
2) Class weights are calculated $w_i = \frac{\sum_{i=1}^{N} c_i}{c_i}$ for each class. Therefore we get weights for each class.
3) Pass these weights as parameter to $WeightedRandomSampler()$ method in PyTorch.

The results after sampling are in figure 2 where the classes are more balanced than before. The test set is not sampled as it is not contribute towards training. Other approaches are also available to sample data set to make it balanced.

C. SVM with HOG features

Animals are a kind of items with a wide range of colour. Shape and other structural indicators provide a more robust representation. Specific direction gradients can capture a certain shape idea. Histogram of Oriented Gradients (HOG) is employed in order to accommodate some shape diversity.

1) Histogram of Gradients: The concept behind HOG is that we arrange the pixels into small cells rather than using each pixel's specific gradient direction. We compute all of the gradient directions for each cell and divide them into various orientation bins [4]. We total the gradient strength for each sample. Therefore, the effects of small random orientations caused by noise are diminished and stronger gradients contribute more weight to their bins. We can see the dominating orientation of the cell from this histogram. We can visualise the image's structure by repeating the process for each cell. The HOG features maintain an object's distinct representation while also allowing for some form fluctuations.

$$Magnitude = \sqrt{G_x^2 + G_y^2} \tag{1}$$

$$Angle = \arctan \frac{G_y}{G_x} \tag{2}$$

2) Feature Extraction: We can specify the number of orientations, pixels_per_cell , and cells_per_block for computing the HOG features of a single channel of an image. The number of orientations is the number of orientation bins that the gradients of the pixels of each cell will be split up in the histogram [5]. The pixels_per_cells is the number of pixels of each row and column per cell over each gradient the histogram is computed. The cells_per_block specifies the local area over which the histogram counts in a given cell will be normalized. In our project we have used 6x6 pixels per cell and 3x3 cells per block with 9 orientations.

3) Support Vector Machine: The classification Support Vector Machine (SVM) using kernel functions for a projection of all the initial features in a new space for large dimension. This projection aims at separating the linear data in the new space. Subsequently, the SVM classifier seeks a linear separator in this new space which transforms itself into a nonlinear separator in the original space. The SVM has been applied in text classification, recognition of humanitarian actions, pedestrians and recognition of objects, etc. The SVMs were introduced by Vapnik [7], this classifier tries to locate a decision boundary by allowing linear separation of the examples of the first class with the examples of the second class in the learning set. The principle of these methods is to find the best hyper plan that maximizes the gap between learning examples and hyper plan.

Kernel Function The function of kernel is to take data as input and transform it into the required form.Different SVM algorithms use different types of kernel functions. These functions can be different types. For example linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid. Radial basis function (RBF) is a function whose value depends on the distance (usually Euclidean distance) to a center (x_c) in the input space. The most commonly used RBF is Gaussian [8]. It has the same form as the kernel of the Gaussian probability density function and it is defined as

$$RBF(x, x_c) = \exp - \frac{||x - x_x||^2}{2\sigma^2} \tag{3}$$

Decision Function The classifier, SVC, and Logistic Regression methods of the Sklearn machine learning framework include the decision function. This method essentially produces a Numpy array, where each member indicates how far a predicted sample for x_test by the classifier is from the Hyperplane and whether it is projected to be on the right or left side of the Hyperplane. Additionally, it indicates the degree to which each value for x_test that the classifier predicts is Positive (large-magnitude Positive value) or Negative (large-magnitude Negative value) with certainty [9].

4) Training Steps: After feature extraction the model is trained for 1000 epochs with gamma 0.001 and rbf kernel and ovr decision function.

