Marine Species Classification Using Deep Learning: Comparative Study

Arshdeep Kaur
Centennial College
Toronto, Canada
akau1031@my.centennialcollege.ca

Varun Rohit Mudunuri
Centennial College
Toronto, Canada
vmudunur@my.centennialcollege.ca

Jaekyeong Jang
Centennial College
Toronto, Canada
jjang35@my.centennialcollege.ca

Younghun Mun
Centennial College
Toronto, Canada
ymun1@my.centennialcollege.ca

Miguel Chadid Rendon
Centennial College
Toronto, Canada
mchadidr@my.centennialcollege.ca

Yajushi Garg
Centennial College
Toronto, Canada
ygarg3@my.centennialcollege.ca

***Abstract*— Marine biodiversity monitoring requires accurate automated species classification, but deep learning models face challenges with limited training data and visually similar marine organisms. This study compares three deep learning approaches for marine species classification. We curated a balanced dataset by selecting 10 classes from an original 23-class marine animals dataset, resulting in 5,978 images with a 70/15/15 train/validation/test split. We evaluated three experimental approaches: (1) custom convolutional neural networks trained from scratch, (2) unsupervised pre-training using autoencoders, and (3) transfer learning from ImageNet-pretrained models. Custom CNNs achieved 50.11% validation accuracy, limited by the moderate dataset size. Autoencoder-based pre-training performed worse than supervised learning by 8-11%, showing that image reconstruction does not align well with classification objectives. Transfer learning with ResNet50 and EfficientNet-B0 achieved the best results at 91.82% and 90.13% accuracy, representing 27-41% improvement over training from scratch. Confusion analysis revealed systematic errors between biologically similar species, particularly dolphins and whales (both cetaceans) and sea rays and sharks (both cartilaginous fish). Our results demonstrate that transfer learning provides substantial advantages for marine classification with limited data, while identifying domain-specific challenges that persist even at high accuracy. This work quantifies the performance gap between different deep learning paradigms for fine-grained marine species recognition.**

Keywords*— *Marine species classification, deep learning, transfer learning, convolutional neural networks, autoencoder, fine-grained recognition, underwater imagery.

I. Introduction

Marine biodiversity plays a critical role in ecosystem health, climate regulation, and food security, yet monitoring marine species populations remains challenging due to the vast scale and inaccessibility of ocean environments [1]. Traditional species identification methods rely on manual classification by marine biologists, which is time-consuming, expensive, and difficult to scale for large-scale monitoring programs. Automated species classification systems powered by deep learning offer a promising solution for rapid, accurate identification of marine organisms from underwater imagery [2].

Deep learning, particularly convolutional neural networks (CNNs), has demonstrated remarkable success in image classification tasks, achieving human-level or superior performance in many domains [3]. However, applying these methods to marine species classification presents unique challenges. First, collecting and annotating large-scale marine datasets is resource-intensive, resulting in limited training data compared to benchmark datasets like ImageNet [4]. Second, many marine species exhibit fine-grained visual similarities, with subtle morphological differences that are difficult for models to distinguish [5]. Third, underwater photography introduces additional complexity through variable lighting conditions, water turbidity, and background clutter .

While several approaches exist for training deep learning models with limited data, their relative effectiveness for marine species classification remains unclear. Custom CNNs trained from scratch offer full control over architecture but may struggle with small datasets. Unsupervised pre-training methods, such as autoencoders, aim to learn useful representations without labels but may not capture discriminative features. Transfer learning leverages knowledge from large-scale datasets like ImageNet, but the domain gap between terrestrial and marine images raises questions about its effectiveness.

This study presents a systematic comparison of three deep learning paradigms for marine species classification using a curated dataset of 10 marine animal classes selected from an original 23-class collection. Our contributions are threefold: (1) we quantify the performance gap between custom CNNs, autoencoder-based pre-training, and transfer learning approaches, (2) we analyze the failure modes of different methods to understand when and why they underperform, and (3) we identify biological and technical challenges that persist even at high classification accuracy. By providing empirical evidence across multiple approaches, this work offers practical guidance for researchers developing marine species classification systems with limited training data.

II. Methodology

A. Dataset Preparation

We used the Sea Animals Image Dataset from Kaggle, which originally contained 13,700 images across 23 marine species classes. To address class imbalance and create a more balanced dataset for training, we selected 10 classes based on two criteria: adequate sample size ($\geqslant$500 images per class) and taxonomic diversity to represent different marine animal groups. The selected classes were: Corals, Dolphin, Jelly Fish, Nudibranchs, Octopus, Puffers, Sea Rays, Sea Urchins, Sharks, and Whale.

The final curated dataset contained 5,978 images with class sizes ranging from 500 to 845 images, resulting in an imbalance ratio of 1.69x (maximum class size divided by

minimum class size). We split the data using a 70/15/15 ratio, yielding 4,181 training images, 892 validation images, and 905 test images. Data quality checks confirmed zero corrupted images with an average resolution of 289×222 pixels. All images were RGB/RGBA format from underwater photography with varying lighting conditions and backgrounds.

B. Experiment 1: Supervised Learning with Custom CNNs

We designed and trained three custom CNN architectures from scratch to establish supervised learning baselines and evaluate the impact of model complexity on performance with limited data.

1. Baseline CNN: A simple four-block architecture with progressively increasing filters (64→128→256→512), batch normalization, dropout (0.25-0.5), and max pooling after each convolutional block. The classifier consisted of three fully connected layers (1024→512→10) with dropout regularization. Total parameters: 104.8M.
2. Medium CNN: An improved architecture with four convolutional blocks and residual-style connections to facilitate gradient flow. This model used moderate filter sizes (64→128→256→512) with global average pooling before classification layers (512→256→10). Batch normalization and dropout (0.3-0.5) were applied throughout. Total parameters: 5.2M.
3. VGG-Style CNN: A deeper architecture inspired by VGG networks with five convolutional blocks using smaller 3×3 filters throughout. Multiple convolutional layers within each block increased model depth while maintaining computational efficiency. Total parameters: 15.3M.

All models used ReLU activation, Adam optimizer with initial learning rate of 0.001, categorical cross-entropy loss, and were trained for 40 epochs with early stopping (patience=10). Input images were resized to 224×224 pixels and normalized to [0,1] range.

C. Experiment 2: Unsupervised Pre-training with Autoencoders

To evaluate unsupervised representation learning, we implemented a convolutional autoencoder for pre-training feature extractors without using class labels.

