Professor Kevin Gold
12/09/24

DS340 Final Project: Sea Animal Classification

Rachel Li, Polly Peng

1. Introduction

Identifying marine species is a critical task in ecological conservation, yet manual classification is time-consuming and error-prone. This project explores the use of convolutional neural networks (CNNs) to automate species identification, aiming to achieve accurate classification while addressing challenges such as class similarity and computational limitations. By leveraging deep learning, we aim to streamline biodiversity monitoring and support conservation efforts with scalable, efficient solutions. This report evaluates three CNN models: **Custom CNN, VGG16, and ResNet50,** highlighting their effectiveness in handling the complexities of marine species classification and their potential impact on ecological research.

2. Dataset Description

The Seadata dataset consists of 13,000 labeled images across 23 marine species, with balanced class distributions. Each image is resized to 224×224 pixels and normalized to a [0, 1] range. Labels are encoded using a **LabelEncoder**. An 80-20 train-test split is applied to evaluate the model, ensuring stratification to preserve class distributions. Two models are trained and tested on the same dataset, ensuring consistency and comparability.

3. Methodology

3.1 CNN Model Architectures

In this project, we experimented with three custom CNN architectures to classify sea animals effectively. The first attempt involved a simple model with three convolutional layers as our benchmark. The training accuracy is 98% and validation 31%. This indicates that we are overfitting the model. For the second attempt, we impletmend batch normalization, max-pooling, and data augmentation techniques such as random flipping, rotation, and zooming to increase the diversity of the training data. This model was designed to extract basic features but struggled with complex patterns, achieving a training accuracy of

```
Model Accuracy: 25.75%
Classification Report:
                precision    recall  f1-score   support

         Clams       0.13      0.13      0.13       100
        Corals       0.22      0.17      0.19       100
         Crabs       0.38      0.24      0.29       100
       Dolphin       0.40      0.29      0.34       157
           Eel       0.12      0.01      0.02       100
          Fish       0.50      0.01      0.02        99
    Jelly Fish       0.48      0.75      0.58       169
       Lobster       0.06      0.02      0.03       100
   Nudibranchs       0.00      0.00      0.00       100
       Octopus       0.00      0.00      0.00       113
         Otter       0.33      0.29      0.31       100
       Penguin       1.00      0.03      0.06        97
       Puffers       0.07      0.04      0.05       107
      Sea Rays       0.15      0.05      0.07       104
   Sea Urchins       0.12      0.82      0.21       116
     Seahorse       0.35      0.06      0.11        96
          Seal       0.30      0.10      0.15        83
        Sharks       0.18      0.09      0.12       118
        Shrimp       0.17      0.01      0.02        98
         Squid       0.00      0.00      0.00        97
      Starfish       0.23      0.05      0.08       100
...
      accuracy                           0.26      2750
     macro avg       0.25      0.19      0.16      2750
  weighted avg       0.27      0.26      0.20      2750
```

34% and a validation accuracy of 3%, no longer overfitting. Latly, we implemented the model to an unseen test dataset, resulting in 25.75% accuracy.

Analysis of Model Performance and Recommendations for Improvement

The classification results reveal several key metrics that provide insight into the model's performance. **Precision** measures the proportion of correctly predicted instances out of all predictions for a particular class. For example, the precision for the **Fish** class is 0.50, meaning half of the instances predicted as **Fish** were correct. However, classes like **Nudibranchs, Octopus,** and **Squid** have a precision of 0.00, indicating that the model failed to make any correct predictions for these classes. This suggests that the model struggles significantly with these categories.

Another metric, **recall**, evaluates the proportion of actual instances of a class that were correctly identified. For example, **Sea Urchins** has a recall of 0.82, demonstrating that the model correctly identified 82% of the actual **Sea Urchins** samples. Conversely, the recall for classes like **Eel**, **Fish,** and **Shrimp** is close to 0.01, indicating that the model identified almost none of the actual samples for these categories.

Moreover, the **F1-score** helps us to see a balanced measure of performance. Higher F1-scores indicate better overall classification for a class. For instance, the **JellyFish** class has an F1-score of 0.58, reflecting relatively strong performance, while **Nudibranchs** and **Octopus** have an F1-score of 0.00, showing complete failure for these categories.

Key Observations

Certain classes, such as **Sea Urchins** (high recall of 0.82) and **JellyFish** (F1-score of 0.58), stand out with relatively good performance. These results suggest that the model has learned some class-specific features for these categories. In contrast, the model performed poorly for several classes, including **Nudibranchs**, **Octopus**, and **Squid**, where all metrics are 0.00, indicating complete failure to classify these correctly. Despite having balanced support across classes, the model’s performance remains inconsistent, likely due to its inability to effectively learn distinguishing features for certain categories.

