Bird Classification Project - BDMA 07 Competition 2026

Olha BALIASINA
CentraleSupélec
3 Rue Joliot Curie, 91190 Gif-sur-Yvette
olha.baliasina@student-cs.fr

Nima KAMALI LASSEM
CentraleSupélec
3 Rue Joliot Curie, 91190 Gif-sur-Yvette
nima.kamalilassem@student-cs.fr

Abstract

We present a systematic approach to fine-grained bird species classification on a 20-class subset of the Caltech-UCSD Birds-200-2011 dataset, developed for a Kaggle competition. Starting from simple baselines (ResNet-50 with frozen backbone, 55% test accuracy), we progressively build a pipeline that achieves 93% accuracy on the public leaderboard through four key contributions: (1) domain-specific pretraining via an EVA-02 Large Vision Transformer backbone fine-tuned on iNaturalist-2021, which provides the single largest accuracy gain (+4% over a ConvNeXt-Base baseline with identical pipeline); (2) a two-stage object detection pipeline combining YOLOv8m with Grounding DINO fallback, achieving over 99% bird localization rate; (3) multi-view inference with confidence-gated blending of full-image and cropped predictions, combined with 5-fold cross-validation and test-time augmentation; and (4) specialist binary classifiers targeting three taxonomically confused species pairs (American/Fish Crow, Rusty/Brewer Blackbird, Yellow-billed/Black-billed Cuckoo) that account for over 80% of classification errors. We also document negative results from Stochastic Weight Averaging and model soup experiments, which failed to improve over our best configuration. Our approach isolates the contribution of each component, providing insights into what matters most for fine-grained classification on small datasets.

1. Introduction

Fine-grained visual classification (FGVC) presents a fundamentally different challenge than general image recognition: inter-class differences are subtle and often localised to small regions, whereas intra-class variation can be substantial (due to pose, lighting, object properties, etc.). Bird species classification is a canonical FGVC benchmark, where species within the same taxonomic family may differ only in plumage details, bill shape, or body proportions that occupy a fraction of the image.

This work addresses a 20-class bird classification task using a subset of the Caltech-UCSD Birds-200-2011 (CUB-200) dataset [18], evaluated through a Kaggle competition. The training set contains approximately 1,200 labeled images (∼60 per class), making this a challenging low-data task where overfitting is a primary concern.

Our approach follows a principled experimental methodology: we begin with standard baselines to establish performance floors, systematically analyze error patterns, and introduce targeted techniques to address each identified bottleneck. As the result of this iterative process we have obtained several key findings:

1. **Pretraining domain matters.** Switching from an ImageNet-22k pretrained ConvNeXt-Base [13] to an iNaturalist-21k [17] pretrained EVA-02 Large [5] within the same training pipeline shows a +4% improvement, the single largest gain in our progression. The iNaturalist backbone already encodes species-discriminative features, requiring only minimal fine-tuning.
2. **Errors concentrate in taxonomic pairs.** After applying domain-specific pretraining, 12 of 20 classes reach 100% out-of-fold accuracy. Residual errors are tightly concentrated in three within-family species pairs, with the American Crow/Fish Crow pair alone accounting for 56% of all misclassifications.
3. **Specialist classifiers make the difference.** Dedicated binary classifiers for the three most confused pairs, integrated via confidence-gated blending, push the final score from 0.925 to 0.930.
4. **Weight averaging provides limited benefit on small datasets.** Both Stochastic Weight Averaging (SWA) [8] and model soup [20] fail to improve over our best single-training run. However, we believe that the model soup approach might be impactful for the rest 50% of test data that will be used for the final evaluation.

Figure 1 provides an overview of the complete pipeline. The remainder of this paper details the dataset analysis (Sec. 3), methodology (Sec. 4), experimental results (Sec. 5), failed experiments (Sec. 6), and conclusions (Sec. 7).