1. Autoencoder Architecture: The encoder consisted of four convolutional blocks with progressively increasing channels (32→64→128→256), reducing spatial dimensions through max pooling to create a compressed latent representation. The decoder mirrored this structure, using transposed convolutions to reconstruct the input image. The bottleneck dimension was 256 channels at 14×14 spatial resolution. Training used mean squared error loss to minimize reconstruction error between input and output images.
2. Classification Phase: After autoencoder pre-training for 25 epochs, we extracted the encoder portion and attached a classification head. We evaluated two strategies: (a) Frozen Encoder where encoder weights remained fixed and only the classifier was trained for 25 epochs, and (b) Fine-tuned Encoder where the entire model was trained end-to-end for 30 epochs with learning rate 0.0001. This two-phase approach tested whether unsupervised feature learning could improve classification performance.

D. Experiment 3: Transfer Learning with State-of-the-Art Models

We evaluated two modern CNN architectures with and without ImageNet pre-training to quantify the benefit of transfer learning

1. Model Architectures: We selected ResNet50 (25.6M parameters) for its residual connections that enable very deep networks, and EfficientNet-B0 (5.3M parameters) for its efficient compound scaling approach [7]. Both models were configured for 10-class output by replacing the final classification layer.
2. Training Strategies: Each architecture was trained under two conditions:
   - From Scratch: Random weight initialization, trained for 40 epochs with learning rate 0.001, providing a baseline for comparison.
   - Transfer Learning: Two-phase approach starting from ImageNet-pretrained weights. Phase 1 (15 epochs): Froze the backbone network and trained only the classification head with learning rate 0.001. Phase 2 (25 epochs): Unfroze all layers and fine-tuned the entire network with reduced learning rate 0.0001.

Data augmentation for transfer learning included random horizontal flips, rotation (±30°), zoom (±15%), and width/height shifts (±10%) to improve generalization.

E. Training Configuration and Evaluation

All experiments used the following configuration: batch size 16 (optimized for GPU memory), Adam optimizer, categorical cross-entropy loss, and mixed-precision training for efficiency. Models were trained on NVIDIA GeForce RTX 4050 GPU. Learning rate scheduling reduced the rate by 50% after 8 epochs of no validation improvement.

We evaluated models using multiple metrics: validation accuracy (primary metric), macro-averaged F1 score (to account for class imbalance), per-class precision and recall, and confusion matrices to identify systematic error patterns. The best model for each approach was selected based on highest validation accuracy, and final performance was reported on the held-out test set.

III. RESULTS

A. Overall Model Performance Comparison

Table I presents the validation accuracy and macro F1 scores for all eight evaluated models. Transfer learning approaches dramatically outperformed all other methods, with ResNet50 achieving the highest accuracy of 91.82% and EfficientNet-B0 reaching 90.13%. These results represent a 27.47% and 23.65% improvement over their from-scratch counterparts, respectively. Among custom CNN architectures, the Medium CNN achieved the best

performance at 50.11% accuracy, while the Baseline CNN reached 44.62%. Notably, the VGG-style CNN exhibited severe model collapse, achieving only 14.13% accuracy by predicting almost exclusively the most frequent class (Jelly Fish). Autoencoder-based approaches underperformed supervised learning baselines, with the fine-tuned encoder reaching 42.38% and the frozen encoder achieving 39.46% accuracy.

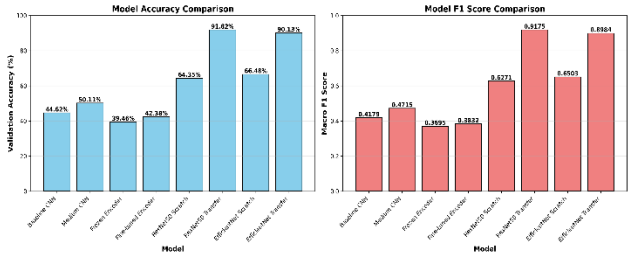


Fig. 1 – Performance comparison across all evaluated models

Table 1: Model Performance Comparison

Model	Val Accuracy	Macro F1	Parameters
ResNet50 (Transfer)	91.82%	0.9175	25.6M
EfficientNet-B0 (Transfer)	90.13%	0.8984	5.3M
EfficientNet-B0 (Scratch)	66.48%	0.6503	5.3M
ResNet50 (Scratch)	64.35%	0.6271	25.6M
Medium CNN	50.11%	0.4715	5.2M
Baseline CNN	44.62%	0.4179	104.8M
Fine-tuned Encoder	42.38%	0.3832	4.0M
Frozen Encoder	39.46%	0.3695	4.0M
VGG-style CNN	14.13%	0.2475	15.3M

B. *Training Dynamics and Convergence*

Figure 2 and 3 illustrates the training and validation accuracy curves for representative models from each experimental approach. Transfer learning models exhibited rapid convergence, with EfficientNet-B0 reaching 89% validation accuracy by epoch 20 and maintaining stable performance thereafter. Training accuracy reached 97-98% while validation accuracy plateaued at 90-91%, indicating a modest 7-8% generalization gap.

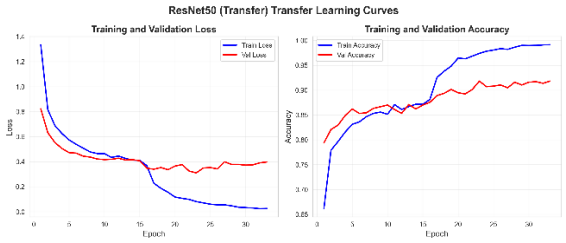


Fig. 2 – ResNet50 Transfer Learning

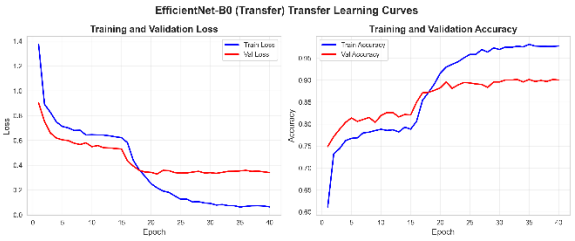


Fig. 3 – EfficientNet-B0 Transfer Learning

Custom CNNs showed slower convergence with the Medium CNN requiring approximately 30 epochs to reach its peak performance of 50%. The training curves revealed higher variance and the models struggled to exceed 35% training accuracy in early epochs, suggesting difficulty learning discriminative features from scratch with limited data.

