TECHNISCHE UNIVERSITÄT ILMENAU
Department of Computer Science and Automation
Institute of Applied Computer Science
Data-intensive Systems and Visualization Group

Research Project

Exploring visual similarities and genetic similarities with machine learning

Submitted By

Niloy Roy
Matrikel Nr. 65522
Winter Semester 2024/25
Research in Computer and Systems Engineering

Supervised By:

Martin Hofmann, M.Sc.

Ilmenau, May 2, 2025

Contents

1 Introduction

Taxonomic classification is the hierarchical organization of living organisms into groups such as order, family, genus, and species, based on their morphological, genetic, and ecological similarities and differences. However, relying solely on a single data modality, such as visual or genetic information, has often proven insufficient for accurate and confident classification, particularly within complex taxonomic groups(Karbstein et al., 2024). Evolutionary convergence and morphological disparity further complicate this task. In addition, ancient fossil specimens, which could offer valuable insights into evolutionary and ecological history, often lack high-quality genetic sequence data, making them even harder to classify.

Given these challenges, deep similarity learning presents a powerful tool for advancing taxonomic studies. Similarity learning is focused on learning how to measure the similarity between data points. Traditional methods, such as cosine similarity or Manhattan distance, have laid the foundation for this field. However, without parameter training, these methods often suffer from the curse of dimensionality, particularly when dealing with complex, high-dimensional datasets. With the rapid growth in large-scale digital image data availability, machine learning models—especially those employing deep learning—have become essential for effectively handling and interpreting such data. These methods allow for the learning of biologically meaningful latent representations from visual data, where the distances between vectors in the latent space reflect semantic and evolutionary relationships. This can open new doors for analyzing morphological diversity and understanding its relationship to genetic variation.

In this study, we aim to explore how advanced similarity learning techniques can be used to model and compare visual similarities and genetic distances among mollusk families. Specifically, we investigate whether deep learning models trained on large datasets of mollusk images can accurately predict taxonomic affinities and improve classification performance. By analyzing the correlation between visual phenotypes and genetic structures, this research seeks to contribute to evolutionary biology, taxonomy, and the development of automated species identification tools. To achieve these goals, we will formulate specific research questions, design and implement targeted experiments, and ultimately present and discuss the insights derived from our findings.

2 Research Question

For the bivalve from the Mollusca phylum relevant in our study, empty shell (Bouchet et al., 2002, 2023) samples are highly represented, and fossil records (Pojeta, 2000) are prevalent too. Because the organisms inside are dead, empty shells don't have any soft tissue, thus making high-quality genetic sequence data unavailable. A similar challenge is true for the fossils: no genetic data is available. This challenge motivates our first research question to examine how well we can identify taxons with our image dataset alone in the absence of genetic data, exemplarily at the subclass and order levels. The first research question to investigate is:

- **With how much confidence can visual similarity serve as a proxy for genetic similarity in taxonomic classification?**

Another major challenge in evolutionary and ecological studies is taxonomically complex groups. We can mention convergent evolution in this context, where unrelated or different species can develop almost indistinguishable morphology merely under the pressure of similar environmental habitats. For example, we can mention the blue mussels of the Mytilus edulis species complex. These three mussels *(M. edulis, M. galloprovincialis, and M. trossulus)* are morphologically so similar that they were long considered a single taxon until molecular evidence proved otherwise (Katolikova et al., 2016; García-Souto et al., 2017). From this, we can hypothesize that to learn a biologically meaningful visual embedding, phylogenetic distance data can provide valuable guidance to improve on the limitations of morphology alone. Therefore, we ask our second research question:

- **To what extent can a biologically meaningful embedding space be learned from visual morphology data, when guided by phylogenetic distance?**

As a part of answering these key research questions, we will also discuss:

- What are the most suitable models or algorithms for feature extraction and similarity learning with large-scale image datasets?
- What metrics best describe the correlation between morphological and genetic similarity learning? What is the most effective way to visualize these relationships?

3 Related Work

The works related to our research questions (focusing on taxonomic classification and embedding learning) were searched following the Systematic Literature Review (SLR) standard. Developed search strings were tried across key databases like IEEE Xplore, Google Scholar, SpringerLink, PubMed, ScienceDirect, and arXiv.The insights from the findings are summarized below, categorized into supervised, self-supervised and unsupervised paradigms.

Supervised learning has been extensively used for taxonomic classification and metric learning. Duan et al. (2023) introduced a deep metric learning method designed to improve fine-grained image retrieval by combining several key ideas. First, the model is similarity-aware at multiple levels, meaning it doesn't treat all different classes as equally dissimilar—instead, it considers how nearby or far they are. Second, it uses deep local descriptors, which extract features from specific parts of an image (like a bird's wings or head), helping the model focus on fine details that matter in similar-looking categories. Finally, the method dynamically adjusts the margin in its loss function based on how similar two samples are—smaller margins for more similar items and larger ones for more different items—making the learned features more precise and better suited to subtle visual differences.

Other supervised metric learning methods include Contrastive loss (Hadsell et al., 2006), which treats all negative samples equally, meaning it doesn't focus on hard negatives—those that are close to the anchor in the embedding space and thus more confusing. Triplet loss (Schroff et al., 2015) builds on this by considering three samples at a time—an anchor, a positive (same class), and a negative (different class)—and enforces that the anchor is closer to the positive than the negative by a margin. This formulation allows the model to focus on relative distances, but training can be inefficient without careful triplet mining strategies. Hierarchical triplet loss (Ge, 2018) further refines this idea by incorporating class hierarchy or similarity levels, assigning adaptive margins based on how similar the negative class is to the anchor's class. This leads to more nuanced supervision and better generalization in fine-grained settings. Ranked list loss (Wang et al., 2019) takes a more global approach by considering the entire set of positives and negatives for each anchor within a batch. It ensures all positives are pulled close while all negatives are pushed beyond a margin, weighted by difficulty. Together, these methods reflect a progression from local pairwise constraints to more structured and context-aware supervision in metric learning.

Hofmann et al. (2024) demonstrated that incorporating genetic distances can significantly improve taxonomic inference. However, their sequential classifier architecture struggled to effectively learn the intended taxonomic hierarchy. To address this, we can discuss the work from (Elhamod and Tung, 2020), which proposed the Hierarchy-Guided Neural Network (HGNN), a model designed to incorporate biological hierarchy between genus and species. HGNN uses a dual-ResNet architecture to learn feature representations for both levels simultaneously. These features are then fused to produce final predictions, and the model is trained with a joint loss

over genus and species labels. This approach improves classification accuracy, particularly on small and imbalanced datasets.

Few-shot learning (FSL) enables models to learn from a limited number of training examples, often as few as five data points per class (Chen et al., 2019). Within this domain, four key categories emerge: metric learning, data augmentation, meta-learning, and Bayesian approaches. **Metric learning** focuses on learning a distance function that allows for the comparison of the similarity between input samples, aiding in the classification of new instances based on proximity to known examples. Prominent methods in this category include prototypical, Siamese (Koch et al., 2015), and matching networks (Vinyals et al., 2016), which utilize embeddings to measure these distances effectively. The second category, **data augmentation** (Chen et al., 2019), employs generative techniques to create synthetic samples from existing data, thereby expanding the dataset and enabling models to learn more robustly. This often involves adversarial networks to produce varied yet relevant instances. In the realm of **meta-learning**, often referred to as "learning to learn" (Li et al., 2017), the goal is to develop algorithms that can quickly adapt to new tasks with minimal data by leveraging knowledge gained from previous tasks, with methods like model-agnostic meta-learning (Finn et al., 2017; Antoniou et al., 2018) being particularly noteworthy.

Self-supervised learning has emerged as a promising alternative by leveraging unlabeled data to generate meaningful representations.Within this paradigm, two main categories have gained traction: contrastive-based methods (e.g., SimCLR(Chen et al., 2020), MoCo(He et al., 2020), BYOL(Grill et al., 2020)) and clustering-guided methods (e.g., SwAV(Caron et al., 2020), DeepCluster(Caron et al., 2018)). Clustering-guided approaches such as SwAV addresses the limitation of contrastive learning's reliance on large batch sizes and negative pairs by using online clustering to learn from semantic consistency, making it more efficient and scalable. Contrastive methods focus on explicit instance separation, whereas clustering-based methods aim to uncover latent group structures. Overall, this paradigm enables models to learn structural similarities efficiently while reducing annotation costs.Hofmann et al. (2024) used SimCLR with the goal of exploring if visually extracted features could reflect genetic distances between bivalve families, without using any labeled data during training. After training, the extracted embeddings were compared to known genetic distances using a normalized square error metric. The results showed a moderately strong correlation (Pearson $r = 0.78$, $p < 0.00001$) between the learned visual similarity and genetic distances at the family level.