Possible Reasons for Poor Performance

1. **Model Complexity**: The current CNN architecture might lack the complexity needed to effectively learn features for all classes.
2. **Class Similarity**: Some classes, such as **Shrimp** and **Lobster**, may have visually similar patterns, making them difficult to differentiate.
3. **Dataset Issues**: The dataset might include noisy or low-quality images, reducing the model’s ability to generalize.

4. **Training Data Size**: An insufficient number of training samples might limit the model's ability to learn representative features.

Model 2: VGG16

The VGG16 architecture is a widely used convolutional neural network known for its straightforward structure, utilizing a series of small convolutional filters to extract hierarchical features. Leveraging its pre-trained weights on the ImageNet dataset, VGG16 serves as a reliable starting point for transfer learning, especially for smaller datasets like Seadata.

During our experiments, VGG16 achieved a test accuracy of **31.96%**, reflecting its limitations in handling the diverse and complex patterns present in the Seadata dataset. While the model captured some basic features, it struggled to generalize across all 23 marine species, as evidenced by its higher loss values during evaluation.

Model 3: ResNet50

To address the challenges faced by VGG16, we adopted ResNet50, a deeper convolutional neural network equipped with residual connections. These connections alleviate the vanishing gradient problem, enabling ResNet50 to learn effectively from complex datasets. Its ability to extract deeper features and maintain efficiency makes it particularly suitable for image classification tasks.

ResNet50 outperformed VGG16 significantly, achieving a test accuracy of **76.62%**. This improvement highlights ResNet50's capability to generalize better across diverse categories of marine species. The lower test loss of **1.0264** further emphasizes its effectiveness compared to VGG16, which had a loss of **2.4742**.

Pre-trained and Fine-tuned Models

Both VGG16 and ResNet50 were implemented as pre-trained models, initialized with **ImageNet weights**, and fine-tuned to adapt to the Seadata dataset. The fine-tuning involved unfreezing the last 20 layers of each architecture, enabling the models to learn dataset-specific features.

We employed the same data augmentation techniques for both models:

- Random rotation
- Horizontal flipping
- Zooming
- Normalization

Hyperparameter	VGG16	ResNet50
Input Shape	(224, 224, 3)	(224, 224, 3)
Optimizer	Adam(lr=1e-4)	Adam(lr=1e-4)
Loss Function	categorical_crossentropy	categorical_crossentropy
Dropout Rate	0.5	0.5
Dense Layer Units	1024	1024
Trainable Layers	Last 20	Last 20
Epochs	**5**	**10**
Batch Size	32	32

Performance Comparison

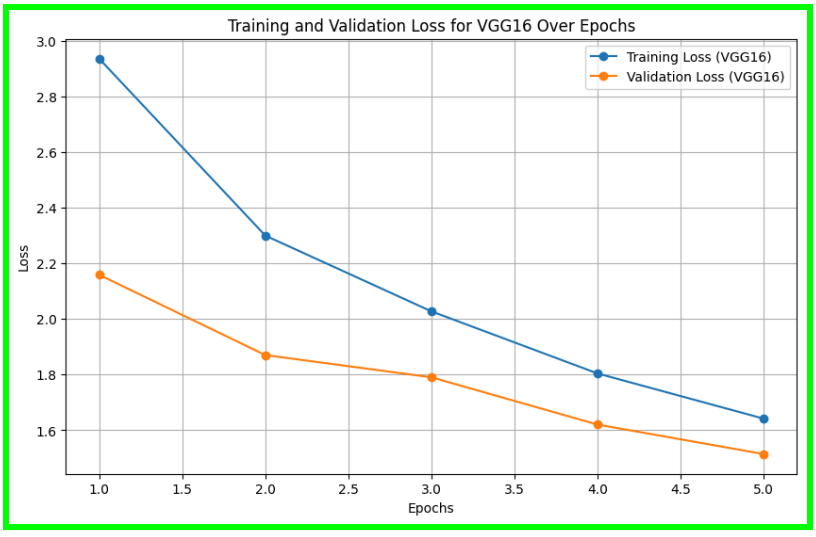
The test results highlight the stark difference in performance between the two models:

- ResNet50: 76.62% test accuracy, 1.0264 test loss
- VGG16: 31.96% test accuracy, 2.4742 test loss

These results illustrate the benefits of adopting deeper architectures like ResNet50, which excel at learning complex and diverse patterns.