The autoencoder approach displayed interesting dynamics during the reconstruction phase, with the training loss decreasing steadily to 1.32 over 25 epochs. However, when the frozen encoder was used for classification, validation accuracy plateaued at 39% despite the classifier being trained for 25 epochs. Fine-tuning the encoder improved performance to 42% but remained well below supervised baselines.

The VGG-style CNN exhibited catastrophic model collapse, with validation loss failing to decrease below 2.30 and the model converging to predict predominantly a single class. Analysis of the confusion matrix revealed that 100% of predictions for nine out of ten classes were misclassified as Jelly Fish, the most represented class in the training set.

C. *Per-Class Performance Analysis*

Table II shows the per-class F1 scores for the three best-performing models. Performance varied significantly across classes, with certain species consistently easier to classify than others.

Table II: Per-Class F1 Scores For Top Models

Class	Medium CNN	ResNet50 (Transfer)	EfficientNet-B0 (Transfer)
Sea Urchins	0.789	0.988	0.977
Jelly Fish	0.635	0.968	0.952
Dolphin	0.596	0.891	0.906
Nudibranchs	0.513	0.933	0.907
Puffers	0.462	0.937	0.949
Corals	0.434	0.920	0.893
Whale	0.439	0.812	0.893
Octopus	0.358	0.845	0.824
Sharks	0.350	0.807	0.803
Sea Rays	0.139	0.857	0.779

Sea Urchins were the easiest class to identify across all models, achieving F1 scores above 0.78 even with the Medium CNN and approaching 0.99 with transfer learning. Jelly Fish also performed well, likely due to their distinctive transparent morphology. Conversely, Sea Rays were the most challenging class for custom CNNs (F1=0.139), though transfer learning improved this to 0.78-0.86. Sharks

and Octopus also presented classification difficulties, particularly for smaller models.

D. Confusion Pattern Analysis

Confusion matrices for the best models revealed systematic error patterns related to biological taxonomy. For ResNet50 (transfer learning), the primary confusion occurred between Dolphins and Whales, with 7-8 images misclassified between these cetacean species. Similarly, Sea Rays and Sharks showed mutual confusion, with 4-5 misclassifications between these cartilaginous fish. These errors are biologically interpretable, as both pairs of species share similar body morphology and evolutionary lineage.

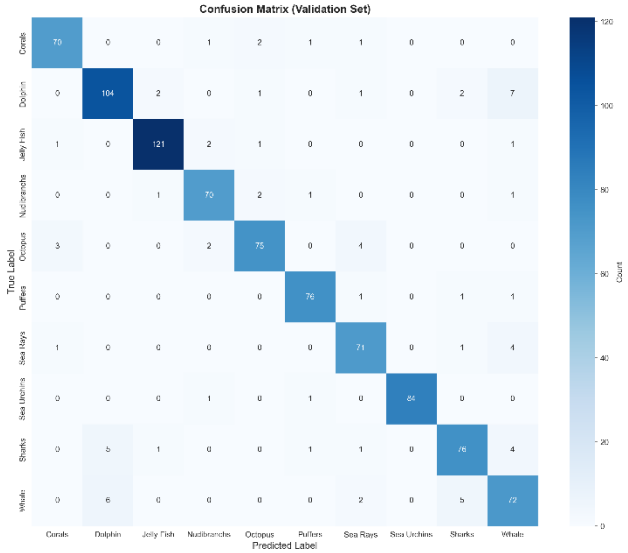


Fig. 4 – Confusion matrix for ResNet50 Transfer Learning

The Medium CNN showed more severe confusion patterns, with substantial misclassification across multiple class pairs. Dolphins were frequently confused with Whales (23 instances) and Sharks (6 instances). Corals showed high confusion with Nudibranchs (14 instances) and Octopus (13 instances), likely due to textural similarities and complex backgrounds. Sea Rays demonstrated the poorest discrimination, being confused with Sharks (25 instances), Sea Urchins (15 instances), and Puffers (12 instances).

For the collapsed VGG-style model, the confusion matrix showed a degenerate pattern where approximately 85-100% of all classes were predicted as Jelly Fish. This classic model collapse phenomenon occurs when the network converges to predicting the most frequent class as a local minimum, failing to learn discriminative features for other classes.

IV. DISCUSSION

A. Transfer Learning Superiority for Limited Data

The substantial performance advantage of transfer learning (27-41% improvement over from-scratch training) validates established principles in deep learning but provides important quantification for marine species classification specifically. ResNet50 and EfficientNet-B0, pre-trained on ImageNet's 1.2 million images across 1,000 categories, learned generalizable low-level and mid-level visual features such as edge detection, texture patterns, and shape recognition [6]. These features transferred effectively to marine imagery despite the domain gap between terrestrial ImageNet images and underwater photography.

The two-phase training strategy proved essential for optimal performance. During Phase 1 (frozen backbone), the classification head learned to map pre-trained features to marine species categories without disrupting learned representations. Phase 2 (unfrozen backbone) allowed fine-tuning of feature extractors to capture marine-specific characteristics such as bioluminescence patterns, fin shapes, and underwater color distributions. This approach achieved 89-90% accuracy within 20 epochs, demonstrating remarkable sample efficiency compared to the 40+ epochs required for custom CNNs to reach half this performance.

The efficiency of EfficientNet-B0 deserves particular attention. With only 5.3M parameters (21% of ResNet50's 25.6M), it achieved 90.13% accuracy, representing excellent computational efficiency. This suggests that compound scaling methods, which systematically balance network depth, width, and resolution, provide advantages for marine classification tasks where deployment on resource-constrained platforms (underwater drones, edge devices) may be necessary.

B. Custom CNN Limitations with Moderate Dataset Size

The plateau of custom CNNs at approximately 50% accuracy reflects fundamental limitations of training deep networks from scratch with moderate-sized datasets. With only 4,181 training images across 10 classes (average 418 images per class), custom architectures struggled to learn the hierarchical feature representations that deep learning requires. The training curves revealed this difficulty, with models taking 20-30 epochs to reach 30% training accuracy and exhibiting high variance in validation performance.

