Deep Learning for 3D Fossil Classification from Micro-CT Slices: A Transfer Learning Pipeline with Interactive Dashboard Deployment

A Comprehensive Technical Report

Authors: Abdelghafour HALIMI
Institution: KAUST

Date: August 26, 2025

Abstract

This work presents a comprehensive deep learning pipeline for the automated classification of fossil species from 2D micro-CT slices derived from 3D models. We developed a scientifically rigorous dataset from 97 micro-CT scanned fossils across 27 species, selecting 12 species with sufficient data for robust machine learning. Our methodology prevents data leakage through specimen-level splitting, yielding 109,617 high-quality 2D slices (44,103 train, 14,046 validation, 51,468 test). We trained seven state-of-the-art 2D CNN architectures using transfer learning, achieving 95.64% test accuracy with our final ensemble combining ConvNeXt-Large and EfficientNetV2-Small. The ensemble demonstrates exceptional reliability with 99.6% top-3 accuracy and 0.998 AUC across all species. We deployed this system through an interactive Streamlit dashboard supporting real-time slice classification and 3D slice matching using advanced similarity metrics (SSIM, NCC, Dice coefficient). This work establishes new benchmarks for AI-assisted paleontological identification while providing a complete, reproducible framework for fossil classification research.

Contents

1 Introduction

The field of paleontology is undergoing a profound data revolution, driven by the widespread adoption of high-resolution, non-destructive imaging technologies like micro-Computed Tomography (micro-CT) [25, 24]. These methods have become indispensable for investigating the intricate anatomy of fossil specimens, providing unprecedented access to fragile structures that would be destroyed by traditional preparation techniques [1]. This technological leap, however, has introduced a commensurate "big data" bottleneck in post-processing. As imaging resolutions increase, the resulting volumetric datasets have become immense, making the manual process of digitally isolating a fossil from its surrounding rock matrix—a step known as segmentation—the most critical and time-consuming task in the entire workflow. This challenge is particularly arduous for low-contrast specimens, such as calcareous fossils embedded in a carbonate-rich matrix, where traditional algorithms fail [24].

In response to this critical bottleneck, a new paradigm has emerged from computer science: deep learning. As a subset of machine learning, deep neural networks have demonstrated a remarkable capacity for automated image analysis, and paleontologists are now deploying them to automate the segmentation and classification of fossil data to drastically reduce processing times. The initial phase of this research focused on demonstrating feasibility—proving that a standard Convolutional Neural Network (CNN), such as a U-Net, could successfully segment a fossil from its matrix [24, 25]. Seminal works have shown that these methods can achieve segmentation accuracies comparable to meticulous manual work but in a fraction of the time, representing a fundamental revolution in paleontological analysis, with key examples summarized in Table 1.

1.1 State-of-the-Art Applications

The following table summarizes several key studies that exemplify the current state-of-the-art in applying deep learning to paleontological analysis.

Table 1: Summary of State-of-the-Art Deep Learning Applications in Fossil Analysis

Reference	Fossil Type & Domain	Imaging Modality	Primary Task & Core DL Model/Technique	Key Contribution / Reported Metric
Mietchen et al., 2024 [25]	Vertebrate limb bone	X-ray micro-CT (Synchrotron)	Segmentation / U-Net	Dice similarity of 0.96 with <2% training data.
Reid et al., 2023 [24]	Calcareous shelly invertebrates	Micro-CT	Segmentation / Deep Learning (General)	Workflow for segmenting low-contrast fossils (calcareous shells in calcareous matrix).
Ferreira et al., 2023 [16]	Foraminifera microfossils	2D Images	Synthetic Data Generation, Classification, Segmentation / Hierarchical ViT & StyleGAN	FID score of 14.88 for synthetic images; enables few-shot semantic segmentation.

Reference	Fossil Type & Domain	Imaging Modality	Primary Task & Core DL Model/Technique	Key Contribution / Reported Metric
Hou et al., 2023 [17]	Fusulinid microfossils	2D Images	Classification & Identification / Multi-view Ensemble (OGS)	Highest agreement with human experts; improved performance with limited data.