4. Results

Model 2: VGG16



The training and validation loss graphs for VGG16 show a consistent decrease over the initial epochs, reflecting the model's ability to learn from the data. The training was originally set to run for 10 epochs; however, analysis revealed that improvements in validation accuracy plateaued in later epochs, providing limited gains in performance. To address this, we incorporated an early stopping mechanism into the code, allowing the training to halt automatically once the validation

loss stopped improving. As a result, the training process now concludes at **epoch 5**, optimizing computational efficiency without sacrificing model accuracy.

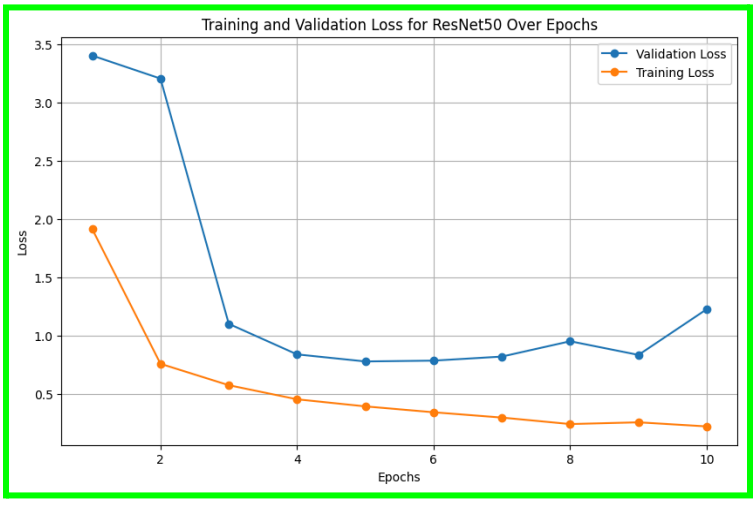
Classification Report for VGG16:

	precision	recall	f1-score	support
Clams	0.14	0.19	0.16	100
Corals	0.15	0.26	0.19	100
Crabs	0.14	0.07	0.09	100
Dolphin	0.31	0.69	0.42	157
Eel	0.12	0.01	0.02	100
Fish	0.00	0.00	0.00	99
Jelly Fish	0.63	0.63	0.63	169
Lobster	0.25	0.01	0.02	100
Nudibranchs	0.07	0.06	0.07	100
Octopus	0.25	0.02	0.03	113
Otter	0.74	0.55	0.63	100
Penguin	0.38	0.14	0.21	97
Puffers	0.20	0.03	0.05	107
Sea Rays	0.14	0.27	0.18	104
Sea Urchins	0.46	0.84	0.59	116
Seahorse	0.33	0.05	0.09	96
Seal	0.00	0.00	0.00	83
Sharks	0.00	0.00	0.00	118
Shrimp	0.28	0.17	0.22	98
Squid	0.12	0.03	0.05	97
Starfish	0.08	0.11	0.09	100
Turtle_Tortoise	0.37	0.93	0.53	381
Whale	0.29	0.14	0.19	115
accuracy			0.32	2750
macro avg	0.24	0.23	0.19	2750
weighted avg	0.26	0.32	0.25	2750

The VGG16 model achieved an overall accuracy of 32%, demonstrating a slight improvement over the benchmark accuracy of 25.75%. While some classes performed relatively well, such as **JellyFish (F1-Score: 0.63)** and **Sea Urchins (F1-Score: 0.59)**, others, including **Fish (F1-Score: 0.00)** and **Seal (F1-Score: 0.00)**, faced significant misclassification challenges. The macro average F1-score of 0.19 and weighted average F1-score of 0.25 highlight the model's limited ability to generalize effectively across all classes. Additionally, VGG16 struggled to consistently capture the complex features in the dataset, particularly for underrepresented or visually similar classes.

Given these results, the improvement from the benchmark was not substantial enough to justify relying on VGG16. This led us to explore more advanced architectures, such as ResNet50, which offers better opportunities for performance improvement. ResNet50's use of residual connections and deeper architecture makes it more capable of learning intricate patterns and generalizing across diverse datasets, positioning it as a stronger candidate for this classification task.