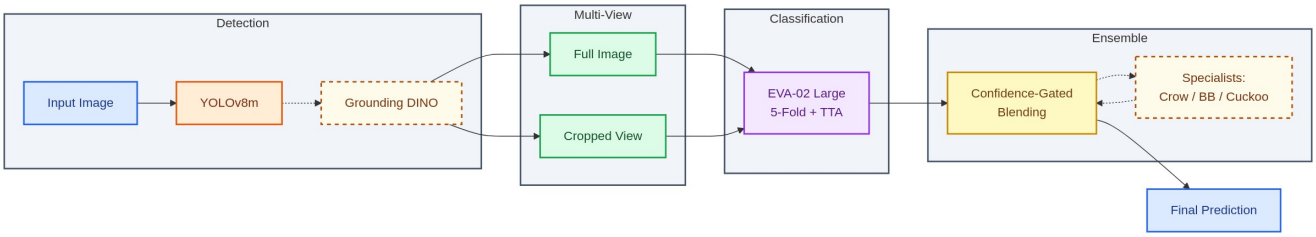

Figure 1. Overview of the proposed pipeline. Each test image passes through a two-stage detection module (YOLOv8m with Grounding DINO fallback), then is classified by a 5-fold ensemble of EVA-02 Large models using multi-view inference and test-time augmentation. For images predicted as one of the three confused species pairs, specialist binary classifiers provide refined predictions via confidence-gated blending.

2. Related work

Fine-grained visual classification. Fine-grained classification on bird datasets has been extensively studied since the introduction of CUB-200-2011 [18]. Early approaches relied on part-based models that localize discriminative regions (head, breast, wings) before classification [23]. More recently, Vision Transformers (ViTs) [4] have achieved strong results by leveraging self-attention to implicitly discover discriminative regions without explicit part annotations. TransFG [6] and similar methods demonstrate that ViT attention maps naturally focus on species-specific features.

Transfer learning and domain-specific pretraining. Transfer learning from ImageNet [2] is standard practice, but the pretraining domain matters significantly for downstream performance. The iNaturalist dataset [17] contains over 2.7 million images spanning 10,000 species, making it an ideal source of domain-aligned features for biological classification. Models such as EVA-02 [5]—which combines CLIP [16] pretraining on 2 billion image-text pairs with fine-tuning on iNaturalist-2021—encode rich species-discriminative representations that transfer effectively to downstream tasks. Our work confirms that this domain alignment is the single most impactful factor for fine-grained bird classification.

Object detection as preprocessing. Localizing the subject of interest before classification reduces background noise and improves fine-grained accuracy [23]. Modern detectors such as YOLOv8 [9] provide real-time bounding box predictions, while zero-shot open-vocabulary models like Grounding DINO [12] can detect objects from text prompts. We combine both in a two-stage cascade for robust localization.

Ensemble and weight averaging methods. Cross-validation ensembles and test-time augmentation (TTA) are widely used to improve robustness. Model soup [20] averages weights from multiple fine-tuned models into a single model, achieving improved accuracy without increased inference cost. Stochastic Weight Averaging (SWA) [8] averages weights along the training trajectory to find flatter loss minima. We evaluate both techniques and find neither provides gains in our low-data regime.

Table 1. Dataset statistics. The 20 classes are roughly balanced, with slight variation in sample count.

Statistic	Value
Number of classes	20
Total training+val images	1,212
Mean images per class	∼60
Min / Max per class	56 / 87
Test images	400
Image resolution (median)	∼350×450 px

3. Dataset analysis

Before designing our pipeline, we conduct a thorough exploratory data analysis (EDA) to understand the dataset characteristics and anticipate challenges.

3.1. Dataset overview

The dataset comprises 20 bird species from the CUB-200-2011 collection [18]. The combined training and validation sets contain 1,212 images with a roughly balanced distribution (56–87 images per class). The test set contains 400 images. Kaggle evaluates on 50% of the test set during the competition (public leaderboard), with the remaining 50% revealed at competition end (private leaderboard).

3.2. Taxonomic structure and confused pairs

A key insight from our analysis is that the 20 species are not taxonomically independent. They form natural family groups where within-family pairs pose the greatest challenge:

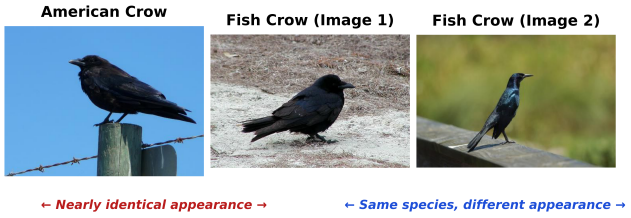

Figure 2. The challenge of fine-grained bird classification illustrated through the American Crow / Fish Crow pair. **Left and center:** Two different species (American Crow and Fish Crow) show nearly identical appearance—similar size, plumage, and posture—making inter-species discrimination difficult. **Center and right:** Two images of the same species (Fish Crow) show dramatically different appearances due to pose, lighting, and background variation. This paradox—where different species look alike while the same species can look different—is the core difficulty of fine-grained visual classification and explains why this pair alone accounts for 56% of all classification errors.