1) Pre-process the training images.
2) Extract features using Histogram of Gradients
3) Set SVM parameters
4) Train the features using SVM
5) Predict the results

D. SVM with SIFT features

Image feature extraction is a key process in image retrieval, and SIFT can extract distinct local features, which is used in the paper. SIFT has been presented in 1999 by Lowe [11],

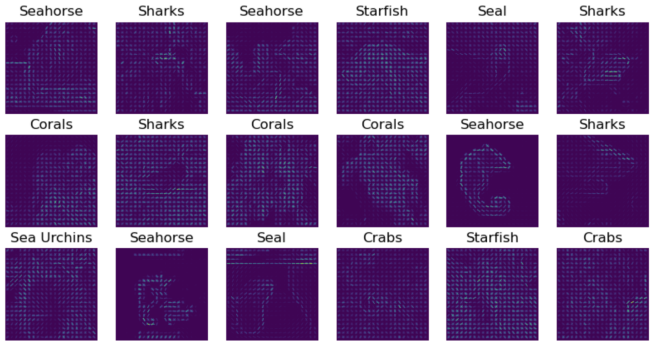

Fig. 3. HOG Images

which is a local descriptor. We first detect the extrema in Scale space, and then extract the invariant descriptors to the location, scale and rotation [12].

1) Scale Invariant Feature Transform(SIFT): The SIFT features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint [14]. The process of calculating SIFT features is [15]:

- **Scale Space Extrema Detection:** The Laplacian of Gaussian extrema are then replaced by the extrema of Difference of Gaussia functions, which are the local maximum and minimum points invariant to scale and rotation, in order to efficiently detect stable keypoint locations in scale space. This is the first step of the SIFT algorithm.
- **Keypoint Localization:** In order to achieve sub-pixel accuracy, the interest point position is slightly corrected by a quadratic interpolation. And then, points that have low contrast or are poorly localized along the edge are removed, that will have better performance by suppressing noise and greater similarity matching.
- **Orientation Assignment:** From each test picture, slope greatness and introduction is precomputed. An introduction histogram is alloted to each key point. Crests essentially compare to overwhelming headings of nearby slopes. Keypoints are made for each crest with comparative greatness, and the doled out introduction is refined by nearby quadratic introduction of the histogram values.
- **Keypoint Descriptor:** Slope tests are amassed into introduction histograms. Each of the 16 sub-regions compares to an 8- introductions containers histogram, driving to a 128-dimensional highlight for each keypoint.
- **Keypoint Matching:** When the SIFT feature vectors are extracted, it needs to match the keypoints. The traditional method is as follows, firstly, the Euclidean distance between the keypoints of the query image and the keypoints of the image in the database is calculated. Secondly, the minimum Euclidean distance (d_1) and the sub-minimum Euclidean distance (d_2) are detected, and then it needs to set a ratio ($ratio$) in advance, if ($\frac{d_1}{d_2} < ratio$) , this pair of keypoints is accepted. Finally, the number of the matched keypoints between two images is counted.

2) Bag of Visual Words (BoVW): The general idea of Bag of Visual Words (BOVW) is to represent an image as a set of features. Features consist of key points and descriptors. A key point is a prominent point in an image. Even if you rotate, enlarge, or reduce the image, the key points will not change. A descriptor is a description of a key point. We use key points and descriptors to build a vocabulary and represent each image as a frequency histogram of the features present in the image. You can later find other similar images or predict image categories from the frequency histogram. Steps in Bag of Visual Words Model [20]:

- **Feature Description:** The main task of feature extraction and description is to extract the representative of local characteristics as the description of image. Traditional method use SIFT descriptor to descriptor the image.
- **Dictionary generation:** The essence of visual dictionaries is primarily to share a whole characteristic space is appropriate. Same interval eigenvectors are taken to represent the same visual term area. K-means clustering algorithm classifies the SIFT functions into different classes, the center of each class is visual words. All visual words form a visual dictionary. The size of the visual dictionary is Number of visual words.
- **Characteristic Quantification:** We can get the visual words of image local characteristic by using the nearest search method to find the classification center in the visual words dictionary of characteristic quantification. We can obtain the histogram of image visual words according the frequency of each visual words, this process is to simplify representation of image.
- **Training Classifier:** Support vector machine (SVM) is a commonly and easily classifier in the field of classification.