In the realm of **unsupervised learning**, several foundational methods have been explored to uncover meaningful structures from unlabeled data. Clustering techniques, such as k-Means and Spectral Clustering, are among the earliest and most widely used approaches, aiming to group data points based on similarity metrics without relying on ground truth labels. These methods have proven effective in applications like image grouping, anomaly detection, and bioinformatics. Dimensionality reduction techniques like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) are frequently used for visualization and preprocessing by transforming high-dimensional data into lower-dimensional representations while preserving key structural properties. On another front, generative models have gained significant attention for their ability to learn complex data distributions. Notably, Variational Autoencoders (VAEs) (Kingma and Welling, 2022) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have been instrumental in advancing tasks like unsupervised feature learning and synthetic data generation.

4 Methodology

4.1 Study of similarity learning paradigms

Table 4.1 provides a structured overview of similarity learning methods, categorizing traditional, machine learning-based, and advanced approaches with representative examples.

Category	Subcategory	Examples
Foundational Methods	Distance Metrics	Euclidean, Manhattan
	Kernel-Based Methods	RBF, Polynomial
	Information-Theoretic Measures	KL Divergence
	String-Based Methods	Edit Distance (Levenshtein)
Machine Learning	**Supervised Learning**	
	Learning to rank	RankNet, DeepRank, DSSM
	Contrastive Embedding Learning	Siamese Networks with triplet loss
	Transfer Learning	Fine-tuning pre-trained RESNets,vision transformers
	Few-shot Learning	Prototypical Networks, Meta-Learning (MAML)
	Self-Supervised Learning	
	Augmentation-based: contrastive	SimCLR, BYOL
	Augmentation-based: clustering-guided	SwAV
	Unsupervised Learning	
	Clustering	k-Means, Spectral Clustering
	Reconstructive	Autoencoders
	Generative	Generative Adversarial Networks
	Reinforcement Learning	
	Reward-Based Metric Learning	Trajectory-ranked Extrapolation)
	Reinforcement Learning for Embeddings	Deep Q-Networks
	Adaptive Similarity Learning with RL	Meta-Learned
	Similarity-Based Policy Optimization	Proximal Policy Optimization(PPO)
	Advanced Techniques, Emerging Trends	
	Zero-shot Learning	CLIP
	Graph-Based	Graph Neural Networks, Graph Attention Networks

Table 4.1: Similarity Learning Methods and Paradigms Overview

First, we divide the topic of similarity learning into three(3) main paradigms based on the supervision style—more specifically, according to the presence and nature of labels during the training phase. Now, we discuss various methods and provide supporting arguments to justify their categorization into each paradigm.

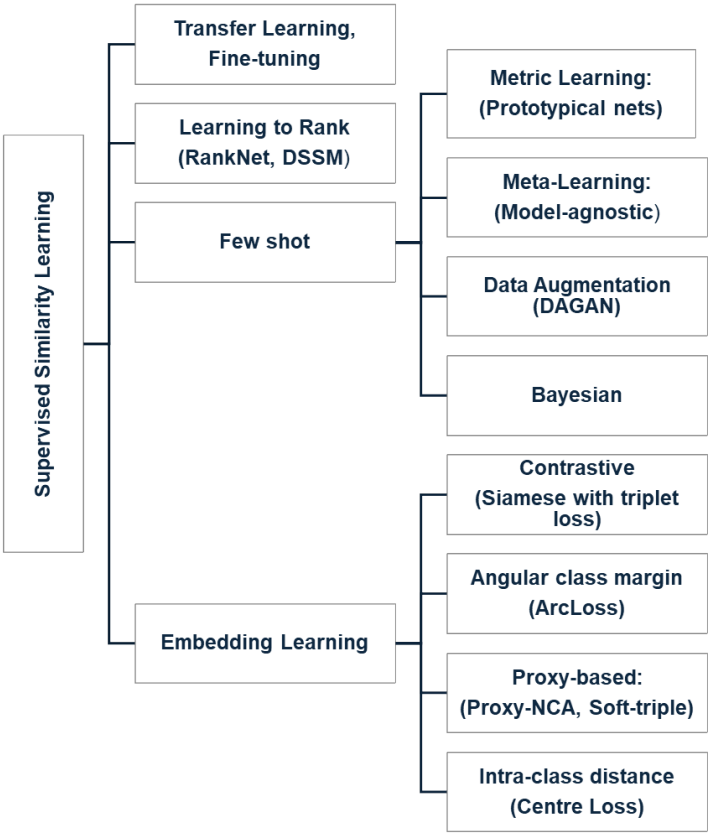

Figure 4.1: Supervised similarity learning

Contrastive embedding learning is one major kind under the supervised category. In contrastive learning, sample-wise methods like Siamese architecture with triplet loss, employ labeled samples arranged into anchor-positive-negative triplets. During training, these networks optimize embedding spaces by enforcing that an anchor is closer to a positive sample (same label) than to a negative sample (different label) by a predefined margin. This encourages positive samples to be closer than negative ones, which naturally leads to better rankings during inference. However, the goal is not to produce a sorted list, but to sculpt the geometry of the embedding space. The learning is considered **strongly supervised** if triplets are created using exact class labels. Conversely, if the triplets are constructed using auxiliary metadata or heuristic measures, where supervision exists only at the bag or group level, the framework aligns with **weakly supervised** learning.

Learning to Rank methods supervise the relative ranking between pairs of items based on labels that determine their comparative relevance or similarity. During training, these methods use labeled item pairs to optimize neural network parameters, ensuring that the most relevant items are at the top of a ranked list. When relevance labels are human-annotated, the approach is considered **strongly supervised**. In contrast, if labels are derived from click data or user behavior metrics, it falls under **weak supervision**.

In rank-based similarity learning, the architecture is often shared with other learning paradigms (especially metric learning or classification), but what makes it unique is how the model is trained (pairwise or listwise losses such as RankNet, ListNet, or LambdaRank) and how it's evaluated (rank-sensitive metrics like Recall@k, mean Average Precision (mAP) etc.). But some architectures are specifically designed or commonly used for ranking tasks rather than metric learning in the strict sense. A notable example is the Deep Structured Semantic Model (DSSM), which learns separate embeddings for queries and documents. Instead of minimizing a traditional embedding distance like in metric learning, DSSM is typically trained with ranking losses (e.g., softmax over cosine similarities) using click data. This makes it especially effective in search engines and recommender systems, where the goal is to rank relevant items higher rather than enforce absolute distances in embedding space.

One concept that underpins contrastive embedding learning, learning-to-rank, and other kinds of learning is **metric learning**, which refers broadly to the task of learning a **foundational** distance (e.g., Euclidean) or similarity function (e.g., cosine similarity) that reflects semantic relationships. This makes more sense as an enabling method than a distinct paradigm itself.

Similarly, **few-shot learning** approaches like prototypical networks and relation networks use labeled examples from limited support sets. In prototypical networks, class prototypes are computed by averaging feature embeddings of labeled support samples, while relation networks learn a distance metric from labeled pairs to classify unseen queries.

Transfer learning and fine-tuning clearly illustrates supervised learning, as both pre-training and fine-tuning phases typically rely on labeled data. For instance, models are commonly pre-trained on large labeled datasets, such as ImageNet-1K, which contains one thousand labeled object categories, using explicit class labels to optimize parameters initially. Later, these pretrained models, including their fully connected layers, can be fine-tuned or adapted explicitly on custom datasets, adjusting to the specific number and nature of classes defined by new supervised labels.

Self-supervised similarity learning is a training approach where the model learns useful representations from unlabeled data by creating its own supervision signals, setting it apart from supervised learning which requires labeled data, and from unsupervised learning which focuses more on discovering data structure without specific prediction tasks.The table 4.1 categorizes self-supervised learning methods based on how they use data augmentations to generate learning signals. SimCLR is placed under "augmentation-based: contrastive" because it learns by pulling together representations of different views of the same image and pushing apart those of other images using contrastive loss. In contrast, SwAV is grouped under "augmentation-based: clustering-guided" because it skips pairwise comparisons and instead learns by assigning cluster labels to different views of the same image and matching those labels.

The methods listed—clustering, dimensionality reduction, and generative modeling—are placed under the **unsupervised learning** paradigm because they learn from unlabeled data by uncovering hidden structures, patterns, or representations. Clustering algorithms like k-Means and Spectral Clustering group similar data points based on intrinsic distances without any ground truth labels. Dimensionality reduction techniques such as PCA and t-SNE also operate without supervision, focusing on simplifying high-dimensional data while preserving its underlying structure. Generative models, including Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), learn to model the data distribution itself, allowing them to generate new samples that resemble the input distribution. Despite differences in goals and

complexity, all these methods align with the unsupervised paradigm by learning meaningful patterns from raw data without relying on labeled examples.