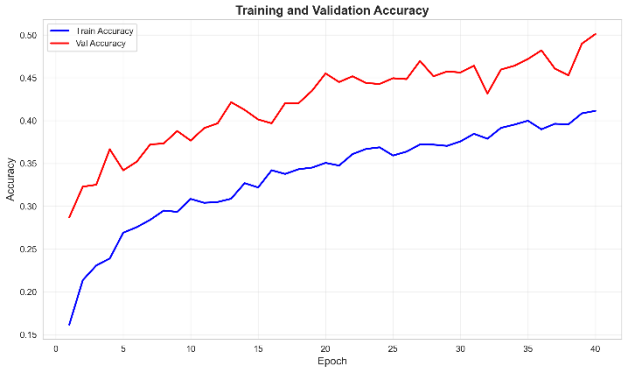


Fig. 5 – Medium CNN training curves

The Medium CNN (5.2M parameters) outperformed the Baseline CNN (104.8M parameters) despite having 20 times fewer parameters. This counter-intuitive result demonstrates that model capacity must be matched to dataset size. The Baseline CNN's excessive parameters led to overfitting, where the model memorized training examples rather than learning generalizable features. In contrast, the Medium CNN's moderate capacity, combined with regularization techniques (batch normalization, dropout, global average pooling), achieved better generalization.

Interestingly, the performance gap between custom CNNs and transfer learning widens substantially for difficult classes. For Sea Rays, the Medium CNN achieved only 0.139 F1 score while transfer learning reached 0.78-0.86. This suggests that fine-grained discrimination of marine species requires feature representations that are difficult to learn from limited data alone, further

emphasizing the value of pre-trained models for this domain.

C. *Autoencoder Pre-training Failure Analysis*

The underperformance of autoencoder-based pre-training (8-11% below supervised baselines) reveals a fundamental misalignment between reconstruction and classification objectives. During the unsupervised pre-training phase, the autoencoder learned to compress and reconstruct images by minimizing pixel-wise mean squared error. This objective encourages the model to preserve overall image appearance, including background details, lighting conditions, and color distributions that are irrelevant or potentially harmful for species classification.

Analysis of the reconstruction outputs showed that the autoencoder successfully captured general image structure and dominant colors (the blue-green hues of underwater scenes) but produced blurry outputs that lost fine-grained details. These details—such as the distinctive spines of sea urchins, the fin patterns of dolphins, or the body spots of nudibranchs—are precisely what enable species discrimination. By optimizing for reconstruction fidelity rather than class separability, the autoencoder allocated representational capacity to features that do not aid classification.

The frozen encoder approach (39.46% accuracy) performed particularly poorly because the learned features were fundamentally misaligned with the classification task. Even with 25 epochs of classifier training, the model could not overcome the limitations of the fixed feature representation. Fine-tuning the encoder (42.38% accuracy) provided modest improvement by allowing some adaptation of features, but starting from a suboptimal initialization meant the model could not match the performance of supervised learning from scratch (50.11%).

These results align with recent findings in representation learning showing that self-supervised methods perform best when the pretext task (e.g., image reconstruction) shares structural similarity with the downstream task. For classification, contrastive learning methods that explicitly learn discriminative features would likely perform better than reconstruction-based approaches.

D. *Model Collapse Phenomenon in VGG-Style Architecture*

The catastrophic failure of the VGG-style CNN (14.13% accuracy) exemplifies model collapse, a well-documented failure mode in deep learning where networks converge to degenerate solutions . The confusion matrix revealed that the model predicted Jelly Fish for 85-100% of all inputs across nine of ten classes, effectively learning to ignore input images and output a constant prediction.

This collapse occurred despite the VGG-style architecture's success in other domains, highlighting the interaction between model depth, dataset size, and class imbalance. With 15.3M parameters and five convolutional blocks, the VGG-style CNN had the capacity to memorize the training set entirely. However, with only 4,181 training examples, the optimization landscape contained many poor local minima. The model discovered that predicting the most frequent class (Jelly Fish, 14.14% of training data) minimized loss during early training epochs.

Once this pattern emerged, gradient descent reinforced it through a feedback loop: as the model increasingly predicted Jelly Fish, the gradient signals for other classes weakened, making it progressively harder to escape this local minimum. The validation loss plateau at 2.30 (compared to 1.42-1.70 for successful models) indicated that the model never learned meaningful features. This failure demonstrates the importance of careful architecture selection and regularization when working with limited data, as well as the need to monitor for collapse during training through per-class metrics rather than overall accuracy alone.

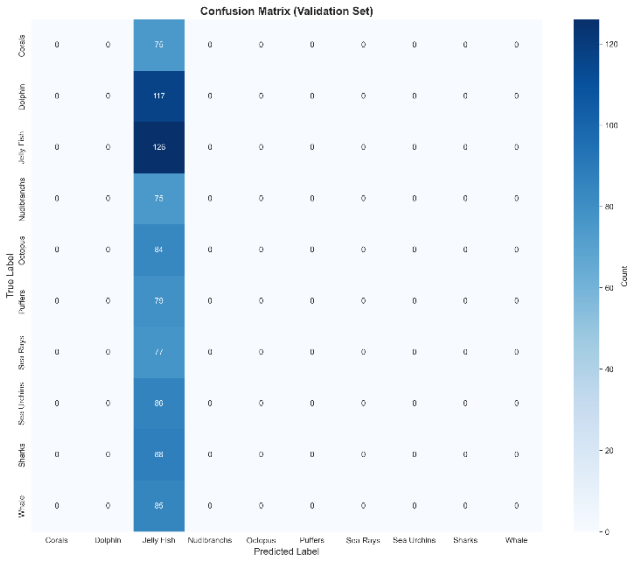


Fig. 6 – VGG-style CNN confusion matrix

E. *Biological and Technical Challenges in Marine Classification*

Even at 91.82% accuracy, systematic confusion patterns persisted that reflect genuine biological similarities rather than technical limitations. The primary confusion between dolphins and whales (7-8 misclassifications in ResNet50) is taxonomically meaningful—both are cetacean marine mammals with streamlined fusiform bodies adapted for aquatic locomotion. Distinguishing them requires identifying subtle differences in dorsal fin shape, body proportions, and size cues that may be ambiguous in underwater photography.

Similarly, the confusion between sea rays and sharks (4-5 misclassifications) reflects their shared evolutionary history as cartilaginous fish (Chondrichthyes). Both possess similar coloration, lack of scales, and when photographed from certain angles, comparable body profiles. These biological constraints suggest that achieving accuracy above 95% may require additional information beyond single images, such as behavioral context, size reference objects, or multi-view observations.

The classes with highest accuracy—sea urchins (98.8% F1) and jelly fish (96.8% F1)—possess distinctive morphological features that are difficult to confuse. Sea urchins' spherical bodies covered in radiating spines and jelly fish's transparent bell-shaped bodies with trailing tentacles are taxonomically unique. This suggests that marine classification accuracy is ultimately bounded by the degree of morphological distinctiveness between species classes.