3) Training Steps: After feature extraction the model is trained for 1000 epochs with optimization parameter set to 30.

1) Pre-process the training images.
2) Extract features using SIFT
3) Create bag of visual words model using K-means
4) Create histogram from local features by matching with bag of visual words.
5) Set SVM parameters
6) Train the BoW model using SVM
7) Predict the results

E. Pretrained ResNet-50

In this paper we have used ResNet-50 pre-trained model trained on ImageNet data set of 1.2 million images with 1000 classes [17]. The ImageNet data set contains some classes which are similar to our classes in the dataset and we would like to utilize the knowledge gained from ImageNet to train our model. The ResNets network architecture uses the Residual Learning method in its components. This method helps to

keep the values of the parameters from falling into a state of saturation (vanishing gradient) by adding the value of the shortcut [18].

The 50-layer ResNet architecture includes the following elements [19]:

- A 7×7 kernel convolution alongside 64 other kernels with a 2-sized stride.
- A max pooling layer with a 2-sized stride.
- 9 more layers—3×3,64 kernel convolution, another with 1×1,64 kernels, and a third with 1×1,256 kernels. These 3 layers are repeated 3 times.
- 12 more layers with 1×1,128 kernels, 3×3,128 kernels, and 1×1,512 kernels, iterated 4 times.
- 18 more layers with 1×1,256 cores, and 2 cores 3×3,256 and 1×1,1024, iterated 6 times.
- 9 more layers with 1×1,512 cores, 3×3,512 cores, and 1×1,2048 cores iterated 3 times.
- Average pooling, followed by a fully connected layer with 1000 nodes, using the softmax activation function.

1) Training Steps:

- Images are normalized and resized to 128x128 dimensions and are loaded into Pytorch Dataloader with specified batch size.
- ResNet-50 model pre-trained weights are loaded from Pytorch.
- **Final Layer:** The final linear layer of ResNet-50 pre-trained model is defined as $2048 \longrightarrow 512$ and $512 \longrightarrow 9$ with activation function as ReLu and dropout 0.2.
- While training the final layer on our data set the pre-trained weights are not updated as back-propogation is disabled for those layers.
- The data is trained for 30 epochs with learning rate of 0.001.
- Trained model is saved on disk.

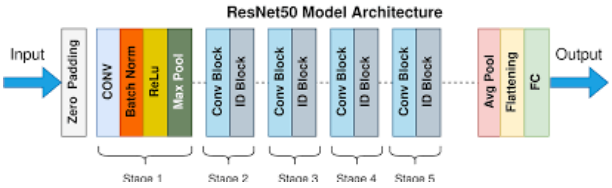

Fig. 4. ResNet50 - Architecture

F. Custom-CNN

The Custom-CNN model uses a 3 convolution layer architecture with intermediary pooling layers followed by 3 linear layers. The architecture of the model is as follows which takes a 128x128 RGB images as input :

- **Convolution Layer 1:** kernel size 11 with depth 16 and zero padding followed by batch normalization and ReLu activation function.
- **Pooling Layer 1:** size 3 and stride 2
- **Convolution Layer 2:** kernel size 7 with depth 32 and zero padding followed by batch normalization and ReLu activation function.
- **Pooling Layer 2:** size 3 and stride 2
- **Convolution Layer 3:** kernel size 3 with depth 32 and zero padding followed by batch normalization and ReLu activation function.
- **Linear Layer 1:** Linear layer with 0.5 dropout and ReLu activation connected to 256 neurons.
- **Linear Layer 2:** Linear layer with 0.5 dropout and ReLu activation connected to 128 neurons.
- **Final Linear Layer 3:** Linear layer connected to 9 neurons.

1) Training Steps:

- Images are normalized and resized to 128x128 dimensions and are loaded into Pytorch Dataloader with specified batch size.
- Model is trained on the training images.
- The data is trained for 50 epochs with learning rate of 0.001.
- Trained model is saved on disk.