4.2 Similarity Learning Methods: Strengths and Weaknesses

4.2.1 Foundational Methods

Foundational similarity methods rely on explicit mathematical formulas to calculate similarity or dissimilarity between two entities. They do not require data-driven learning and remain fixed once a metric is defined. For example, Cosine similarity specifically measures the cosine of the angle between two vectors. Its key strength, **scale invariance**, emerges directly from its dependence solely on the angle between vectors, thereby ignoring the magnitude of the vectors. Consequently, it excels in high-dimensional spaces where magnitudes can vary widely, such as in sparse data representations like genomic sequences or TF-IDF vectors for textual data. However, this same feature becomes a critical weakness when vector magnitude holds meaningful information, as Cosine similarity inherently neglects magnitude differences. Additionally, because Cosine similarity captures direction rather than absolute distance, it tends to be **sensitive to noise**, especially in high-dimensional spaces. Small variations or noise in data points can lead to significantly different angles, and therefore substantial changes in similarity measures, highlighting its vulnerability when precise distinctions between data points are required.

4.2.2 Supervised Similarity Learning

To analyze the strengths and weaknesses of the **supervised category**, we can start the discussion with the contrastive metric learning methods. Methods like Siamese networks with triplet loss learn by enforcing relative similarity constraints instead of class labels prediction: the model is trained to reduce the distance between anchor and positive samples (same class) while increasing the distance to negative samples (different class).This allows for **class-agnostic generalization** and makes the approach particularly effective in tasks like **face verification and one-shot learning**. However, constructing meaningful triplets and managing hard negative mining are **computationally intensive** and **require careful tuning** respectively. In contrast, **ArcFace** eliminates the need for sample pairing by learning from individual labeled instances using a modified softmax loss with an additive angular margin. It tightly clusters intra-class features and maximizes inter-class angular separability, leading to **highly discriminative embeddings** in classification tasks. Yet, this benefit depends heavily on clean and accurate labels; mislabeled samples introduce hard constraints that can distort the embedding space. Furthermore, ArcFace's **computational cost grows** with the number of classes, posing challenges for large-scale problems. In essence, strongly supervised metric learning excels at shaping feature space with labeled data but struggles when supervision is limited or noisy.

Transfer learning works by leveraging a model pre-trained on a large-scale dataset—such as ImageNet—and adapting it to a smaller, related task through fine-tuning. This workflow allows lower-level features (like edges or textures) learned from the source task to be reused, significantly **reducing the training time and the amount of labeled data needed for the target task**. For example, a ResNet model trained on ImageNet can be adapted to classify medical X-ray images by fine-tuning the later layers while retaining the general visual features captured earlier in the network. However, this advantage becomes a limitation when the source and target domains differ significantly; in such cases, the reused features may no longer

be relevant, leading to performance degradation—a problem known as **domain mismatch**. Additionally, since fine-tuning is typically performed on small datasets, there's a heightened risk of **overfitting**, especially when **too many layers(too many trainable parameters)** are updated without sufficient regularization. Thus, while transfer learning offers strong benefits in efficiency and performance, its effectiveness depends on the alignment between source and target domains and the care taken during fine-tuning.

Few-shot learning aims to generalize from a limited number of labeled examples based on prior knowledge and learning mechanisms suitable for data-scarce environments. Within this paradigm, **metric learning approaches** such as Prototypical Networks and Relation Networks rely on computing similarity between a query and a small set of labeled support samples. Prototypical Networks compute class prototypes by averaging embeddings of support samples, making them efficient and interpretable, but they assume a simple, unimodal distribution for each class, which limits performance in more complex settings. Relation Networks improve on this by learning a relation module to compare support-query pairs, but the learned metric can become sensitive to overfitting due to limited supervision. **Meta-learning approaches**, like Model-Agnostic Meta-Learning (MAML), train models to quickly adapt to new tasks using gradients from very few examples. This ability to rapidly fine-tune is powerful but computationally expensive and highly sensitive to task sampling during meta-training. **Data augmentation methods**, such as DAGAN, attempt to synthesize additional examples from few-shot data to enrich training; while this reduces data scarcity, it also introduces the challenge of generating high-quality, diverse augmentations without introducing noise or class ambiguity. **Bayesian approaches** incorporate uncertainty modeling into the learning process, which is advantageous when data is sparse, but they are often computationally demanding and hard to scale.Overall, few-shot learning is great at working with very little labeled data, but its effectiveness depends on how well the model uses prior knowledge and avoids overfitting to the small number of training examples.

4.2.3 Self-supervised Similarity Learning

Self-supervised learning operates by generating pseudo-labels through data augmentations, eliminating the need for manual annotation. A well-known example is SimCLR, which uses a contrastive loss to bring augmented views of the same image closer in the embedding space while pushing apart views of different images. This approach relies heavily on the presence of effective negative samples. Therefore, it requires very large batch sizes (typically 256–8192) to ensure a diverse and meaningful set of negatives, making it computationally demanding. In contrast, SwAV addresses this limitation by using a clustering-guided learning that avoids explicit pairwise comparisons and the need for negative samples. Instead, SwAV assigns cluster prototypes to image views and encourages consistency in assignments across different augmentations. This design reduces computational load and improves representation learning in smaller batch settings.

4.2.4 Unsupervised Similarity Learning

Unsupervised similarity learning can be approached through clustering, dimensionality reduction, and generative modeling—each with its strengths and weaknesses rooted in how they learn from data without labels. Clustering methods, such as k-Means and Spectral Clustering, group data points based on similarity measures, including Euclidean distance and graph-based

affinities. These methods are straightforward and interpretable, but their effectiveness depends heavily on the choice of distance metric and the number of clusters, which must often be specified in advance. For example, k-Means assumes spherical clusters and struggles with complex or overlapping data distributions, while Spectral Clustering can capture non-linear structures but is computationally expensive for large datasets. Dimensionality reduction techniques like PCA (Principal Component Analysis) and t-SNE reduce high-dimensional data to lower dimensions for visualization or further processing. PCA is efficient and preserves global structure, but it's limited to linear relationships. t-SNE excels at preserving local similarities and is widely used for visualizing cluster structures, but its non-parametric nature and sensitivity to hyperparameters (like perplexity) make it hard to generalize to new data. Lastly, generative models such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) can learn rich latent representations by reconstructing or generating data. VAEs (Kingma and Welling, 2022) provide a probabilistic embedding space useful for similarity tasks, but can produce blurry reconstructions. GANs (Goodfellow et al., 2014) generate high-quality samples and learn powerful feature representations, yet they are notoriously hard to train due to instability and mode collapse. Overall, while these unsupervised methods are useful for learning similarity without labels, their success largely depends on the underlying data structure, model assumptions, and training stability.

4.2.5 Key Differences Across Learning Paradigms

To summarize, the following table highlights the major differences across traditional, supervised, self-supervised, and unsupervised methods:

Table 4.2: Strengths and weaknesses of self-supervised and unsupervised similarity learning

Paradigm	**Strengths**	**Weaknesses**
Self-Supervised Learning	• Utilizes unlimited **unlabeled data** for training • Can learn very fine-grained features	• Learning might not align with target similarity • Quality evaluation only via downstream tasks (indirect)
Unsupervised Learning (e.g., clustering, auto-encoders)	• Discovers **natural groupings**/patterns without labels • Often fast and scalable algorithms (for clustering, PCA)	• No **guarantee of semantic relevance** of clusters • Can pick up on dataset biases • Lacks an objective measure of success without labels

Table 4.3: Strengths and weaknesses of supervised similarity learning

Paradigm	Strengths	Weaknesses
Supervised (Contrastive Embedding Learning)	• Embedding useful for retrieval and visualization	• Hard negative mining • Large batches
Supervised (Learning to Rank)	• Suited for information retrieval and recommender systems	• Sensitive to *label noise*
Few-shot Learning	• Learns to generalize from **minimal examples** • Tackles new classes without re-training	• Relies on representative meta-training distribution • Sensitive to domain shift in unseen classes
Transfer Learning, Fine-tuning (Knowledge Transfer)	• **Reduced training** effort • Fast convergence and often improved generalization • Cross-domain applications (using pre-trained models)	• Domain mismatch • Risk of **over-fitting**

5 DataSet

The provided dataset consists of labeled morphological images and phylogenetic distances. We have 71,888 shell images from the *Bivalvia* class of the *Mollusca* phylum. We have taxonomic labels (in order: **subclass, order, family, genus, species**) for each image in their filenames. The 2D images differ in their dimensions (height, width). The images collected from Hofmann et al. (2024) span 4,144 species across 884 genera, 74 families, 26 orders, and 6 subclasses. After being sourced from various repositories, the .jpg files were verified against the World Register of Marine Species (WoRMS) (WoRMS Editorial Board, 2024).