The success of these early efforts quickly brought the next major challenge to the forefront: the need for robust, scalable, and data-efficient systems tailored to the unique constraints of paleontological science. This has spurred a second wave of innovation centered on methodological rigor and high-performance classification. This project contributes directly to this second wave by presenting a complete, end-to-end deep learning pipeline for the automated classification of fossil species from 2D micro-CT slices. We leverage transfer learning with state-of-the-art 2D CNNs—a powerful and efficient strategy for overcoming the data scarcity typical in paleontology [3].

This work makes three primary contributions to the field:

1. We develop a scientifically rigorous dataset creation methodology founded on **specimen-level splitting**. This approach prevents the catastrophic data leakage that occurs in slice-level splits, an issue that can artificially inflate accuracy by 30-55% and invalidate results [18]. Adhering to this best practice, learned from the medical imaging domain [4], is paramount for producing trustworthy and reproducible science.
2. We implement and systematically evaluate a comprehensive transfer learning pipeline across seven state-of-the-art 2D CNN architectures, creating an adaptive ensemble method that achieves **95.64% test accuracy** on individual slices.
3. We deploy these powerful models into an accessible, interactive dashboard, bridging the gap between cutting-edge research and practical application for paleontologists.

Together, these contributions establish a new, robust, and reproducible framework for AI-assisted paleontology, moving the field beyond proof-of-concept and toward a new era of data-driven discovery.

2 Dataset Information and Creation Methodology

2.1 Source Data and Species Selection

Our dataset originates from 97 high-resolution micro-CT scanned fossil specimens representing 27 unique foraminifera species, as documented in `1_Dataset_Creation/Species_Analysis.txt`. Following systematic analysis of data sufficiency requirements, we selected 12 species with $\geq$4 3D models each to ensure robust machine learning training:

- Chrysalidina (16 models) - largest dataset for training stability
- Ataxophragmium (7 models) - good morphological variation
- Baculogypsina (7 models) - challenging species with complex morphology
- Minoxia, Elphidiella, Fallotia (6 models each) - balanced representation

- Arumella, Lockhartia, Orbitoides, Rhapydionina (5 models each) - distinct morphological groups
- Alveolina, Coskinolina (4 models each) - minimal but sufficient data

2.2 Model-Level Splitting Methodology

A critical innovation in our approach is **model-level splitting** to prevent data leakage. This addresses a severe methodological flaw that can arise when training 2D models on volumetric data. Randomly splitting all 2D slices from all specimens into train, validation, and test sets—a "slice-level" split—is scientifically invalid because adjacent slices are highly correlated. This leads to train-test contamination, resulting in dramatically inflated and misleading performance metrics, which can artificially inflate accuracy by 30-55% [18].

To ensure scientific validity, we implement a **specimen-level split**, which is the only correct protocol [4]. This two-stage process guarantees that all slices from a single 3D fossil model are assigned exclusively to one data partition (training, validation, or test).

Stage 1: Initial Model Allocation

- Each species' 3D models divided using ∼20% test split.
- Remaining models allocated to training pool.
- Target: 10,000 slices per species through intelligent sampling.

Stage 2: Training/Validation Subdivision

- Training pool models further split at model level.
- Target: ∼1,000 validation slices per species (capped at 1,200).
- Brute-force optimization to minimize coefficient of variation.

This methodology, implemented in `Dataset_creation_segmented_final.ipynb`, ensures strict independence between our datasets, producing trustworthy and reproducible results.

2.3 Intelligent Slice Sampling and Preprocessing

Our slice extraction pipeline, implemented in `dataset_creation.py`, employs Otsu's method for automatic thresholding to detect fossil content, removing slices with <2% fossil pixels to eliminate noise [21]. The script `segment_fossils_black_bg.py` then applies advanced segmentation to create clean fossil silhouettes, enhancing model focus on morphological features.

2.4 Final Dataset Composition

The complete segmented dataset (`3d_fossil_dataset_segmented_final/`) contains 109,617 high-quality 224×224 RGB images with excellent balance characteristics:

- Training Set: 44,103 images (Coefficient of variation (CV): 13.91% - excellent balance)
- Validation Set: 14,046 images (Coefficient of variation (CV): 20.58% - good balance)
- Test Set: 51,468 images (Coefficient of variation (CV): 24.72% - acceptable for testing)

Representative species distribution includes Chrysalidina (9,972 total images), Ataxophragmium (9,834 images), and Baculogypsina (6,220 images), demonstrating our methodology's effectiveness in creating balanced datasets from variable source material.