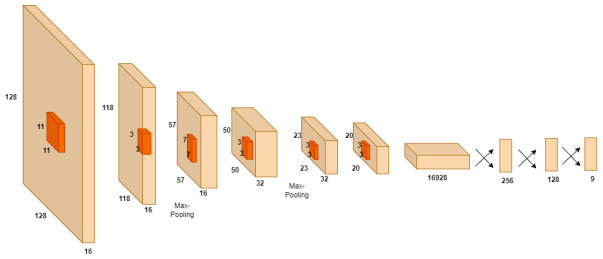

Fig. 5. Custom-CNN Architecture

IV. Evaluation and Results

Evaluating all the models on test set of 1784 images containing 9 classes the results of accuracy, f-score, recall and precision is presented in the table 1.

TABLE I
Experimental Results

Model/Metric	Accuracy	F-score	Recall	Precision
ResNet-50	64.4	0.692	0.844	0.595
Custom-CNN	73.6	0.78	0.896	0.765
SVM with HOG	46.33	0.492	0.595	0.437
SVM with SIFT	9.66	0.101	0.12	0.125

A. System Environment

The models were trained locally on a mid-range system with the following specifications:

- CPU: Intel 11th Gen i7
- GPU: Nvidia RTX 4GB
- RAM: 16GB memory

The pretrained models such as ResNet did not need training for the previous layers, but the final fully connected lavers would still have to be trained to work with the classes in the Sea animals dataset.

B. Evaluation Criteria

Evaluation metrics measure the quality of a model, they are used to check the correctness of the model. The metrics make use of the predictions made by the model on held out testing set. While accuracy is one of the most useful metric there is for classification there are other metrics that also help in the evaluation of the model.

1) Accuracy: It is the most intuitive performance measure, and it is simply a ratio of correctly predicted observation to the total observations. However, only the accuracy is not sufficient to predict the performance of the CNN model, since the type of dataset (symmetric vs non-symmetric) might affect the accuracy.

$$Accuracy = \frac{TP + TN}{TP + TF + FP + FN} \tag{4}$$

2) Precision: It is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision rate indicates low false positive rate.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

3) Recall: It is the ratio of correctly predicted positive observations to all observations in actual class. Generally, the recall rate of 0.5 and above is considered good.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

4) F-Scores: It is the weighted average of Precision and Recall i.e., it takes both false positives and false negatives into account for predicting the correctness of the model

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{7}$$

C. SVM with HOG features

The model trained on histogram of gradients using SVM achieves accuracy of 46.33%. The accuracy of the model is at lower end however the model training time was quicker as compared to training time of deep learning models. From confusion matrix represented in figure 6 the model is able to predict some of the classes correctly and confuses with most of the training images. It fails to classify tortoise class. Histogram of gradients rely on the structural cues of the images. From observing HOG images in figure 3 we can see that some of the HOG images of corals and starfish look similar and other classes are also similar to each other. Therefore the model is not able to predict some of the images correctly where the images have similar structure in them. Background of the images such as shape of land surrounding corals in many images affect the performance of the model as it confuses the classes with coral class. The performance of the model can be improved by better preprocessing of images however it might still be inferior to deep learning models.