On top of that, the labels were available as a meta.csv file, which maps the **filenames** to the respective taxons. 71,888 images are labeled in the **5(five)** taxons mentioned in the previous paragraph.

Upon preliminary inspection, we can observe the smallest subclass is Paleoheterodonta with 67 images, and the largest is Imparidentia with 33,852 images. The same degree of class imbalance was found upon investigation for other taxonomic levels as well (e.g., order, family, genus and species). The lower the taxonomic level, the more the imbalance and more fine-grained features are important to classify them correctly.

For the distance data, the pairwise distances were extracted from a phylogenetic tree and presented as a symmetric and square matrix **(74*74)** for 74 families. Families under the same order will have less distance between them in comparison with families from different orders. The family-level phylogenetic distances were derived from a molecular phylogenetic tree based on 103 bivalve species. The original tree from Bieler et al. (2014) was reconstructed using TreeSnatcher Plus, and family-level distances were calculated as the sum of branch lengths along the shortest paths between taxa. These were later normalized (via softmax) and used during training to align model predictions with phylogenetic relationships by Hofmann et al. (2024).

6 Experiments

6.1 Experiment 1: Classification with feature transfer

6.1.1 Experiment design

We designed the first experiment to answer our first research question. Our goal is to implement a taxonomic classifier based on visual similarity features only. At first, we can begin with the **paradigm selection** arguments. We select a supervised regime for several reasons. Firstly, we want to leverage the available taxonomic labels for each of the images in our dataset, which are verified and highly reliable. Moreover, evolutionary convergence and morphological disparity demonstrate the challenges and ambiguity of relying on the visual similarity clusters without any label (unsupervised). However, it might be argued that our dataset has class imbalance (discussed earlier in the Dataset section), which can make a case against the supervised regime because it relies on the quality and quantity of labels. In reply to this, we can present the data-based (e.g., resampling with augmentation), model-based (e.g., balanced bagging random forest, neural net with weighted loss function), and metrics-wise methods to deal with imbalanced data (Das et al., 2022). Besides, the few-shot learning paradigm has also been proven robust in dealing with class imbalance. Because these available methods are also under the hood of supervised learning, we choose this paradigm for our task.

In terms of implementation, a downstream classification-based pipeline was chosen because of its flexibility. Once high-quality fine-grained features are extracted, we can explore various modeling approaches, including traditional machine learning classifiers such as Support Vector Machines (SVM) or Random Forests, as well as neural network architectures with fully connected layers for parameter learning. For feature extraction, we select a convolution-based pre-trained model to reduce training effort. Convolution networks can extract hierarchical features and are robust against local invariance. We do not train parameters of the pre-trained model through deep training because our goal is to implement feature transfer only. Finally, we train a random forest with the extracted features and opt for the classification performance with the test data. Random Forests are well-suited for high-dimensional classification tasks because they aggregate predictions from an ensemble of decision trees. In this voting-based approach, the influence of noisy or non-discriminative features is naturally diminished, as they contribute less consistently to correct classifications and are effectively 'voted out' by the majority. Moreover, with random forest, it is quite easy to perform feature ranking, which also gives it an edge in terms of interpretability. During evaluation, we will take a random guess as a baseline and compare our performance metrics with other feature-transfer-based works from the literature. In addition, to address the **imbalance** in our dataset, we employ a **weighted Random Forest**, which assigns higher penalties to misclassifications of the minority class, thereby improving the model's sensitivity towards underrepresented categories.

Albanese et al. (2015) chose to use a Random Forest classifier after the feature selection step was done. Hofmann et al. (2024) demonstrated that the experiment with the sequential classifier architecture did not learn the taxonomic hierarchy any better than the parallel model. In the parallel classifier approach, the extracted features were fed into independent classifiers for different taxons. Based on this, our pipeline will consist of feature extraction using a pre-trained ConvNextTiny model followed by a supervised classification with a Random Forest classifier. Labels for subclass and order were extracted independently to ensure distinct classification pathways.

Our pre-trained model ConvNeXt, introduced by Liu et al. (2022), achieved state-of-the-art performance in image classification on the ImageNet benchmark by building on the strengths of convolutional networks like ResNet50 while integrating key advancements from Swin Transformers. The transformer architecture introduces global receptive fields and self-attention, enabling ConvNeXt to capture long-range dependencies and contextual relationships, which traditional convolutional networks struggle with due to their localized kernel-based processing.

This pipeline will integrate the strengths of deep learning-based feature extraction with a classical machine learning classifier, ensuring both computational efficiency and interpretability.

6.1.2 Experiment Pipeline

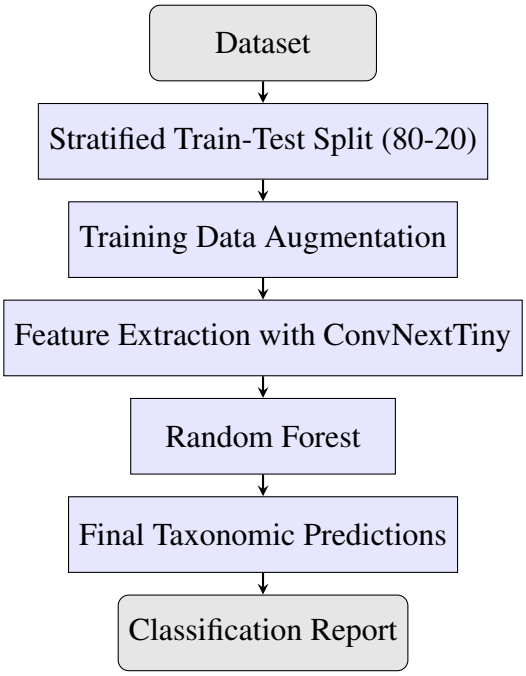

Figure 6.1: Downstream classification pipeline

Pre-processing image data

Class imbalance To address the class imbalance, random Gaussian noise addition was applied. The three most frequent orders were *Cardiida* (15,237 samples), *Pectinida* (13,344 samples), and *Venerida* (9,826 samples), whereas the least frequent were *Cyamioidea* (7 samples),

Trigoniida (67 samples), and *Gastrochaenida* (71 samples). For subclass classification, the most frequent were *Imparidentia* (33,852 samples), *Pteriomorphia* (29,145 samples), and *Protobranchia* (3,273 samples), while the least frequent was *Paleoheterodonta* (67 samples). A stratified train-test split was employed, ensuring a proportional representation of each class.

Dataset dimension variations Resizing was done to (224, 224) as a part of the standard deep learning pipeline. The size (224,224) was chosen because of the requirements of the ConvNeXt model.

Normalization The official documentation of the convNeXtTiny in TensorFlow keras (Google Research Team, 2025) mentions that the model expects pixel values in unit-8 or floats in the range 0 to 255. So no such normalization in this direction was conducted. The model has a normalization layer inside itself.

Feature Extraction with ConvNextTiny

ConvNextTiny, a compact version of the ConvNext architecture, was utilized for feature extraction due to its efficiency in capturing hierarchical representations. The model was pre-trained on ImageNet and employed as a fixed feature extractor, meaning its weights were frozen to prevent additional training. The feature extraction resulted in embeddings of size 768 per image. The ConvNextTiny model consists of approximately 27.8 million parameters, which could lead to over-fitting. Therefore, it was used in a frozen state, and the number of trainable parameters was reduced to zero. Without constraints on computational resources and time for gradient computation, selectively unfreezing the last few layers of the feature extractor could offer fine-tuning, a well-balanced alternative.

Random Forest for Taxonomic Classification

Following feature extraction, a Random Forest classifier was trained on the extracted features to classify instances into their respective taxonomic groups. Random Forest was chosen due to its robustness against overfitting, ability to handle high-dimensional data, and interpretability in decision-making. The classifier was trained using 100 decision trees (`n_estimators = 100`) with a fixed random seed for reproducibility. Given the high-dimensional nature of ConvNext-extracted features, Random Forest effectively aggregated multiple decision paths, improving classification reliability across different taxonomic groups.

6.2 Experiment 2: Distance-guided embedding learning

6.2.1 Experiment design

In response to our second research question, we developed this experiment to explore the possibility of learning biologically meaningful embeddings from visual morphology data, guided by the phylogenetic distance between taxonomic families.