Model 3: ResNet50



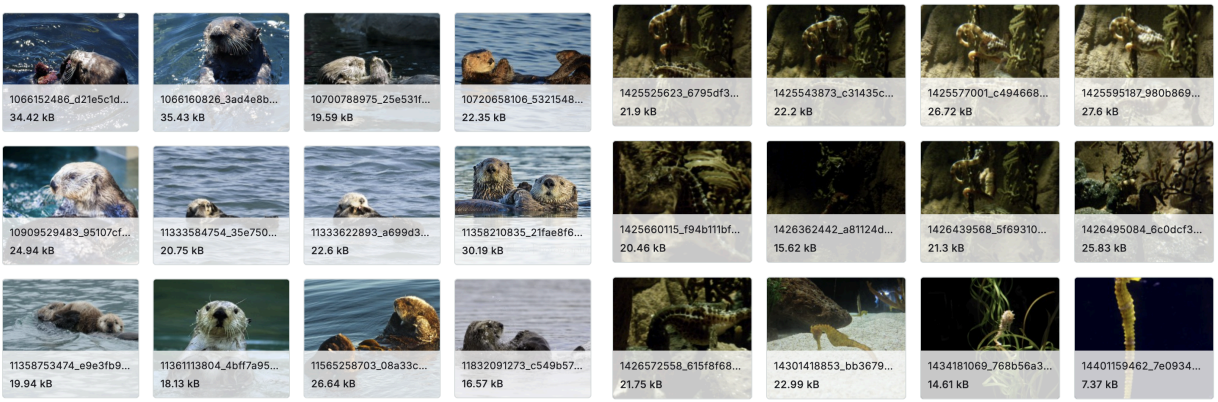
The training loss follows a smooth downward trend, indicating that the model is successfully optimizing the parameters without interruption. The validation loss, on the other hand, shows a sharp decrease in the first few epochs, stabilizing around Epoch 4 before beginning to fluctuate slightly in later epochs.

This fluctuation in validation loss, coupled with a continued decrease in training loss, **suggests a slight overfitting** in the later stages of training. To address this, we incorporated model checkpoints into the code, enabling the selection of the model with the lowest validation loss during the 10 epochs. This approach ensures the best-performing model is chosen, balancing training performance with generalization to unseen data. Overall, ResNet50 exhibits a significant improvement over VGG16 in both stability and accuracy, making it the better choice for this task.

Classification Report for ResNet50:

	precision	recall	f1-score	support
Clams	0.63	0.67	0.65	100
Corals	0.53	0.80	0.64	100
Crabs	0.97	0.86	0.91	100
Dolphin	0.84	0.81	0.82	157
Eel	0.63	0.67	0.65	100
Fish	0.65	0.43	0.52	99
Jelly Fish	0.91	0.96	0.93	169
Lobster	0.72	0.74	0.73	100
Nudibranchs	0.75	0.72	0.73	100
Octopus	0.70	0.46	0.56	113
Otter	0.98	0.99	0.99	100
Penguin	0.94	0.74	0.83	97
Puffers	0.79	0.49	0.60	107
Sea Rays	0.72	0.68	0.70	104
Sea Urchins	0.98	0.96	0.97	116
Seahorse	0.91	0.52	0.66	96
Seal	0.85	0.48	0.62	83
Sharks	0.90	0.51	0.65	118
Shrimp	0.60	0.76	0.67	98
Squid	0.74	0.77	0.76	97
Starfish	0.90	0.98	0.94	100
Turtle_Tortoise	0.76	0.98	0.86	381
Whale	0.56	0.87	0.68	115
accuracy			0.77	2750
macro avg	0.78	0.73	0.74	2750
weighted avg	0.78	0.77	0.76	2750

The ResNet50 model achieved a significant improvement in overall accuracy, reaching **76.62%.** The metrics indicate strong performance across most classes, particularly for **Otter**, which achieved an impressive **F1-score of 0.99**. This high performance can be attributed to the clear, high-quality images available in the dataset, which provide distinct visual features for the model to learn. Additionally, otters possess unique characteristics, such as facial structure and body outline, that likely made classification easier.



Otter Dataset VS Seahourse Dataset

On the other hand, **Seahorse** emerged as one of the worst-performing classes, with an F1-score of 0.66. This is likely due to several factors, including the presence of low-quality or poorly lit images in the dataset and the seahorse's complex and camouflaged features, which can resemble other marine species. These challenges make it harder for the model to distinguish seahorses from similar-looking classes.

Overall, the ResNet50 model demonstrates great ability to generalize across diverse marine species, particularly for classes with distinct visual features and higher-quality image data. However, the results also highlight the **importance of dataset quality and balanced representation** to achieve consistent performance across all classes.

5. Conclusion:

Our study highlights the potential of deep learning models for marine species classification using the Seadata dataset of **13,000 images across 23 species.**

- **Custom CNN**: Achieved 25.75% test accuracy but struggled with overfitting and limited generalization.
- **VGG16**: Improved test accuracy to 31.96%, but high loss (2.4742) indicated challenges with complex patterns.
- **ResNet50**: Outperformed both models with 76.62% test accuracy and a loss of 1.0264, leveraging residual connections to generalize effectively across diverse classes.

With its robust performance, ResNet50 demonstrates the viability of automating species identification to accelerate biodiversity monitoring and conservation efforts, offering scalable solutions for ecological challenges.

Work Cited

- Sea Animals Image Dataset
- ChatGPT 4o was used for the project