Technical challenges specific to underwater imagery also contributed to errors. Variable lighting conditions create significant color shifts, with deeper water appearing increasingly blue-green as red wavelengths are absorbed. Background clutter from coral reefs, rocky substrates, and other marine organisms complicates segmentation and can introduce confounding features. Some images captured subjects at oblique angles or partially obscured, reducing the availability of diagnostic features. These challenges highlight the continued need for domain-specific data augmentation strategies and potentially multi-modal approaches incorporating depth information or video sequences.

V. CONCLUSION

This study compared three deep learning paradigms for marine species classification using a curated 10-class dataset of 5,978 underwater images. Transfer learning with ImageNet-pretrained models achieved the best results, with ResNet50 reaching 91.82% accuracy and EfficientNet-B0 achieving 90.13%—representing a 27-41% improvement over training from scratch. Custom CNNs plateaued at 50% accuracy due to limited training data (~4,200 images), while autoencoder-based pre-training underperformed supervised baselines by 8-11%, demonstrating that reconstruction objectives do not align with classification tasks.

The VGG-style architecture exhibited complete model collapse (14.13% accuracy), highlighting the importance of matching model capacity to dataset size. Confusion analysis revealed biologically meaningful errors between taxonomically similar species—dolphins and whales (cetaceans) and sea rays and sharks (cartilaginous fish)—suggesting that single-image classification accuracy above 95% may require additional contextual information.

Our results provide practical guidance for marine classification systems: transfer learning should be the default approach for datasets under 10,000 images, while autoencoder pre-training is not recommended when labeled data is available. Future work should explore modern self-supervised learning methods, domain-specific data augmentation for underwater imagery, and multi-modal approaches incorporating temporal or depth information to address remaining biological ambiguities.

REFERENCES

[1] M. J. Costello and B. Chaudhary, "Marine biodiversity, biogeography, deep-sea gradients, and conservation," *Current Biology*, vol. 27, no. 11, pp. R511-R527, 2017.

[2] A. Allken, N. O. Handegard, S. Rosen, T. Schreyeck, T. Mahiout, and K. Malde, "Fish species identification using a convolutional neural network trained on synthetic data," *ICES Journal of Marine Science*, vol. 76, no. 1, pp. 342-349, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248-255.

[5] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," in *Proc. British Machine Vision Conference (BMVC)*, 2014.

[6] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 3320-3328.

[7] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105-6114.

Marine Species Classification Using Deep Learning

Systematic Comparison of Custom CNNs, Unsupervised Pre-training, and Transfer Learning Approaches

COMP263 Group 5

DATE: 12-Dec-2025

Problem Statement & Motivation

Problem

- ❑ Slow manual identification
- ❑ Requires expert knowledge
- ❑ Prone to human error
- ❑ Needs scalable automation

Our Approach

- ❑ Deep-learning image classification
- ❑ Kaggle marine species dataset
- ❑ 10 target classes (23 classes)
- ❑ Compare 3 model types

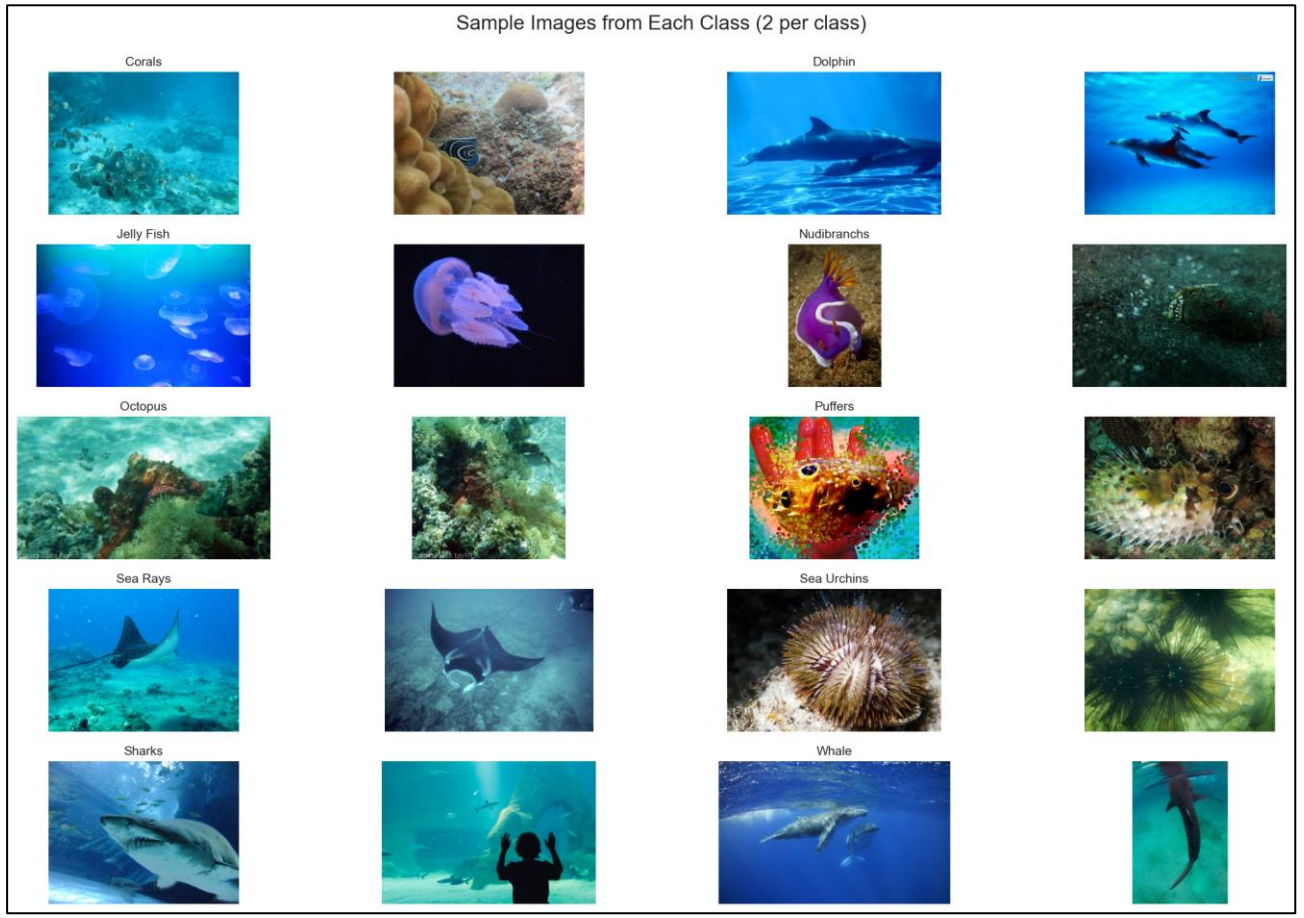
Dataset Overview & Statistics

Dataset Summary

- 5,978 RGB images
- 10 marine animal classes
- Image sizes vary (~170–300 px)
- No corrupted images detected