Rather than relying solely on explicit class labels, we can incorporate genetic distance into the supervised training framework. Previous studies (Hoyal Cuthill et al., 2019; Hunt and Pedersen, 2022; Pedersen et al., 2019) have successfully utilized similarity metrics to infer relationships between taxa. Albanese et al. (2015) introduces phylogenetic-based feature weighting to rank the samples by their contribution to differentiation and improvement of biological analysis without predefined taxonomies. Inspired by these works, we will compute pairwise distances between family-level visual embeddings and compare them with phylogenetic tree distances to assess their alignment.

In this experiment, pre-trained ResNet50 was employed to extract feature representations from images. The learned representations were then aggregated using weighted attention-based pooling to derive family-wise embedding vectors. Finally, cosine similarity was computed between family feature vectors, and the resulting visual distance matrix was structured to align with the phylogenetic distance table.

ResNet50's residual connections and ability to **extract hierarchical features** make it particularly suited for capturing complex patterns in visual data. Furthermore, by leveraging a pre-trained version of ResNet50, we can benefit from a feature space that has already been trained on large-scale data. The output features from ResNet50 will be used as embeddings for each image, which will then be aggregated at the family level.

The next step in the experiment is to **aggregate the extracted embeddings** into a single vector per family. To achieve this, we will use a weighted attention pooling mechanism. This choice is informed by the need to assign varying importance to different parts of the feature map, based on their relevance to the family-specific representation.

The **weighted attention pooling mechanism** is implemented as follows:

Step 1: Each family's feature map, extracted from the ResNet50 model, will be passed through an attention network that learns to assign different weights to the features.

Step 2: These attention weights will be computed via a softmax function applied to the output of a linear transformation of the feature map. **This ensures that the most relevant features for family classification are weighted more heavily, while less relevant features are down-weighted.**

Step 3: The weighted sum of the feature map will then form the aggregated family embedding.

Step 4: The learned embeddings will be normalized to unit vectors, ensuring that they can be directly compared using cosine similarity.

To **evaluate** the quality of the learned family embeddings, we will compute the cosine similarity between each pair of family embeddings. This visual similarity matrix will then be compared with the phylogenetic distance matrix, which quantifies the genetic distances between families based on their evolutionary relationships.

We will define a **loss function** that minimizes the difference between the visual and genetic distance matrices. Specifically, we will use Mean Squared Error (MSE) between the upper triangular values (excluding the diagonal) of the two matrices, ensuring that we focus on the pairwise relationships between families. This will allow us to directly optimize the embedding space to be consistent with known phylogenetic information.

6.2.2 Experiment Pipeline

Below is figure demonstrating the major steps in this experiment.

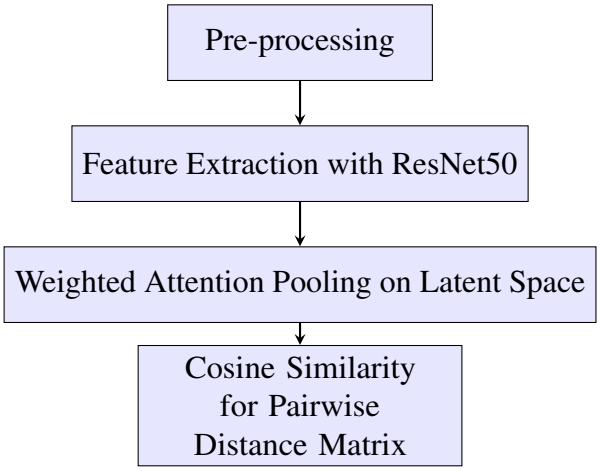

Figure 6.2: Experiment 2 pipeline for distance-aware embedding learning

Pre-processing image data

To address the class imbalance, the majority classes were downsampled and the minority classes were upsampled through random augmentations (e.g., HorizontalFlip, Rotation, ColorJitter, Affine). Considering the limited available GPU memory in Google Colab Pro+, the total number of samples was kept (150*74) = 11,100 [150 samples per family].

Because the huggingFace backbone model was pre-trained on imagenet-1K dataset and the model itself requires the pixel values in [0,1] range, transformations on the whole dataset was applied: firstly, to convert the pixel values to the desired range and then to convert into such a distribution so that the mean and the standard deviation of the imagenet-1K dataset pixels are maintained. Resizing was done to (224, 224) as a part of the standard deep learning pipeline. The size (224,224) was chosen because it is the requirement of the ResNet model.

Feature extraction

In this experiment, a hugging face SWAV model with a ResNet-50 backbone was used, pre-trained on ImageNet. The model processes input images through a series of convolutional and batch normalization layers, producing feature vectors[shape: (num of samples, 2048)] in a latent space.

Weighted Attention-based Pooling

After obtaining image-level embeddings using ResNet50, the next step was to aggregate these embeddings at the family level. Instead of simple averaging, weighted attention-based pooling was applied to capture non-linear relationships in the latent space, allowing the model to emphasize more informative feature vectors within each family while down-weighting less representative samples.

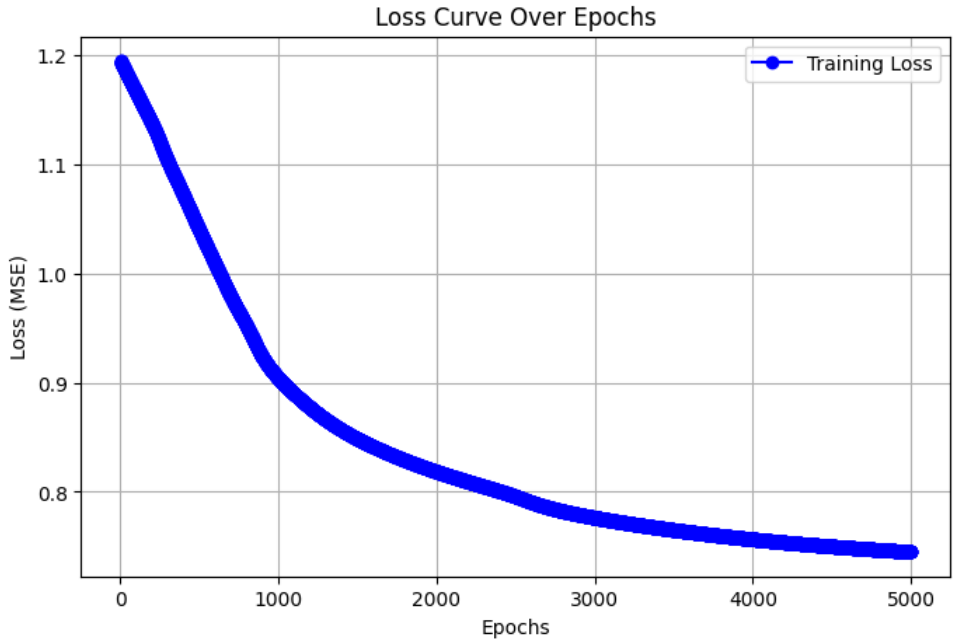

Figure 6.3: Training Process

The pooling was performed using an attention-based mechanism, where each image embedding within a family was assigned a learnable weight. These weights were computed using a softmax function over a linear attention layer, emphasizing the most informative features while down-weighting redundant information. The final family-level representation was computed as:

$$\mathbf{F}_i = \sum_{j=1}^{N_i} w_j \mathbf{x}_j \tag{6.1}$$

where $\mathbf{F}_i$ is the feature vector for family i, $\mathbf{x}_j$ represents the embedding of an individual image, and w_j is the learned weight assigned to $\mathbf{x}_j$.

The attention pool parameters were learned by minimizing the loss function (Mean squared error) between the genetic and visual distance matrix entries. To make this meaningful, several steps were necessary. Firstly, it made sense to keep only the common families in both matrices in the same order. Secondly, the MSE was computed only with the upper triangle values of the symmetric square matrices so that meaningless redundancy is avoided. To clarify further, in a square and symmetric matrix, m[a][b]= m[b][a].Last but not the least, the genetic distance matrix was normalized to match the same behavior as the cosine distance matrix. The range [0-1.7] was normalized to [-1,1].

Finally, 5000 epochs were run using the ADAM optimizer and learning rate 1e-4, which reduced the loss from 1.1946 to 0.7451. Figure 5.3 plots the loss function against the number of epochs.

Cosine Similarity for Pairwise Distance Computation

To assess how well the learned visual representations capture taxonomic relationships, cosine similarity was computed between family feature vectors. Cosine similarity was chosen because

it measures the angular distance between two vectors, making it effective for comparing high-dimensional embeddings. The cosine similarity between two families A and B was defined as:

$$d(A, B) = 1 - \frac{\mathbf{F}_A \cdot \mathbf{F}_B}{\|\mathbf{F}_A\| \|\mathbf{F}_B\|} \tag{6.2}$$

where $\mathbf{F}_A$ and $\mathbf{F}_B$ are the feature vectors for families A and B. Since cosine similarity focuses on directional alignment rather than magnitude, it is well-suited for evaluating structural similarity in the feature space.