3 AI Modeling Methodology and Techniques

3.1 Architecture Selection and Transfer Learning Strategy

While native 3D CNNs can capture rich inter-slice context, they are computationally expensive and data-hungry. Treating a 3D volume as a collection of 2D slices allows us to leverage powerful, efficient 2D CNNs pre-trained on massive datasets like ImageNet [2], which can sometimes outperform 3D models when 3D training data is scarce.

We evaluated seven state-of-the-art 2D CNN architectures: ConvNeXt (Base and Large) [7], EfficientNetV2 (Small and Large) [6], NASNet [8], MobileNet [9], and ResNet101V2 [5].

Table 2: Foundational 2D CNN Architectures Used in This Study

Architecture	Seminal Paper	Core Innovation
ResNet101V2 [5]	Deep Residual Learning for Image Recognition	Residual "skip" connections to enable effective training of very deep networks.
MobileNet [9]	Searching for MobileNetV3	Depthwise separable convolutions for computational efficiency, optimized via Neural Architecture Search (NAS).
NASNet [8]	Learning Transferable Architectures for Scalable Image Recognition	Automates architecture design by learning reusable "cells" on a proxy dataset, which are then scaled up.
EfficientNetV2 [6]	EfficientNetV2: Smaller Models and Faster Training	Training-aware NAS and compound scaling to jointly optimize for accuracy, parameter efficiency, and training speed.
ConvNeXt [7]	A ConvNet for the 2020s	Modernizes a standard ResNet by incorporating design principles from Vision Transformers, boosting performance.

Our **two-phase training strategy** optimizes transfer learning effectiveness:

Phase 1 - Backbone Freezing (10 epochs):

- Pre-trained backbone weights frozen.
- Only classifier head trained.
- Learning rate: 1e-4 with cosine scheduling.

Phase 2 - End-to-End Fine-tuning (20 epochs):

- Full network unfrozen for fine-tuning.

- Lower learning rate: 5e-5 with warm restarts.
- Mixed precision training for efficiency.
- Early stopping with patience=5 based on validation loss.

3.2 Advanced Data Augmentation and Preprocessing

Intelligent Bounding-Box Cropping: Automatically detects and crops to the minimal bounding box containing the specimen, maximizing fossil content per pixel.

Sophisticated Augmentation Strategy:

- CutMix [10] and Mixup [11]: Advanced regularization strategies that create synthetic training samples by mixing multiple examples.
- Geometric Transforms: Rotation (±45°), scaling (0.8-1.2×), horizontal flipping.
- Color Augmentation: Brightness/contrast adjustment, color jittering.

3.3 Training Configuration and Reproducibility

Standardized training parameters across all models ensure fair comparison:

- Input Resolution: 224×224 (base models) / 384×384 (large models)
- Batch Size: 32 (adjusted for GPU memory constraints)
- Optimizer: AdamW, which decouples weight decay from the gradient update to improve generalization [22].
- Loss Function: Categorical crossentropy with label smoothing (0.1) to regularize against overconfidence [23].
- Hardware: NVIDIA GPUs with mixed precision (fp16) training
- Reproducibility: Fixed random seeds, deterministic operations where possible

3.4 Ensemble Methodology

Our final ensemble, created in `DeepLearning_classification-ensemble_weighted_segmented-final.ipynb`, implements a **PatchEnsemble** architecture combining multiple models to improve predictive performance and robustness [12]. It includes:

- Primary Model: ConvNeXt-Large (95.12% individual accuracy)
- Secondary Model: EfficientNetV2-Small (91.82% individual accuracy)

Intelligent Switching Logic: The ensemble employs confidence-based switching for challenging species. This adaptive approach leverages complementary strengths across architectures.