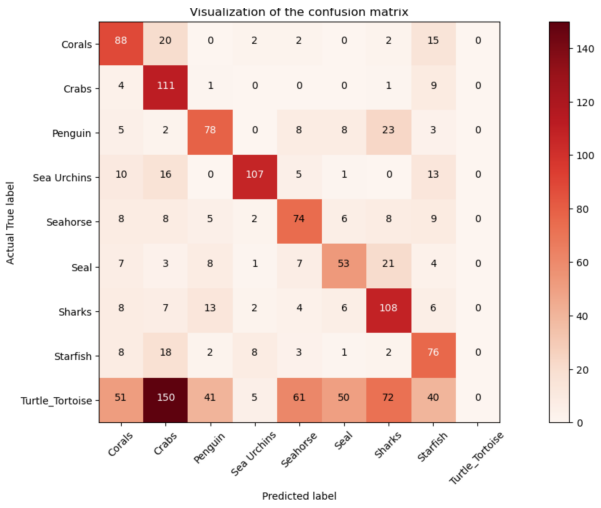

Fig. 6. SVM-HOG Confusion Matrix

D. SVM with SIFT features

The model trained on SIFT features using SVM achieves accuracy of 9.66% which is the lowest accuracy achieved by solution approaches presented in our paper. The model completely fails to classify any of the given classes which can be inferred from confusion matrix in figure 7. SIFT features rely on the appearance of objects and are scale and rotation in variant. However in this case the images of sea animals are surrounded by water and corals and not all the animals are visible fully from appropriate angle. SIFT features in this case might be similar for classes and therefore the model is not able to distinguish properly among classes. The accuracy of the model can be improved if the sea animals are extracted from images and all the background is removed except the animal.

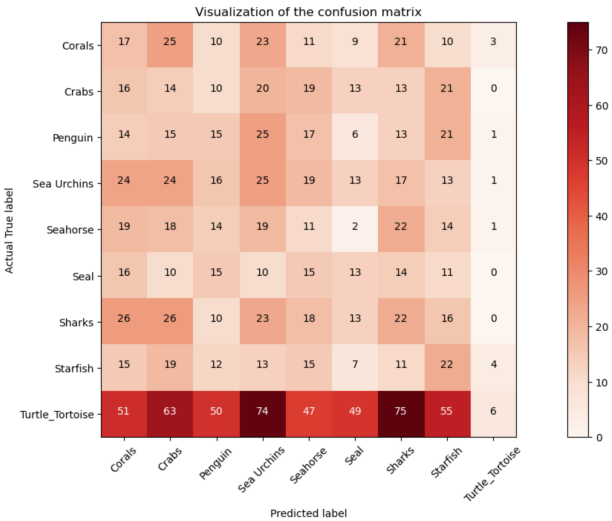

Fig. 7. SVM-SIFT Confusion Matrix

E. Pre-trained ResNet-50

The ResNet50 pre-trained model gives accuracy of 64.4% and precision 0.595. The model achieves high accuracy and low training loss in initial epochs represented in figure 8 as compared to Custom-CNN model which gradually attains high accuracy. The model performs better than traditional machine learning models. The model results as represented in confusion matrix in figure 9 is able classify all the classes correctly except tortoise class as it completely fails to classify tortoise class correctly. The reason could be after looking at the data closely the tortoise data have very less images of tortoise as a whole animal and most of the images contain either the face or the back of the tortoise. The sampling method used have also further reduced the number of training images of tortoise while balancing data classes. This issue can be addressed by improving the sampling method and adding more relevant images in the tortoise class.