The 74 family feature vectors were stacked first. As a first step to compute the pair-wise cosine similarity, each of them was L2-normalized. Finally, the dot product between the unit vectors and their transposed was computed. This resulted in a (74*74) visual distance matrix. As cosine similarity, their values belonged in the [-1,1] range where the highest value 1 (cos0) signifies perfect similarity and the lowest -1 expresses the most dissimilarity.

Distance Matrix Alignment with Taxonomic Hierarchy

The final step involved structuring the computed visual similarity matrix to align with the hierarchical structure of the taxonomic labels. The ground-truth genetic distance matrix, derived from phylogenetic data, was used as a reference. Families were ordered according to their placement in higher taxonomic levels (orders), ensuring that the visual similarity matrix mirrored the organization of the genetic distance matrix.

7 Results and Discussion

7.1 Evaluation metrics

The evaluation of the two experiments relies on appropriate metrics that measure classification performance and similarity alignment, ensuring that the selected models effectively capture taxonomic structures and genetic relationships.

7.1.1 First Experiment: Classifier with feature transfer

For subclass and order classification using ConvNext feature extraction and a random forest classifier, we employ standard classification metrics:

- **Overall Accuracy:** Measures the percentage of correctly classified instances across all taxonomic levels. High accuracy indicates that the extracted features retain relevant taxonomic information.
- **F1 Score:** Defined as:
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7.1}$$
It balances precision and recall, making it particularly useful when dealing with class imbalances, which are common in taxonomic datasets.
- **Recall:** Measures the proportion of correctly classified instances out of all actual instances for a given class:
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{7.2}$$
A high recall ensures that closely related taxa are not mistakenly omitted, which is crucial for hierarchical taxonomic classification.
- **Confusion Matrix:** Provides a breakdown of correct and incorrect predictions per class, offering insight into misclassifications and whether errors occur between closely related taxa.

These metrics ensure that the classification approach is both precise and sensitive to the hierarchical taxonomic relationships.

7.1.2 Second Experiment: Distance-aware embedding learning

For the ResNet-based feature extraction experiment, we evaluate how well visual similarities align with genetic distances using the following metrics:

- **Pearson's Correlation Coefficient:** Measures the linear relationship between learned pairwise cosine distances and the given genetic family distances:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2}\sqrt{\sum(Y_i - \bar{Y})^2}} \tag{7.3}$$

  A high Pearson correlation indicates that visually extracted features correspond well with genetic distances, validating the effectiveness of self-supervised learning.

- **Heatmap Visualization:** Displays the pairwise similarity matrix, offering an intuitive way to analyze whether visually similar taxa cluster in alignment with genetic families. Misalignment in the heatmap can reveal inconsistencies in feature extraction.

These metrics collectively validate whether the learned visual embeddings align with biological taxonomy and ensure that learned features meaningfully reflect phylogenetic structures.

7.2 Results

7.2.1 Classification with Feature Transfer

Table 6.1 is the classification report for subclass level. Number of support(ground truth) per class(subclass) demonstrates the class imbalance in the test dataset induced by stratified split (train 80%, test 20%). While the majority sample classes do relatively well in terms of accuracy metrics, the minority sample classes bring the overall accuracy lower. For example, the class 3 with only 13 ground truth supports is never predicted.

Class	Precision	Recall	F1-Score	Support
0	0.00	0.00	0.00	521
1	0.50	0.01	0.02	589
2	0.64	0.87	0.74	6771
3	0.00	0.00	0.00	13
4	0.00	0.00	0.00	655
5	0.76	0.68	0.72	5829
Accuracy	0.69			14378
Macro Avg	0.32	0.26	0.25	14378
Weighted Avg	0.63	0.69	0.64	14378

Table 7.1: Classification Report or the 6 subclasses (0-5).

The same pattern is observed for the order level classification (Table 6.2, page 18). For example, Class 19 with the highest number of support shows promising F1 score and recall (0.65) while the class 5 with 1 support sample is never predicted. The overall accuracy score is more affected this time as the number of classes increases.

In this study, we adopt the **random guess** performance reported in prior literature using the same dataset as a **baseline** for comparison. Relative to this baseline, our feature transfer

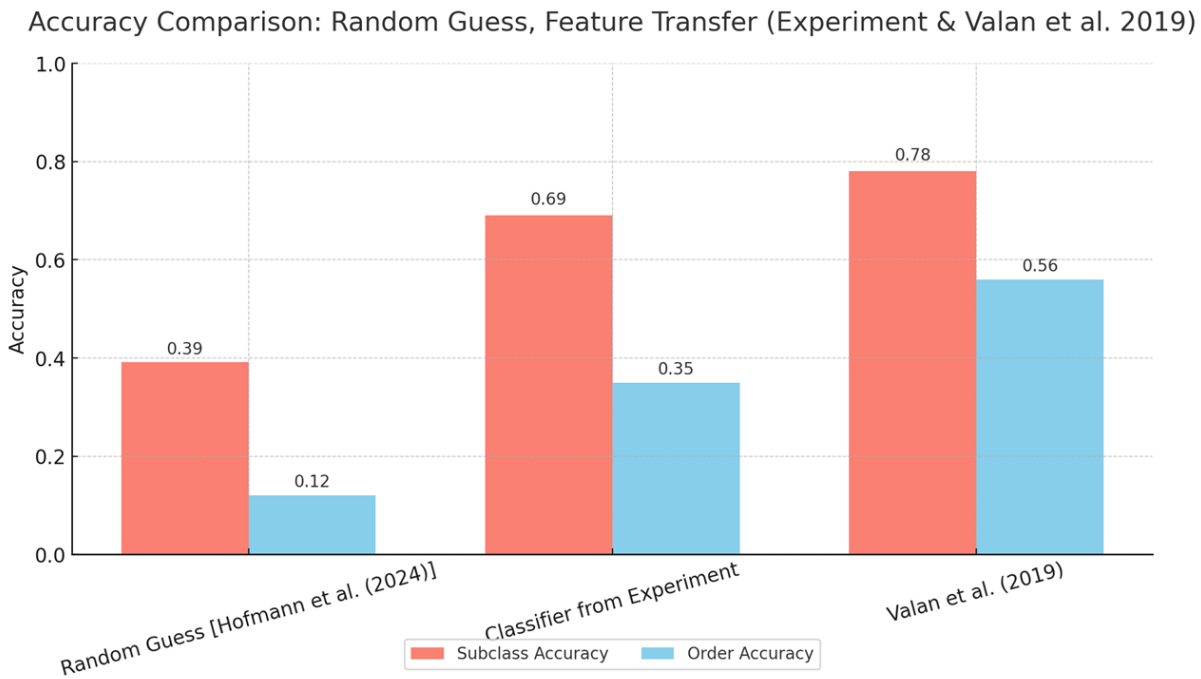

Figure 7.1: Accuracy comparison with baseline

approach yields significantly improved accuracy. Notably, at the subclass level, our model's performance aligns well with other feature transfer-based methods, such as those presented by Valan et al. (2019). However, at the order level, our classification accuracy is noticeably lower compared to similar studies. This discrepancy suggests that our method may face **challenges in distinguishing fine-grained morphological features**. In their work, Valan et al. (2019) employed a **feature fusion** strategy, combining representations from multiple intermediate layers of a VGG16 network, which likely contributed to their improved performance on finer taxonomic levels.

Figure 7.2 at page 22 compares the standard random forest metrics with two other variants of it. In weighted Random Forest, class weights are adjusted to penalize misclassification of minority classes more. During training, the algorithm applies a higher weight to errors made on minority classes. These weights are passed to the base decision trees. The performance with this variant at the **subclass** level **(accuracy: 65%)** is highly **significant** for our imbalanced dataset. A bagging ensemble, on the other hand, trains each base estimator (typically a decision tree) on a balanced subset of the data. Each bootstrap sample is generated by undersampling the majority class so that the classes are balanced. Therefore, the drop in average accuracy (38%) with this variant was pretty much expected with the imbalanced support set of our experiment.

Limitations, improvement of feature transfer pipeline

Table 7.2, page 23 summarizes the limitations of our first experiment. To mitigate class imbalance, the number of samples in minority classes was increased to 1,000 using Gaussian noise augmentation. While classes with fewer than 1,000 samples were upsampled, the majority classes remained unchanged, resulting in a persistent imbalance—with sample distributions ranging from 1,000 to 15,000, maintaining a 15:1 skew.