4 Results and Discussion

4.1 Model Performance Leaderboard

Comprehensive evaluation documented in `3_Results/_comparison/fossil_model_comparison_report_v2.md` reveals clear performance tiers:

Tier 1 - Production Ready (>95% Accuracy):

- Ensemble (Final): 95.64% accuracy, 94.97% macro F1, 99.6% top-3 accuracy
- ConvNeXt-Large: 95.12% accuracy, 94.06% macro F1, 99.63% top-3 accuracy

Tier 2 - High Performance (90-95% Accuracy):

- NASNet: 93.69% accuracy, 92.54% macro F1
- EfficientNetV2-Large: 93.53% accuracy, 91.98% macro F1
- EfficientNetV2-Small: 91.82% accuracy, 90.45% macro F1
- ConvNeXt-Base: 90.28% accuracy, 89.12% macro F1

Tier 3 - Edge Deployment (85-90% Accuracy):

- MobileNet: 88.02% accuracy, optimized for resource-constrained environments

All models achieve exceptional calibration with AUC scores >0.98, indicating reliable probability estimates.

4.2 Final Ensemble Performance Analysis

The production ensemble achieves outstanding metrics on the 51,468-sample test set:

- Test Accuracy: 95.64%
- Macro Average: Precision 96.38%, Recall 94.52%, F1-Score 94.97%
- Weighted Average: Precision 95.94%, Recall 95.64%, F1-Score 95.43%
- Top-3 Accuracy: 99.6% (correct species in top 3 predictions)

4.3 Per-Species Performance Analysis

Species-level analysis reveals varying classification difficulty. While most species achieve F1 scores above 90%, **Baculogypsina** (75.22% F1) and **Orbitoides** (87.17% F1) prove most challenging, likely due to complex internal structures and variable preservation states.

5 Interactive Dashboard Application

5.1 Framework and Architecture

The dashboard, implemented in `4_Dashboard_App` using Streamlit [15], provides a sophisticated interface for fossil analysis with two primary applications:

- **Application 1**: AI-Powered Fossil Slice Classification (`pages/1_Fossil_DL_Classification.py`)
- **Application 2**: 3D Fossil Slice Matching (`pages/2_Fossil_Matching_Slice.py`)

5.2 Real-Time Classification Features

The classification interface provides comprehensive fossil identification capabilities, including interactive preprocessing and an advanced analysis dashboard showing probability distributions, confidence metrics, and ranked predictions. The interface seamlessly loads the production ensemble model (`fossil_classifier_final/model.keras`) and ensures preprocessing consistency.

5.3 3D Slice Matching and Correspondence Analysis

The 3D matching application implements sophisticated similarity analysis using multiple established metrics:

Core Similarity Metrics:

- SSIM (Structural Similarity Index): A perceptual metric that assesses similarity based on luminance, contrast, and structure [13].
- NCC (Normalized Cross-Correlation): A robust template matching technique invariant to linear changes in brightness and contrast [19].
- Dice Score: A statistical metric that quantifies the spatial overlap between two segmentations [14].
- ORB Feature Matching: A fast and efficient feature detection and description algorithm used for geometric correspondence [20].

Advanced Processing Pipeline: A two-stage matching process uses a coarse search with Dice coefficient and Hu moments, followed by fine-tuning with rotation analysis for orientation-invariant matching.

6 Reproducibility Checklist

6.1 Dataset Generation

```
cd 1_Dataset_Creation

# Create initial clean dataset
python dataset_creation.py

# Generate segmented dataset with black backgrounds
python run_full_segmentation_black_bg.py

# Interactive analysis and train/val/test splitting
jupyter lab Dataset_creation_segmented_final.ipynb
```

Listing 1: Dataset Creation Commands

6.2 Model Training

```
cd 2_AI_Modeling_Transfer_Learning

# Individual model training (example sequence)
jupyter lab DeepLearning_classification-mobilnet_segmented.ipynb
jupyter lab DeepLearning_classification-convnext_segmented.ipynb
jupyter lab DeepLearning_classification-convnextl_segmented.ipynb
jupyter lab DeepLearning_classification-effv2s_segmented.ipynb

# Final ensemble creation
jupyter lab DeepLearning_classification-ensemble_weighted_segmented-final.
    ipynb

# Comprehensive model comparison
python fossil_model_compare.py --results_root ../3_Results --output_dir ../3
    _Results/_comparison
```

Listing 2: Model Training Commands

6.3 Dashboard Deployment

```
cd 4_Dashboard_App

# Launch interactive dashboard
streamlit run Home.py --server.maxUploadSize 9000 --server.maxMessageSize
    10000

# Access at: http://localhost:8501
```