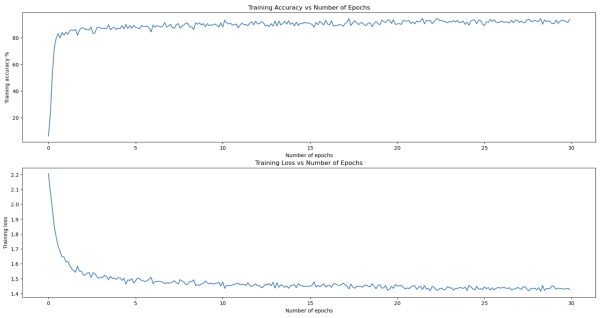

Fig. 8. ResNet50-Training Loss and Accuracy

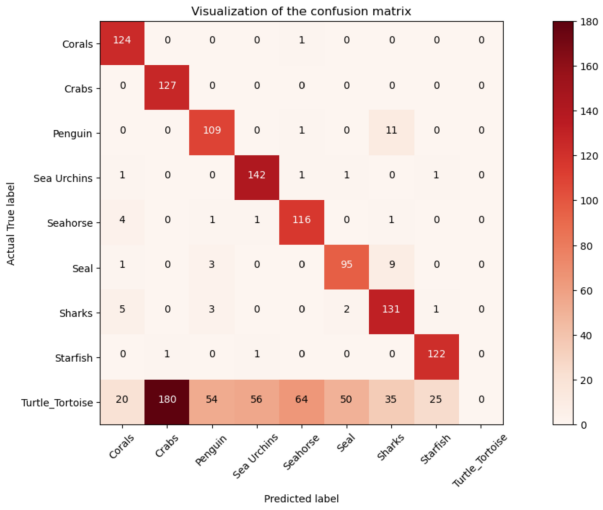

Fig. 9. ResNet50 Confusion Matrix

F. Custom-CNN

The vanilla Custom-CNN model performs the best among all the four approaches with accuracy of 73.6%. After observing the confusion matrix in figure 11 we can see that the model is able to predict all the classes correctly except the tortoise class. The model confuses tortoise class with all the classes. The reason could be after looking at the data closely the tortoise data have very less images of tortoise as a whole animal and most of the images contain either the face or the back of the tortoise. The sampling method used have also further reduced the number of training images of tortoise while balancing the data set and therefore only a few classes are classified correctly with the test data set. This issue can be addressed by improving the sampling method and adding more relevant images in the tortoise class. It can be observed that the training loss and accuracy in figure 10 are decreasing and increasing respectively as the epochs increase.

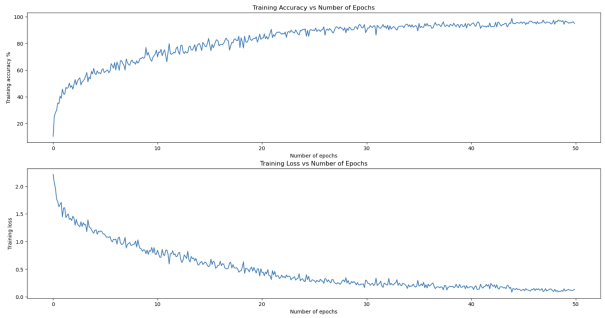

Fig. 10. CustomCNN-Training Loss and Accuracy

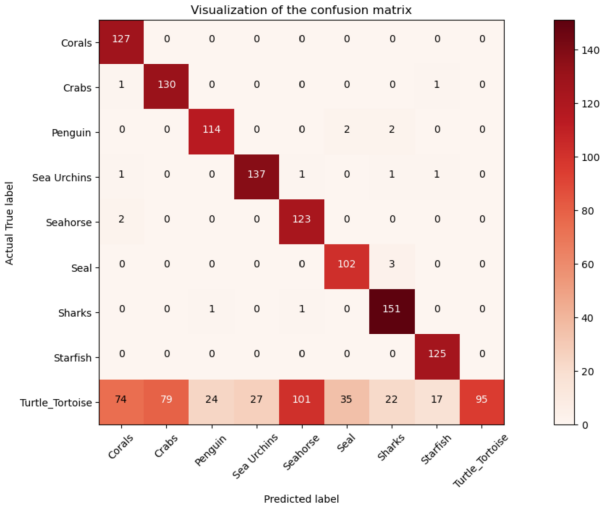

Fig. 11. Custom-CNN Confusion Matrix

V. Conclusion

Deep Learning architectures are proven to effective than traditional machine learning models in classifying sea animals. Deep learning architectures are able to identify semantic features from images which the traditional hand crafted features are not able to capture. The traditional features like HOG and SIFT approaches rely on the shape, orientation and color of the object. In sea animals classification the images actual class was surrounded by other things like the water and corals which affect the performance of traditional machine

learning approach. SIFT approach completely fails in this scenario and is probably more suitable for natural objects rather than sea animals classification. Histogram of gradient is a promising feature and the performance of the model can be used with better pre-processing of images. However when it comes to train images on raw pixels deep learning approaches are proven to be very effective. Transfer learning approach using ResNet50 has reasonable accuracy however custom CNN architecture performed better than ResNet50 and the performance of custom-CNN architecture can be improved by improving the architecture design and parameters of the neural network. Probably a deep neural network can be suitable for the task. Hyper parameter tuning can slightly improve the performance of the model. Transfer learning approach can be effective if the source and target data set are similar in nature. In most cases deep learning outperforms when it comes to image classification problems.