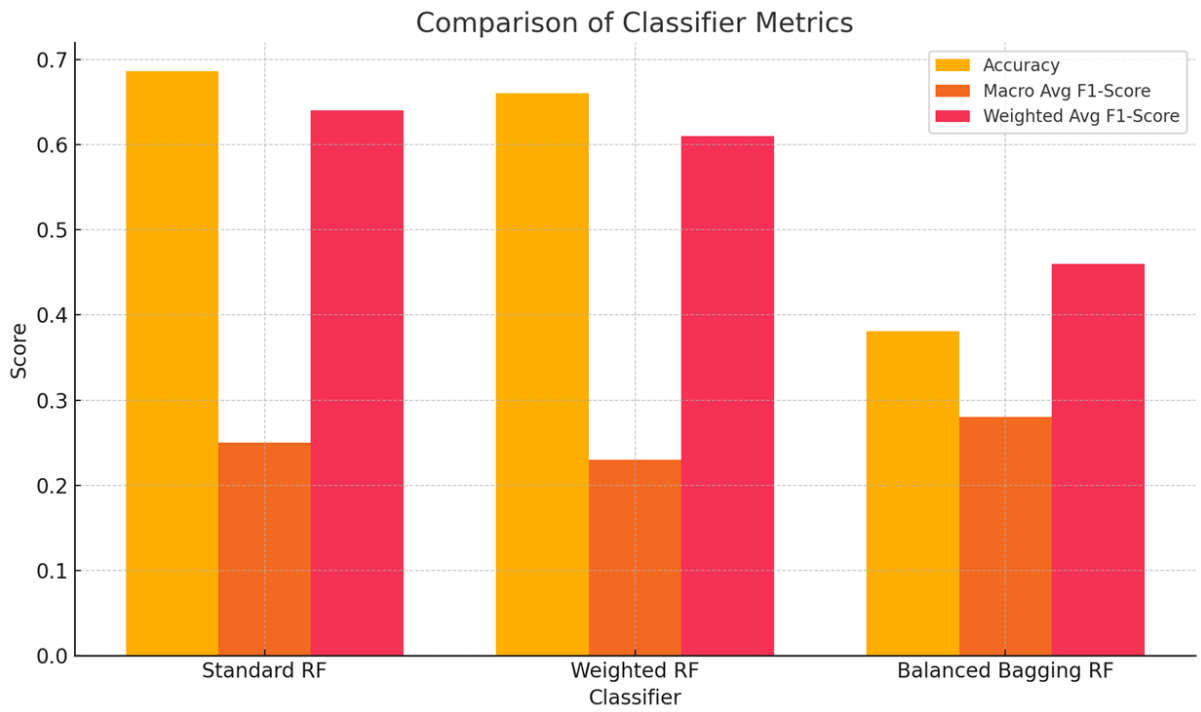

Figure 7.2: Subclass level accuracy comparison with three(3) different random forest variants

Furthermore, during feature extraction due to **zero(0) repetition**, each batch went through augmentation only once (augmentation model inside extraction model), limiting the model's exposure to varied representations. This constrained its ability to learn robust and generalized features.

7.2.2 Distance-guided embedding learning Experiment

Distance-guided embedding learning

Cosine Similarity and Distance Matrix Evaluation The computed visual similarity matrix, based on cosine distances between family-level feature vectors, was compared to the ground-truth genetic distance matrix. **Pearson's correlation coefficient** between the two matrices was found to be **0.4032 (p-value: 5.7419e-109)**, indicating the degree to which learned morphological features align with known phylogenetic relationships.

Interpretation: Interpreting the Results Based on Similarity Scale (Where higher values mean more similarity and -1 is extremely different)

Pearson Correlation (0.4032) implies moderate positive linear Relationship. Since both genetic distances (after transformation) and cosine similarity now follow the same convention (higher = more similar, lower = more different), a moderate positive Pearson correlation (0.4032) means there is some alignment between genetic similarity and visual similarity. However, the relationship is not perfectly linear, meaning that genetic similarity does not fully explain visual similarity.The transformation of genetic distances partially aligns with visual features but is not fully predictive.

Table 7.2: Limitations of the **feature transfer experiment** and plausible enhancements

Limitations	**Implication**	**Improvement**
Class imbalance: Only **data-based** methods	Accuracy at order taxon drops, F1-score drops	**Model-based:** - Hyper-parameter tuning of Random Forest - SVM
Class imbalance: Only up-sampling With limited augmentation	Without down-sampling the majority classes, imbalance skew persists	**Resampling** **Repetitions** in `train_features` extractions Comprehensive **augmentation**
High-dimensional feature space	Weak, noisy and non-discriminative features remaining	- Fusion - Ranking - Reduction with PCA

P-values were found extremely small meaning results are statistically significant. The tiny p-values confirm that the correlation is real (not due to chance), even if it's not very strong. Overall, my findings are meaningful, but genetic and visual similarity are only moderately related.

Heatmap Visualization To provide an intuitive comparison, the genetic distance matrix and the computed visual similarity matrix were visualized using heatmaps. Figure 7.3 shows the two matrices side by side, allowing for a direct examination of clustering patterns and structural alignment.

We plotted both the distance matrix in their square and symmetric form. Values of both of them were in the range of [-1,1] range, where the highest value 1 means the highest similarity (self-similarity), the light-colored diagonal. And the least value -1 will mean lowest similarity in our case which will be the darkest. The families were arranged according to their orders before plotting. While the overall pattern of brighter and darker clustered areas match in both of the pictures, significant differences can be observed upon scrutiny.

Limitations, improvement of embedding learning pipeline

Taxonomically aware downsampling ensures that when reducing the number of samples from a highly represented family, for example, from 20,000 to 1,000—we retain the full range of taxonomic diversity present in the original set. This means that all distinct genera and species originally represented should still be included in the downsampled subset. In essence, the goal is to preserve as much taxonomic variation as possible, rather than just **randomly** selecting samples. **No training on our data** due to using only pre-trained weights; improvement is possible by **fine-tuning** intermediate to final layers to enhance features and tackle overfitting.

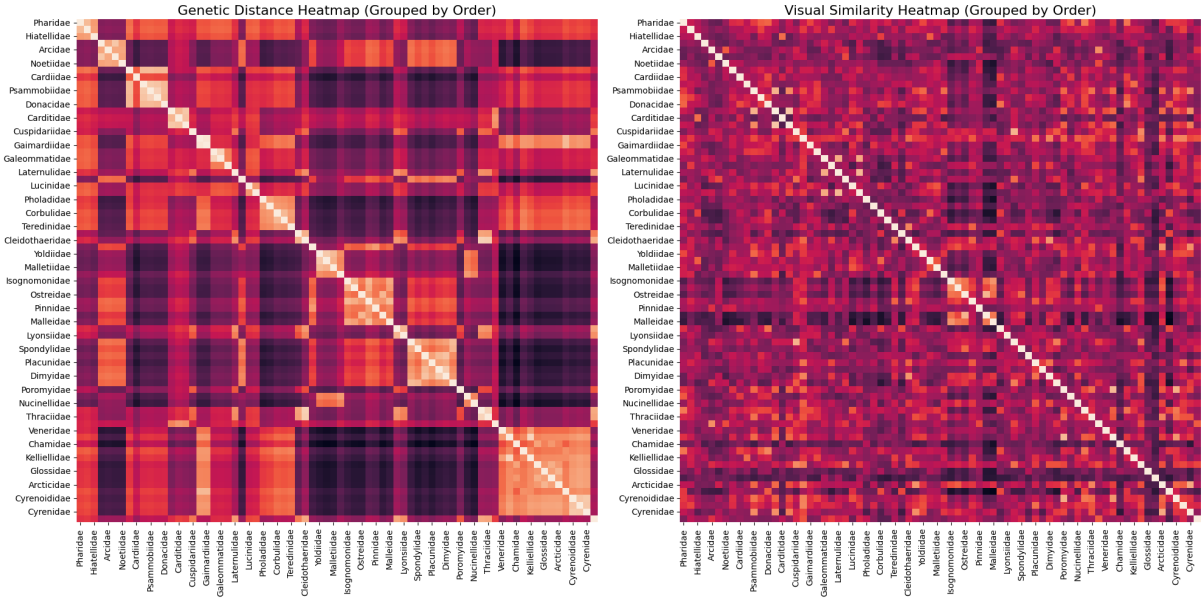

Figure 7.3: Comparison of Genetic Distance Heatmap (left) and Visual Similarity Heatmap (right).