Key Challenges

- High intra-class variability
- Inter-class similarity
- Underwater noise

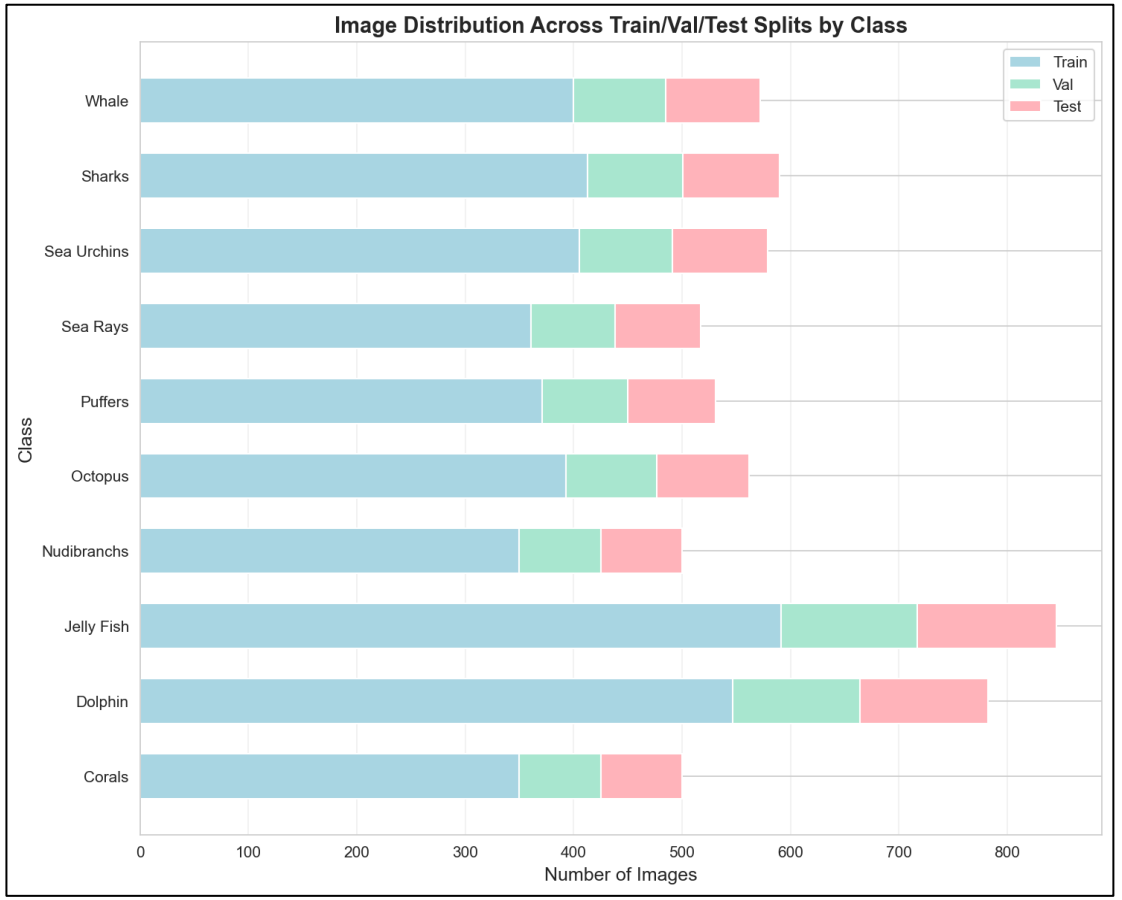


Dataset Overview & Statistics

Key Insights

- ❖ Moderate imbalance (~1.7×) across classes
- ❖ Stratified 80/15/15 train/val/test split
- ❖ Standard preprocessing:

  Resize → CenterCrop → Normalize (ImageNet stats)
- ❖ Training only: light augmentation



Training Infrastructure & Setup

Hardware

- GPU: NVIDIA RTX 3070 Laptop (8GB VRAM)
- CUDA 12.1 with cuDNN
- Mixed precision training (torch.cuda.amp)

Training Configuration

- Batch Size: 32
- Loss: CrossEntropyLoss (class weights)
- Optimizer: Adam
- Scheduler: ReduceLROnPlateau (patience=3, factor=0.5)
- Early Stopping: patience=10

Software

- Framework: PyTorch 2.5.1 with CUDA support
- Python 3.12
- Key Libraries: torchvision, scikit-learn, matplotlib