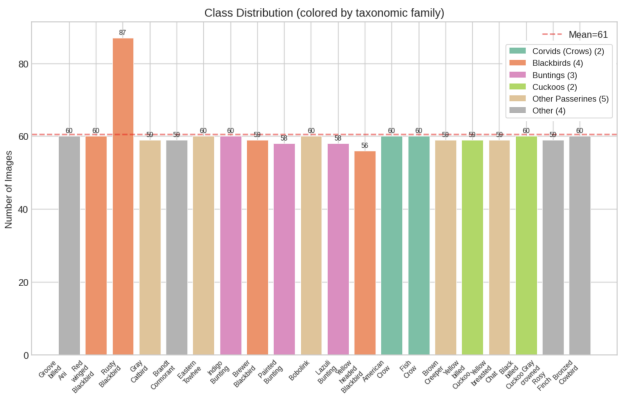

Figure 3. Training+validation set class distribution. As we are limited in the amount of labeled data, after trying initial baseline approaches, we have combined all the labeled images and switched to the stratified K-fold cross-validation. The obtained dataset is roughly balanced, with slight variation (56–87 images per class).

- **Corvids:** American Crow, Fish Crow
- **Icterids:** Red-winged Blackbird, Rusty Blackbird, Brewer Blackbird, Yellow-headed Blackbird, Bobolink, Bronzed Cowbird
- **Cardinalids:** Indigo Bunting, Lazuli Bunting, Painted Bunting
- **Cuculids:** Yellow-billed Cuckoo, Black-billed Cuckoo, Groove-billed Ani

Species between different families are visually distinct and easy to classify (as confirmed by our baselines). The difficulty lies entirely within families. As we show in Sec. 5.5, four pairs account for over 90% of all misclassifications after applying our best backbone.

3.3. Image characteristics

Images vary considerably in resolution (100–500 px on the shorter side), bird-to-frame area ratio, and background complexity. Using YOLOv8m bird detection, we find that the median bird occupies approximately 15–25% of the frame, with a long tail of images where the bird is very small (<5% of the frame). These small-bird images are disproportionately difficult, as the model must classify from limited visual information.

4. Method

Our pipeline consists of four stages: (1) object detection and cropping, (2) feature extraction with a domain-specific backbone, (3) 5-fold ensemble with multi-view inference and TTA, and (4) specialist binary classifiers for confused species pairs. We describe each component below.

4.1. Two-stage object detection

Raw images often contain significant background clutter (branches, water, sky) that can distract the classifier. We introduce a two-stage detection pipeline to localize birds before classification.

Stage 1: YOLOv8m. We use a pretrained YOLOv8m [9] model (COCO class 14 = "bird") with a confidence threshold of 0.25 and 15% bounding box padding on each side. When multiple bird detections are present, we select the largest by area. This handles approximately 97% of images.

Stage 2: Grounding DINO fallback. For images where YOLOv8m does not detect a bird, we apply Grounding DINO [12] with the text prompts like `"a bird"` and `"a bird perched in a tree"` and a box threshold of 0.25. This zero-shot open-vocabulary detector recovers most remaining cases, bringing total detection coverage above 99%. The ∼1% of undetected images (heavily occluded or very distant birds) fall back to center-crop processing.

All bounding boxes are precomputed and cached before training to avoid runtime overhead during the training loop.

4.2. Multi-view dataset

During training, each image is presented in one of two views, selected stochastically:

- **Full image**: The entire image with standard augmentations applied.
- **Cropped view**: The detected bird region, cropped and resized to the target resolution.

The probability of selecting the cropped view follows a curriculum schedule:

$$p_{\text{crop}}(e) = p_{\text{start}} + (p_{\text{end}} - p_{\text{start}}) \cdot \frac{e}{E} \quad (1)$$

where e is the current epoch, E is the total number of epochs, and we set $p_{\text{start}} = 0.3$, $p_{\text{end}} = 0.5$. This ensures that early in training the model sees more full images (learning contextual cues such as habitat), while later it increasingly trains on crops (focusing on fine-grained details).

4.3. Backbone: EVA-02 Large with iNaturalist pretraining

We use EVA-02 Large [5], a Vision Transformer with 304M parameters. This model was pretrained on CLIP's merged2B dataset (2 billion image-text pairs) and subsequently fine-tuned on iNaturalist-2021 (2.7M images, 10k species). The native input resolution is 336×336 pixels.