Class	Precision	Recall	F1-Score	Support
0	0.67	0.17	0.27	316
1	0.29	0.28	0.29	1365
2	0.40	0.44	0.42	3048
3	0.17	0.35	0.23	589
4	0.26	0.19	0.22	196
5	0.00	0.00	0.00	1
6	0.09	0.55	0.15	22
7	0.18	0.09	0.12	229
8	0.04	0.36	0.07	14
9	0.10	0.46	0.17	39
10	0.26	0.15	0.19	310
11	0.20	0.21	0.20	721
12	0.14	0.04	0.06	455
13	0.02	0.08	0.03	25
14	0.49	0.25	0.33	828
15	0.12	0.06	0.08	399
16	0.10	0.07	0.08	211
17	0.44	0.34	0.38	657
18	0.10	0.08	0.09	105
19	0.65	0.65	0.65	2669
20	0.05	0.36	0.09	33
21	0.28	0.18	0.22	44
22	0.03	0.04	0.03	107
23	0.04	0.54	0.07	13
24	0.31	0.29	0.30	1965
25	0.05	0.18	0.08	17
Accuracy		0.35		14378
Macro Avg	0.21	0.25	0.19	14378
Weighted Avg	0.38	0.35	0.36	14378

Table 7.3: Classification Report for for the 26 subclasses (0-25)

Correlation Method	Value
Pearson's correlation coefficient	0.4032 (p-value: 5.7419e-109)

Table 7.4: Correlation Methods and Their Values

8 Future Enhancement

Future work includes exploring **fine-tuning** strategies to enhance model performance. For classification tasks, fine-tuning involves adjusting neural network parameters to improve on feature transfer and learning from the specific characteristics of the dataset. In the context of embedding learning, fine-tuning can help achieve better feature representations, enabling more accurate similarity measurements and clustering.

Another important area is **handling class imbalance**, which is especially relevant in biological datasets with uneven taxonomic representation. One approach is taxonomically aware re-sampling, where samples are selected to preserve hierarchical diversity across classes. Additionally, training neural networks with weighted loss functions can help mitigate the bias toward overrepresented classes. Exploring **few-shot learning** techniques may also allow the model to generalize from limited examples, which is particularly valuable when data availability is constrained for rare species.

Lastly, **Hierarchy-Guided Neural Networks (HGNNs)** present a promising direction. These networks incorporate biological taxonomic hierarchies, such as genus and species, simultaneously, into the learning process to improve classification. Approaches like dual-ResNet architectures combined with joint loss functions can be leveraged to optimize performance across multiple taxonomic levels. Such models have shown improved accuracy when trained on small, imbalanced datasets, making them suitable for real-world biological classification tasks (Elhamod and Tung, 2020).

9 Conclusion

Our first experiment focused on taxonomic classification at the subclass and order levels, addressing our initial research question. The goal was to evaluate how well visual features can serve in the absence of genetic similarity data in this context. Using feature transfer, we achieved an average classification accuracy**(subclass: 69%, order: 35%)** that was significantly higher than random chance, consistent with results reported in similar studies. This baseline performance suggests strong potential for improvement through further feature engineering and parameter optimization. Additionally, the challenges posed by class imbalance led us to explore the potential of N-shot learning paradigms, which we identified as a promising direction for future experimentation with our dataset.

In our second research question, we investigated whether a meaningful visual embedding space could be learned using phylogenetic distance as guidance. Our results showed a satisfactory positive linear correlation (Pearson: **0.4032**) with a very low p-value(**5.7419e-109**). This outcome is encouraging and motivates further refinement of our approach through taxonomically informed resampling and fine-tuning.

Moreover, in this study, we systematically categorized various similarity learning methods into appropriate paradigms by analyzing their training methodologies, learning objectives, and the strengths and weaknesses that arise from them. To conclude, we can mention some of the technical details of our implementation. The experiments were developed using Jupyter notebooks with Google Colab and run on an A100 GPU available with Pro+ subscription. The notebooks have been uploaded to the student's GitLab account of TU Ilmneau. The augmented and balanced image dataset as well as the extracted feature vector tensors, have been saved as a tensor for further usage in future experiments. The first experiment was implemented using TensorFlow, while the second was developed in PyTorch.

Bibliography

Albanese, D., De Filippo, C., Cavalieri, D., and Donati, C. (2015). Explaining diversity in metagenomic datasets by phylogenetic-based feature weighting. *PLoS Computational Biology*, 11(8):e1004186.

Antoniou, A., Edwards, H., and Storkey, A. (2018). How to train your maml. *arXiv preprint arXiv:1810.09502*.

Bieler, R., Mikkelsen, P. M., Collins, T. M., Glover, E. A., González, V. L., Graf, D. L., Harper, E. M., Healy, J. M., Kawauchi, G. Y., Sharma, P. P., et al. (2014). Investigating the bivalve tree of life–an exemplar-based approach combining molecular and novel morphological characters. *Invertebrate Systematics*, 28(1):32–115.

Bouchet, P., Decock, W., Vanhoorne, B., and Vandepitte, L. (2023). Marine biodiversity discovery: the metrics of new species descriptions. *Frontiers in Marine Science*, 10:929989.

Bouchet, P., Lozouet, P., Maestrati, P., and Heros, V. (2002). Assessing the magnitude of species richness in tropical marine environments: exceptionally high numbers of molluscs at a new caledonia site. *Biological Journal of the Linnean Society*, 75(4):421–436.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*, pages 132–149.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9912–9924.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. (2019). A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*.

Das, S., Mullick, S. S., and Zelinka, I. (2022). On supervised class-imbalanced learning: An updated perspective and some key challenges. *IEEE Transactions on Artificial Intelligence*, 3(6):973–991.

Duan, C., Feng, Y., Zhou, M., Xiong, X., Wang, Y., Qiang, B., and Jia, W. (2023). Multilevel similarity-aware deep metric learning for fine-grained image retrieval. *IEEE Transactions on Industrial Informatics*, 19(8):9173–9187.

Elhamod, M. and Tung, F. (2020). Hierarchy-guided neural network for species classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3957–3966.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1126–1135. JMLR.org.

García-Souto, D., Sumner-Hempel, A., Fervenza, S., Pérez-García, C., Torreiro, A., González-Romero, R., Eirín-López, J. M., Morán, P., and Pasantes, J. J. (2017). Detection of invasive and cryptic species in marine mussels (bivalvia, mytilidae): A chromosomal perspective. *Journal for Nature Conservation*, 39:58–67.

Ge, W. (2018). Deep metric learning with hierarchical triplet loss. In *European Conference on Computer Vision (ECCV)*, pages 269–285.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Google Research Team (2025). Convnexttiny - keras applications. Accessed: 2025-03-03.

Grill, J. B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., and Buchatskaya, E. e. a. (2020). Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21271–21284.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738.

Hofmann, M., Kiel, S., Kösters, L. M., Wäldchen, J., and Mäder, P. (2024). Inferring taxonomic affinities and genetic distances using morphological features extracted from specimen images. *Systematic Biology*.

Hoyal Cuthill, J. F., Guttenberg, N., Ledger, S., Crowther, R., and Huertas, B. (2019). Deep learning on butterfly phenotypes tests evolution's oldest mathematical model. *Science Advances*, 5(4):eaaw4967.

Hunt, R. and Pedersen, K. S. (2022). Rove-tree-11: The not-so-wild rover, a hierarchically structured image dataset for deep metric learning research. *Asian Conference on Computer Vision (ACCV)*, pages 2967–2983.

Karbstein, K., Kösters, L., Hodač, L., Hofmann, M., Hörandl, E., Tomasello, S., Wagner, N. D., Emerson, B. C., Albach, D. C., Scheu, S., Bradler, S., de Vries, J., Irisarri, I., Li, H., Soltis, P., Mäder, P., and Wäldchen, J. (2024). Species delimitation 4.0: integrative taxonomy meets artificial intelligence. *Trends in Ecology & Evolution*, 39(8):771–784.

Katolikova, M., Khaitov, V., Väinölä, R., Gantsevich, M., and Strelkov, P. (2016). Genetic, ecological and morphological distinctness of the blue mussels *mytilus trossulus* gould and *m. edulis* l. in the white sea. *PLOS ONE*, 11(4):e0152963.

Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2.

Li, Z., Zhou, F., Chen, F., and Li, H. (2017). Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s.

Pedersen, M., Bruslund Haurum, J., Gade, R., and Moeslund, T. B. (2019). Detection of marine animals in a new underwater dataset with varying visibility. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Pojeta, J. (2000). Cambrian pelecypoda (mollusca). *American Malacological Bulletin*, 15:157–166.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Valan, M., Mokonyi, K., Maki, A., Vondráček, D., and Ronquist, F. (2019). Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks. *Systematic Biology*, 68(6):876–895.

Vinyals, O., Blundell, C., Lillicrap, T., and Wierstra, D. (2016). Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29.

Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R., and Robertson, N. M. (2019). Ranked list loss for deep metric learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5207–5216.

WoRMS Editorial Board (2024). World Register of Marine Species (WoRMS). Online Database. Accessed: February 2024.