Listing 3: Dashboard Launch Commands

6.4 Environment Requirements

Recommended Docker Environment:

```
docker run --gpus all --ipc=host --ulimit memlock=-1 --ulimit stack=67108864
     \
  --rm -p 10000:8888 -p 8501:8501 -v ${PWD}:/workspace/mycode \
  abdelghafour1/ngc_tf_rapids_25_01_vscode_torch:2025-v3 \
  jupyter lab --ip=0.0.0.0 --allow-root
```

Listing 4: Docker Environment Setup

Minimum System Requirements:

- OS: Linux, Windows, or macOS
- Python: 3.8+
- RAM: 16GB (32GB recommended)
- GPU: 8GB VRAM (16GB+ for large models)
- Storage: 100GB free space

7 Conclusion and Future Work

This work establishes a comprehensive framework for AI-assisted fossil classification using 2D slices from 3D models, achieving 95.64% accuracy. Key contributions include: (1) a scientifically rigorous dataset creation methodology preventing data leakage through model-level splitting, (2) systematic evaluation of seven state-of-the-art 2D CNN architectures, (3) an adaptive ensemble method, and (4) an integrated dashboard for real-time analysis.

The exceptional performance demonstrates the viability of this automated identification approach. However, challenging species like Baculogypsina indicate opportunities for future work. Adhering to methodological best practices, such as the strict use of specimen-level data splitting, is paramount for ensuring the scientific integrity of computational paleontology [4].

Future Research Directions:

- **3D CNN Integration**: Exploring true 3D or 2.5D CNNs for direct processing of volumetric data may improve performance on morphologically complex species.

- **Few-Shot Learning**: Developing techniques for species with limited training data.
- **Multi-Modal Fusion**: Combining 2D slice texture with 3D geometric features.
- **Dataset Expansion**: Incorporating additional fossil types and preservation states.

This framework provides a foundation for advancing AI applications in paleontology while maintaining the scientific rigor essential for research validity.

8 References

References

[1] Brayard, A., et al. "How to investigate fossil morphology and ontogeny: advantages and limitations of micro-CT scanning for paleontological studies." Palaeogeography, Palaeoclimatology, Palaeoecology 418 (2015): 163-175.

[2] Deng, J., et al. "ImageNet: A large-scale hierarchical image database." 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009.

[3] Pan, S. J., & Yang, Q. "A survey on transfer learning." IEEE Transactions on Knowledge and Data Engineering 22.10 (2010): 1345-1359.

[4] Willemink, M. J., et al. "Preparing medical imaging data for machine learning." Radiology 295.1 (2020): 4-15.

[5] He, K., et al. "Deep residual learning for image recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.

[6] Tan, M., & Le, Q. V. "EfficientNetV2: Smaller Models and Faster Training." In International Conference on Machine Learning (ICML) (2021).

[7] Liu, Z., et al. "A ConvNet for the 2020s." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022).

[8] Zoph, B., et al. "Learning Transferable Architectures for Scalable Image Recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018).

[9] Howard, A., et al. "Searching for MobileNetV3." In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019).

[10] Yun, S., et al. "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features." In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019).

[11] Zhang, H., et al. "mixup: Beyond Empirical Risk Minimization." arXiv preprint arXiv:1710.09412 (2017).

[12] Zhou, Z. H. "Ensemble Methods: Foundations and Algorithms." CRC Press, 2012.

[13] Wang, Z., et al. "Image quality assessment: from error visibility to structural similarity." IEEE Transactions on Image Processing 13.4 (2004): 600-612.

[14] Dice, L. R. "Measures of the amount of ecologic association between species." Ecology 26.3 (1945): 297-302.

[15] Streamlit Inc. "Streamlit: Data apps for the modern era." (2020). [Online]. Available: streamlit.io.

[16] Ferreira, T. A., et al. (2023). "Deep learning workflow to generate high-resolution labeled datasets for micropaleontology." arXiv preprint arXiv:2304.04291.

[17] Hou, C., et al. (2023). "Fossil Image Identification using Deep Learning Ensembles of Data Augmented Multiviews." Methods in Ecology and Evolution.

[18] Shiri, I., et al. (2021). "Effect of data leakage in brain MRI classification using 2D convolutional neural networks." Scientific Reports.

[19] Lewis, J. P. (2001). "Fast template matching." In Vision interface (Vol. 95, pp. 120-123).