REFERENCES

[1] Sea Animals Classification, https://www.kaggle.com/datasets/vencerlanz09/sea-animals-image-dataste, 2023

[2] Kamaldeep Singh, How to Improve Class Imbalance using Class Weights in Machine Learning, https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/, 2020.

[3] Documentation PyTorch, https://pytorch.org/docs/stable/data.html#torch.utils.data.Sampler, 2023

[4] M. R. Guedira, A. E. Qadi, M. R. Lrit and M. E. Hassouni, "A novel method for image categorization based on histogram oriented gradient and support vector machine," 2017 International Conference on Electrical and Information Technologies (ICEIT), Rabat, Morocco, 2017, pp. 1-5, doi: 10.1109/EITech.2017.8255229.

[5] L. Mao and L. Tang, "Pedestrian Detection Based on Gradient Direction Histogram," 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 2022, pp. 939-943, doi: 10.1109/IPEC54454.2022.9777626.

[6] Siddiqui, Shoaib Salman, Ahmad Malik, Imran Shafait, Faisal Mian, Ajmal Shortis, Mark Harvey, Euan. (2017). Automatic fish species classification in underwater videos: Exploiting pretrained deep neural network models to compensate for limited labelled data. ICES Journal of Marine Science. 75. 10.1093/icesjms/fsx109.

[7] C. Cortes and V. Vapnik, Support-vector network, Machine Learning, vol. 20, pp. 273297

[8] H. Chen, B.R. Bakshi, Chapter 3.12.3, Comprehensive Chemometrics Chemical and Biochemical Data Analysis, 2009

[9] Linear Classifier SVM, https://gawron.sdsu.edu/python_for_ss/course_core/book_draft/text/linear_classifier_svm.html

[10] Applications of Machine Learning in Chemical and Biological Oceanography. Balamurugan Sadaiappan, Preethiya Balakrishnan, Vishal C.R., Neethu T. Vijayan, Mahendran Subramanian, and Mangesh U. Gauns.

[11] D. G. Lowe. Distinctive image features from sacle-invariant keypoints., International Journal of Computer Vision, vol. 60, pp. 91- 110, 2004

[12] G. M. Wang and X. W. Chen. Study on a New Algorithm of Feature Matching-SIFT, Journal of Yancheng Institute of Technology, vol. 20, pp. 1-6, 2007

[13] Mahmood, Ammar Bennamoun, Mohammed An, Senjian Sohel, Ferdous Boussaid, F. Hovey, Renae Kendrick, Gary Fisher, Robert. (2016). Automatic annotation of coral reefs using deep learning. 1-5. 10.1109/OCEANS.2016.7761105.

[14] Scale Invariant Feature Transform, https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

[15] B. Li, X. Kong, Z. Wang and H. Fu, "SIFT-Based Image Retrieval Combining the Distance Measure of Global Image and Sub-Image," 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 2009, pp. 706-709, doi: 10.1109/IIH-MSP.2009.180.

[16] A Parallel Classification Model for Marine Mammal Sounds Based on Multi-Dimensional Feature Extraction and Data Augmentation. Wenyu Cai, Jifeng Zhu, Meiyan Zhang and Yong Yang.

[17] ResNet501. Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun

[18] Using SURF to Improve ResNet-50 Model for Poultry Disease Recognition Algorithm. Quach, Luyl-Da and Quoc, Nghi Pham and Thi, Nhien Huynh and Tran, Duc Chung and Hassan, Mohd Fadzi.

[19] ResNet-50: The Basics and a Quick Tutorial. Datagen.

[20] L. Zhi-Jie, "Image Classification Method Based on Visual Saliency and Bag of Words Model," 2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), Nanchang, China, 2015, pp. 466-469, doi: 10.1109/ICICTA.2015.122.