Architecture. The EVA-02 backbone extracts 1024-dimensional patch token features, which we aggregate via mean pooling over all spatial tokens. On top of this, we attach a classification head consisting of dropout ($p = 0.3$) followed by a linear layer mapping to 20 classes. For comparison, we also evaluate two simpler baselines: ResNet-50 [7] (25M parameters, ImageNet-1k pretrained) and ConvNeXt-Base [13] (89M parameters, ImageNet-22k pretrained).

Generalized mean pooling (GeM). For CNN-based backbone ConvNeXt-Base, we replace global average pooling with Generalized Mean (GeM) pooling [15], which is a stronger pooling for fine-grained recognition:

$$\mathbf{f} = \left[\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} x_s^p \right]^{1/p} \tag{2}$$

where p is a learnable parameter initialized to 3.0. GeM with $p > 1$ emphasizes high activations, effectively focusing on the most discriminative spatial regions. For the ViT backbone (EVA-02), we use simple mean pooling over patch tokens, as ViTs do not produce spatial feature maps in the same way as CNNs.

Why iNaturalist pretraining matters. ImageNet pretraining teaches general visual features (edges, textures, object shapes), while iNaturalist pretraining teaches species-discriminative features (plumage patterns, bill morphology, body proportions). For a task requiring distinction between visually similar bird species, this domain alignment is critical. Our experiments (Sec. 5.3) confirm that the pretraining domain is the single most impactful factor: switching from ConvNeXt-Base (IN-22k) to EVA-02 (iNat-21k) within the same pipeline improves Kaggle accuracy from 0.87 to 0.91 (+4%), and with addition of bird detection, cross-validation and TTA—to 0.925.

4.4. Training procedure

Differential learning rates. We apply separate learning rates to the pretrained backbone (3×10^{-6}) and the randomly initialized classification head (5×10^{-4}), a ratio of $\sim 167\times$. This is critical because the iNaturalist-pretrained backbone already contains near-optimal features and requires only fine adjustment, while the head must learn the 20-class decision boundary from scratch. Using a single learning rate risks either under-training the head (if too low) or catastrophically forgetting backbone features (if too high).

Learning rate schedule. We perform a two-phase schedule: (1) a linear warmup over 2 epochs (starting from $0.1\times$ the target learning rate), followed by (2) cosine annealing to 10^{-7}. The warmup phase is essential to prevent large initial gradient updates from destroying the pretrained features—a phenomenon known as catastrophic forgetting [10].

Regularization. Given the high capacity of EVA-02 Large (304M parameters) relative to our small training set ($\sim$1,200 images), aggressive regularization is essential:

- **Label smoothing** ($\epsilon = 0.1$): Prevents overconfident predictions and improves calibration. Note that label smoothing artificially suppresses training accuracy (appearing as $\sim$91–93%) while validation accuracy reaches $\sim$97%, which is expected behavior.
- **Dropout** ($p = 0.3$): Applied before the final classification layer.
- **Weight decay** (0.05): Higher than the typical 0.01 for CNNs, following standard practice for ViT fine-tuning [4].
- **Mixup** [22] ($\alpha = 0.3$, $p = 0.2$): Linearly interpolates between two training images and their labels, encouraging smoother decision boundaries.
- **CutMix** [21] ($\alpha = 1.0$, $p = 0.2$): Replaces a rectangular patch of one image with a patch from another, along with proportional label mixing. Complements Mixup by providing localized augmentation.

Data augmentation. Training augmentations include random resized crop (scale 0.75–1.0), horizontal flip ($p = 0.5$), shift-scale-rotate, CLAHE, color jitter (brightness, contrast, saturation, hue), Gaussian blur, Gaussian noise, and CoarseDropout (equivalent to CutOut [3]). All augmentations are implemented via the Albumentations library [1]. For validation, we apply only resize and center crop with ImageNet normalization.

5-fold stratified cross-validation. We train on the combined train+validation set using stratified 5-fold CV, ensuring each fold preserves the class distribution. Each fold

trains for a maximum of 20 epochs with early stopping (patience = 8 epochs based on validation accuracy). This provides us with 5 complementary models that have each seen 80% of the data for training.