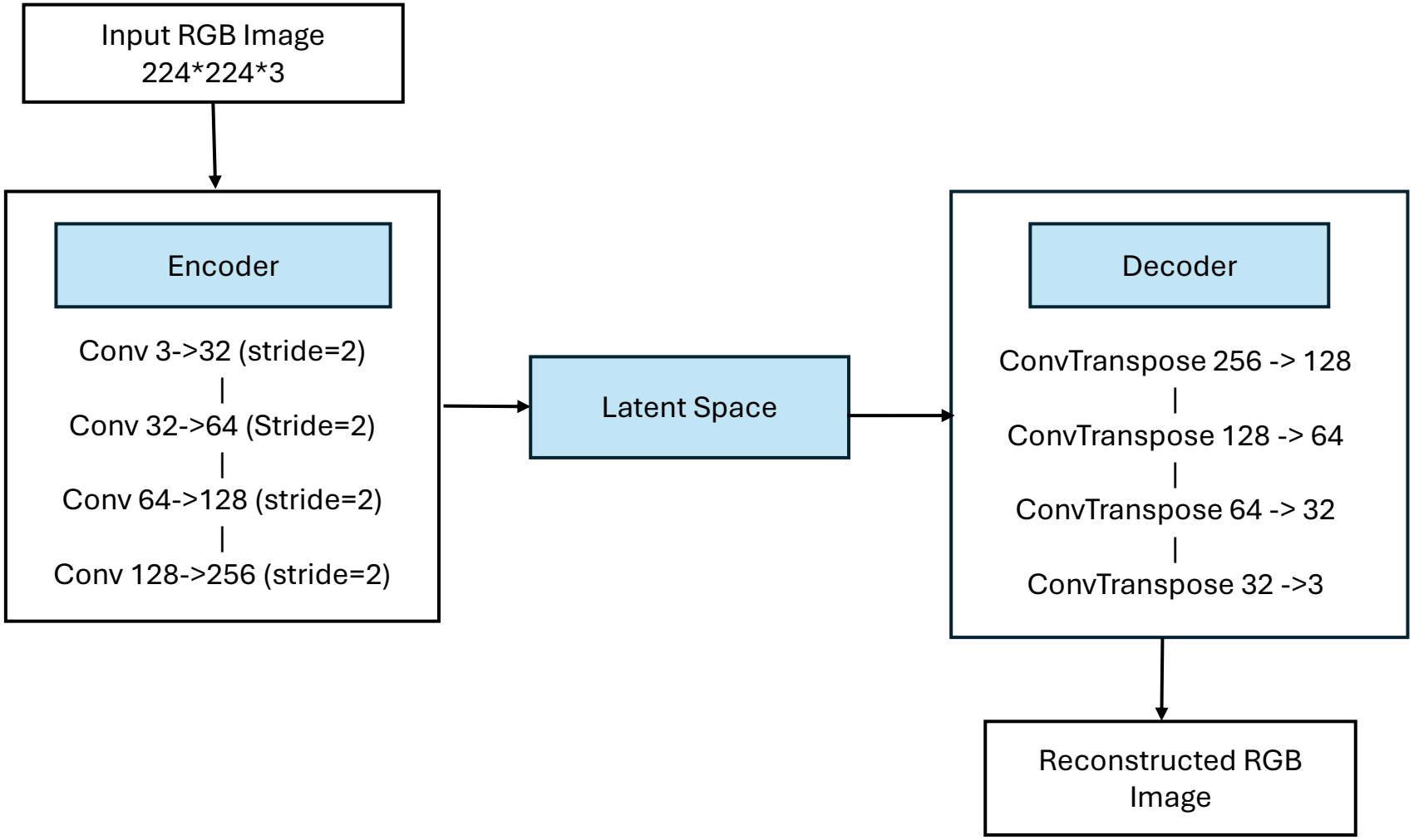
Experiment 1 - Custom CNN Architectures

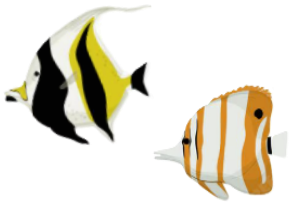
Model	Parameters	Architecture Summary	Outcome / Issue
Baseline CNN	104M	4 blocks: Conv → BN → ReLU → MaxPool; Dropout(0.25–0.3); Classifier 1024→512→10	Heavy, overfits, slow to train
Medium CNN	7M	3 blocks with 2×Conv each; 32→64→128 channels; Dropout(0.2–0.4); Classifier 256→10	Best balance, stable training
VGG-Style CNN	18M	4 VGG-like blocks: 64→128→256→256; deeper than Model 2	Collapsed → predicts only 1 class (Jellyfish)

Experiment 1 - Results Analysis

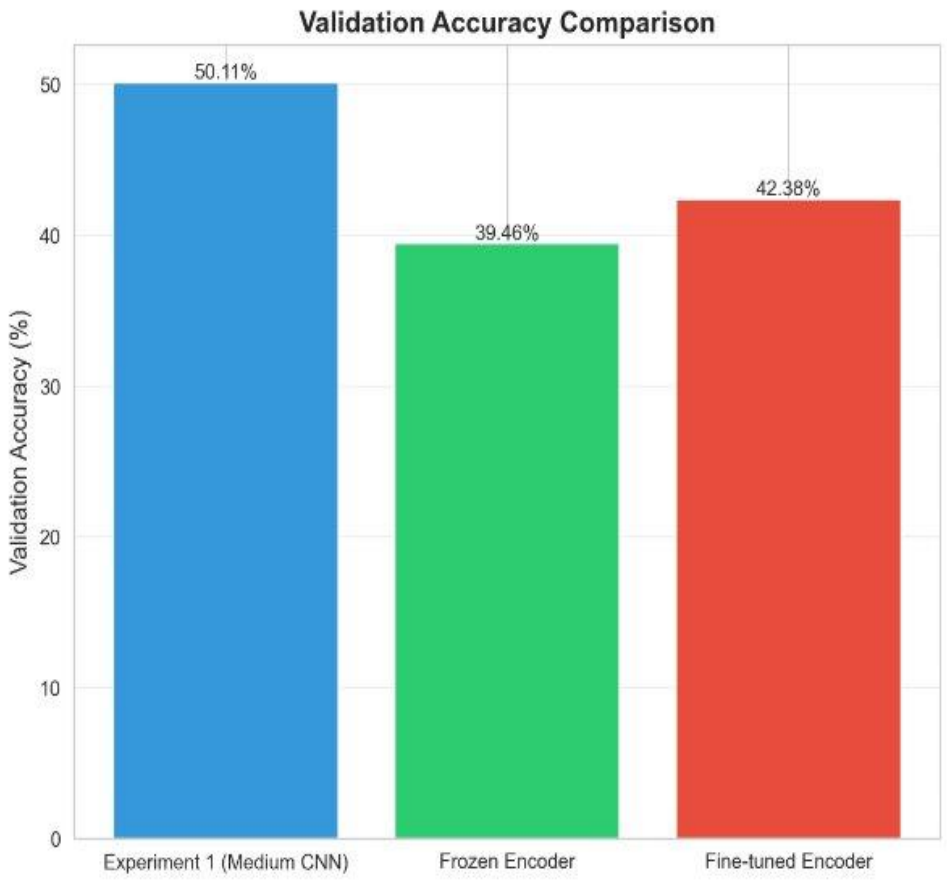
Model	Parameters	Accuracy	F1 Score	Training Time
Baseline	104M	44.62%	0.4179	34 min
Medium	7M	50.11%	0.4715	27 min
VGG-Style	18M	14.13%	0.2475	19 min (failed)

Experiment 2 - Unsupervised Pre-training





Experiment 2 – Result Analysis



Experiment 3 - Transfer Learning Setup

Strategy A
Train From Scratch

- Random initialization
- Optimizer: Adam (lr = 0.001, weight_decay = 1e-4)
- Train full model (40 epochs)
- Early stopping (patience = 10)

Strategy B Transfer Learning (Phase-1)

- Load ImageNet pretrained backbone
- Freeze all convolutional layers
- Train only classifier head
- Optimizer: Adam (lr = 0.001)