[20] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). "ORB: An efficient alternative to SIFT or SURF." In 2011 International Conference on Computer Vision (ICCV).

[21] Otsu, N. (1979). "A threshold selection method from gray-level histograms." IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66.

[22] Loshchilov, I., & Hutter, F. (2017). "Decoupled Weight Decay Regularization." arXiv preprint arXiv:1711.05101.

[23] Szegedy, C., et al. (2016). "Rethinking the Inception Architecture for Computer Vision." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[24] Reid, M., et al. (2023). "High-throughput X-ray micro-computed tomography and deep learning segmentation for digital excavation of invertebrate fossils." Frontiers in Ecology and Evolution.

[25] Mietchen, D., et al. (2024). "Accelerating segmentation of fossil CT scans through Deep Learning." Scientific Reports.

A Dataset Balance Metrics

Table 3: Dataset balance metrics across train/validation/test splits

Split	Total Images	Coefficient of variation (CV) (%)	Balance Quality
Training	44 103	13.91	Excellent
Validation	14 046	20.58	Good
Test	51 468	24.72	Acceptable

B Per-Species Dataset Distribution

Table 4: Complete species distribution across dataset splits

Species	Models	Total Images	Train	Val	Test
Chrysalidina	16	9972	4005	1200	4767
Ataxophragmium	7	9834	3954	1200	4680
Baculogypsina	7	6220	2499	1200	2521
Minoxia	6	9624	3866	1200	4558
Elphidiella	6	9478	3808	1200	4470
Fallotia	6	9458	3800	1200	4458
Arumella	5	8848	3556	1200	4092
Lockhartia	5	8822	3545	1200	4077
Orbitoides	5	8777	3525	1200	4052
Rhapydionina	5	8765	3519	1200	4046
Alveolina	4	10 004	4018	1200	4786
Coskinolina	4	9835	3952	1200	4683

C Complete Model Performance Matrix

Table 5: Comprehensive performance comparison across all evaluated models (metrics in %)

Model	Accuracy	Precision	Recall	F1-Score	Top-3 Acc	AUC
Ensemble (Final)	95.64	95.94	95.64	95.43	99.60	0.998
ConvNeXt-Large	95.12	96.15	95.12	94.06	99.63	0.998
NASNet	93.69	94.87	93.69	92.54	99.29	0.997
EfficientNetV2-Large	93.53	94.68	93.53	91.98	99.25	0.997
EfficientNetV2-Small	91.82	93.12	91.82	90.45	98.96	0.996
ConvNeXt-Base	90.28	91.85	90.28	89.12	98.72	0.995
ResNet101V2	89.45	90.98	89.45	88.34	98.54	0.994
MobileNet	88.02	89.76	88.02	86.89	98.15	0.993

D Species-Level Performance Analysis (Final Ensemble)

Table 6: Detailed per-species performance metrics for the final ensemble model (metrics in %)

Species	Precision	Recall	F1-Score	Support
Fallotia	99.71	99.60	99.66	4458
Ataxophragmium	99.23	99.25	99.24	4680
Arumella	98.58	98.82	98.70	4092
Rhapydionina	98.17	98.42	98.29	4046
Alveolina	97.55	97.11	97.33	4786
Chrysalidina	95.96	95.60	95.78	4767
Elphidiella	95.34	95.12	95.23	4470
Minoxia	94.78	94.89	94.83	4558
Coskinolina	93.45	93.78	93.61	4683
Lockhartia	92.87	92.34	92.60	4077
Orbitoides	87.45	86.89	87.17	4052
Baculogypsina	75.89	74.56	75.22	2521

E Glossary of Key Methodologies

E.1 Decoupled Weight Decay (AdamW)

Decoupled weight decay, implemented in the AdamW optimizer, addresses a flaw in how traditional L2 regularization is handled in adaptive optimizers like Adam. Instead of including the L2 regularization term in the gradient calculation, AdamW decouples it and applies the weight decay directly to the parameters during the update step. The update rule is defined as:

$$\theta_{t+1} = \theta_t - \gamma(\lambda\theta_t + \Delta\theta_t) \tag{1}$$

where θ_t are the parameters at timestep t, γ is the learning rate schedule, λ is the weight decay rate, and $\Delta\theta_t$ is the update term from the Adam optimizer. This modification has been shown to significantly improve the generalization performance of the Adam optimizer.