Strategy B Transfer Learning (Phase –2)

- Unfreeze entire model
- Fine-tune all layers
- Optimizer: Adam (lr = 0.0001)

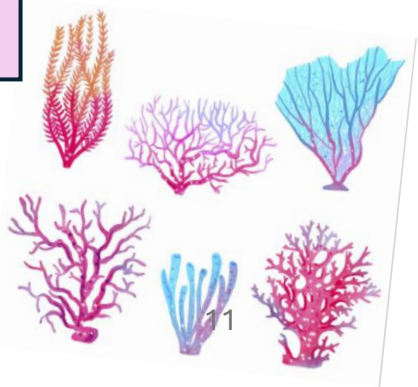
Experiment 3 - Results

Model	From Scratch	Transfer Learning	Improvement
ResNet50	64.35%	91.82%	+27.47%
EfficientNet-B0	66.48%	90.13%	+23.65%

Best: Sea Urchins (95%), Jellyfish (93%)

Hard: Sea Rays (85%)

Issue: Visual similarity to Sharks



Comprehensive Comparison & Analysis

Rank	Model	Accuracy	Macro F1	Params	Key Insight
1	ResNet50 Transfer	91.82%	0.9175	25.6M	Transfer learning winner
2	EfficientNet Transfer	90.13%	0.8984	5.3M	Efficient alternative
3	EfficientNet Scratch	66.48%	0.6503	5.3M	SOTA helps even w/o transfer
4	ResNet50 Scratch	64.35%	0.6271	25.6M	Deep architecture needs data
5	Medium CNN	50.11%	0.4715	7M	Best custom architecture
6	Baseline CNN	44.62%	0.4179	104M	Overparameterized
7	Fine-tuned Encoder	42.38%	0.3832	26M	Unsupervised < supervised
8	Frozen Encoder	39.46%	0.3695	26M	Reconstruction ≠ classification

Technical Challenges & Solutions

Class Imbalance

- **Problem**: Dataset Bias
- **Solution**: Weight CrossEntropyLoss
- **Result**: Balanced class performance

Small Validation Set

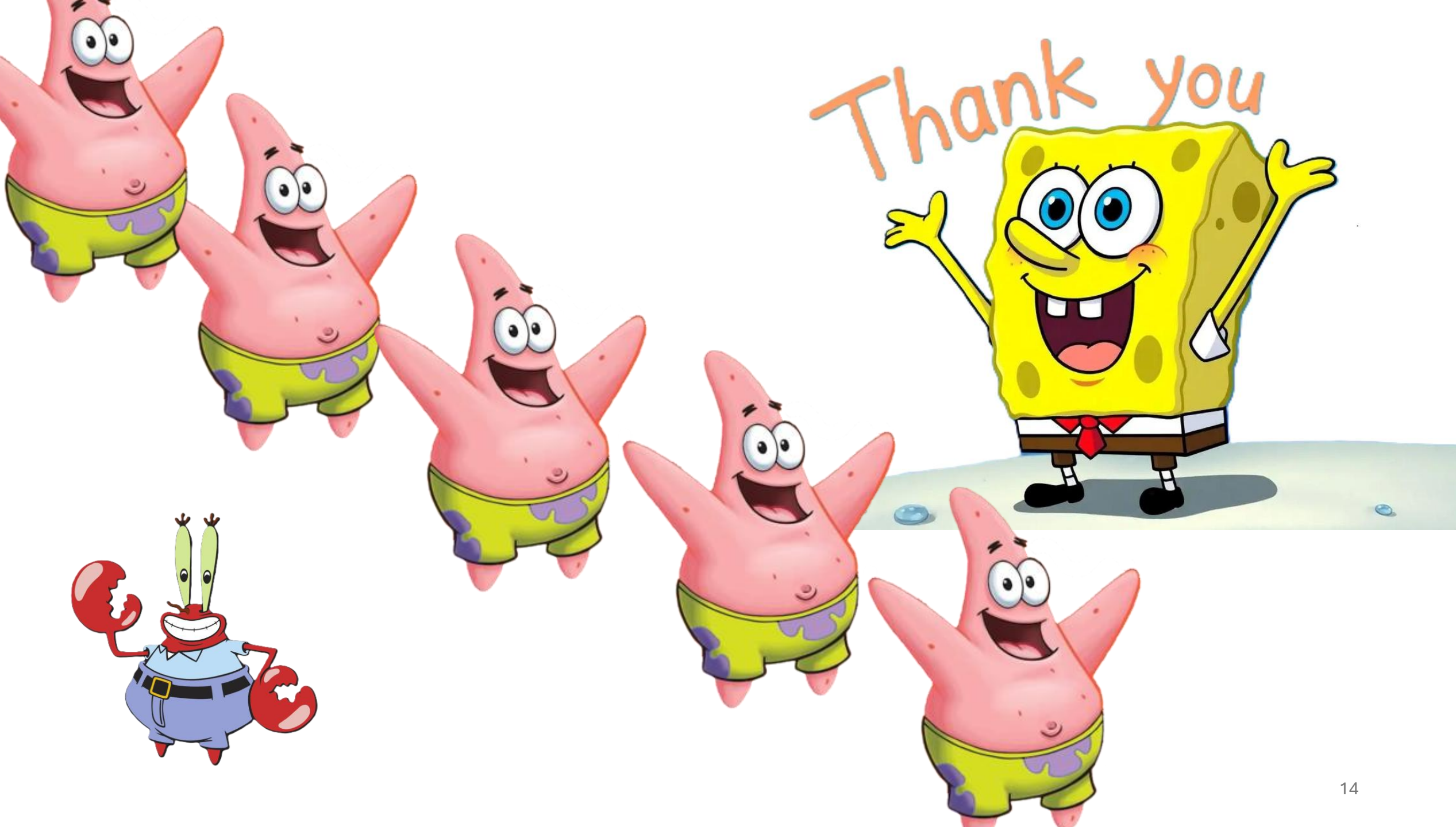
- **Problem**: Unstable Metrics
- **Solution**: Restructured to 70/15/15 split
- **Result**: Reliable validation

VGG Training Collapse

- **Problem**: Low Accuracy
- **Solution**: SOTA architectures (ResNet, EfficientNet)
- **Result**: 90%+ accuracy

Overfitting

- **Issue:** Training vs. Validation Gap
- **Fix:** Augmentation & Regularization
- **Outcome:** Generalized Performance (92% Val Acc)


Thank you