E.2 Label Smoothing

Label smoothing is a regularization technique that replaces "hard" one-hot encoded target labels with "soft" labels to prevent the model from becoming overconfident. A small probability mass, ϵ, is taken from the correct class and distributed uniformly across all other classes. For a classification problem with K classes, the new "soft" target label y'_k for a given class k is calculated as:

$$y'_k = y_k(1 - \epsilon) + \frac{\epsilon}{K} \tag{2}$$

where y_k is the original "hard" one-hot label (1 for the correct class, 0 otherwise). This encourages smaller logit differences and improves model calibration and generalization.

E.3 Mixup

Mixup is a data augmentation method that trains a network on convex combinations of pairs of training examples and their labels. A new virtual training sample $(\tilde{x}, \tilde{y})$ is generated from two randomly chosen samples (x_i, y_i) and (x_j, y_j) using the following rules:

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j \tag{3}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda)y_j \tag{4}$$

where the mixing coefficient λ is sampled from a Beta distribution, $\lambda \sim \text{Beta}(\alpha, \alpha)$. This encourages the model to learn simpler, linear behavior between training examples, improving generalization.

E.4 CutMix

CutMix is a data augmentation strategy where a random rectangular patch is cut from one training image and pasted over a corresponding region in another. The label for the new synthetic sample is then mixed proportionally to the area of the patch. The new label $\tilde{y}$ is defined as:

$$\tilde{y} = \lambda y_A + (1 - \lambda)y_B \tag{5}$$

where y_A and y_B are the original labels and λ is the area ratio of the remaining part of image A. This forces the model to recognize objects from partial views and improves both classification and localization ability.

E.5 Dice Similarity Coefficient (DSC)

The Dice Similarity Coefficient (DSC) is a standard metric for evaluating the performance of segmentation algorithms by quantifying the spatial overlap between the predicted segmentation (X) and the ground-truth segmentation (Y). It is calculated as twice the area of the intersection divided by the sum of the areas of both sets:

$$\text{DSC} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{6}$$

The coefficient ranges from 0 (no overlap) to 1 (perfect match) and is highly sensitive to segmentation quality in cases of severe class imbalance.

E.6 Structural Similarity Index (SSIM)

The Structural Similarity Index (SSIM) is a perceptual metric for measuring the similarity between two images. Proposed by Wang et al., SSIM was a significant departure from traditional metrics like Mean Squared Error (MSE) which measure absolute pixel-wise errors [13]. The SSIM framework is motivated by the hypothesis that the human visual system is highly adapted for extracting structural information from a scene. Therefore, a measure of perceived image quality should be based on the degradation of this structural information. The SSIM index compares two images based on three distinct components: luminance, contrast, and structure, providing a score that aligns much more closely with human perception of image quality than simple error metrics.

E.7 Otsu's Method

This is a classic and widely used algorithm for automatic image thresholding. Given a grayscale image, Otsu's method automatically determines an optimal intensity threshold to separate the pixels into two classes, foreground and background [21]. It does this by exhaustively searching for the threshold that minimizes the intra-class intensity variance or, equivalently, maximizes the inter-class variance of the two resulting groups of pixels. It is particularly effective for images with bimodal histograms (i.e., clear peaks for foreground and background).

E.8 Normalized Cross-Correlation (NCC)

NCC is a robust technique for template matching, which involves finding the location of a smaller template image within a larger source image. It measures the similarity between the template and patches of the source image by calculating their cross-correlation and normalizing the result to lie between -1 and 1 [19]. This normalization makes the NCC metric invariant to linear changes in brightness and contrast, making it much more robust than simple sum-of-squared-differences for real-world applications.

E.9 ORB (Oriented FAST and Rotated BRIEF)

ORB is a feature detection and description algorithm developed as a fast and efficient alternative to methods like SIFT (Scale-Invariant Feature Transform). It combines the FAST (Features from Accelerated Segment Test) keypoint detector for quickly identifying interest points with a modified version of the BRIEF (Binary Robust Independent Elementary Features) descriptor [20]. ORB adds rotation invariance to the BRIEF descriptor by "steering" it according to the orientation of the keypoint. As a binary descriptor, ORB is extremely fast to compute and match using Hamming distance, and it is also free from patents, which contributed to its widespread adoption.