4.5. Inference pipeline

At test time, we combine multiple sources of information:

Fold ensemble with accuracy weighting. The 5 fold models are ensembled with accuracy-cubed weights:

$$w_k = \frac{a_k^3}{\sum_{j=1}^{5} a_j^3} \tag{3}$$

where a_k is the validation accuracy of fold k. The cubic exponent assigns disproportionate influence to higher-accuracy folds, as small differences in fold accuracy can reflect meaningful differences in model quality.

Test-time augmentation (TTA). Each image is classified under 4 geometric transforms: (1) original, (2) horizontal flip, (3) scale $\times 1.1$, and (4) scale $\times 0.9$. Logits (not softmax probabilities) are averaged across transforms, as logit averaging produces better calibrated ensemble predictions [11].

Multi-view confidence-gated blending. For each test image with a detected bird bounding box, we compute separate predictions on both the full image and the bird crop. These are combined using a confidence-gated heuristic:

- If crop confidence > 0.85 and exceeds full-image confidence by > 0.08: use crop predictions exclusively (*crop_only*).
- If full-image confidence < 0.90: blend as $0.75 \cdot \text{full} + 0.25 \cdot \text{crop}$ (*multiview*).
- Otherwise: use full-image predictions exclusively (*full*).

This heuristic prevents overriding confident full-image predictions while providing the crop as a "second opinion" on uncertain cases. Crops smaller than 50×50 pixels are rejected to avoid classifying from insufficient visual information.

4.6. Specialist binary classifiers

Motivated by the error concentration in three species pairs (Sec. 5.5), we train dedicated binary classifiers—one for each pair:

1. **Crow specialist**: Fish Crow vs. American Crow
2. **Blackbird specialist**: Brewer Blackbird vs. Rusty Blackbird
3. **Cuckoo specialist**: Yellow-billed Cuckoo vs. Black-billed Cuckoo

Each specialist uses the same EVA-02 Large backbone (iNaturalist pretrained) with `global_pool='avg'`, dropout ($p = 0.2$), and a 2-class linear head. Training uses the same 5-fold CV and multi-view dataset as the main model. Crucially, we apply differential learning rates (backbone: 2×10^{-6}, head: 10^{-4}) to prevent overfitting on the small per-pair dataset ($\sim$120 images, $\sim$24 per validation fold).

Specialist override logic. At inference, a specialist is activated when the main model's prediction belongs to the paired classes, *or* when the combined probability mass on both classes exceeds 0.3. The specialist prediction is blended with the main model using a dynamic weight:

$$\alpha = \min(0.5,\ 0.2 + 0.3 \cdot c_{\text{spec}}) \tag{4}$$

where $c_{\text{spec}} = \max(\mathbf{p}_{\text{spec}})$ is the specialist's confidence. For the crow pair:

$$p_i' = (1 - \alpha) \cdot p_i + \alpha \cdot s_i, \quad i \in \{\text{Fish, American}\} \tag{5}$$

followed by renormalization over all 20 classes. This dynamic blending scales from cautious ($\alpha = 0.2$) when the specialist is uncertain to assertive ($\alpha = 0.5$) when it is confident.

5. Experiments

5.1. Experimental setup

All experiments are conducted on Google Colab using a single NVIDIA A100 GPU. Training uses mixed-precision (FP16) via PyTorch's `autocast` and `GradScaler` for memory efficiency. Models are implemented using the `timm` library [19] for backbone loading and Albumentations [1] for data augmentation. The optimizer is AdamW [14] throughout.

5.2. Baseline experiments

We establish three baselines using a simplified pipeline: no object detection, no cross-validation (using the provided with the dataset train/val split), no TTA, and standard training augmentations only.

ResNet-50 frozen backbone (0.550). Using ResNet-50 [7] with ImageNet-1k pretrained weights and a frozen backbone (only training the classification head) gives 55% accuracy—barely above the 5% random baseline for 20 classes, but illustrating that ImageNet features alone provide some discriminative signal. The head is trained with learning rate 10^{-3} for 30 epochs.

ResNet-50 fine-tuned (0.715). Unfreezing and fine-tuning the full backbone with learning rate 10^{-4} improves

Table 2. Baseline model comparison using a simple pipeline (no detection, no cross-validation, no TTA). The improvement from ResNet-50 frozen to fine-tuned demonstrates the importance of backbone adaptation. The jump to ConvNeXt-Base reflects richer IN-22k pretraining and higher resolution.

Model	Pretrain	Res.	Kaggle
ResNet-50 (frozen)	IN-1k	224	0.550
ResNet-50 (fine-tuned)	IN-1k	224	0.715
ConvNeXt-Base	IN-22k	384	0.870

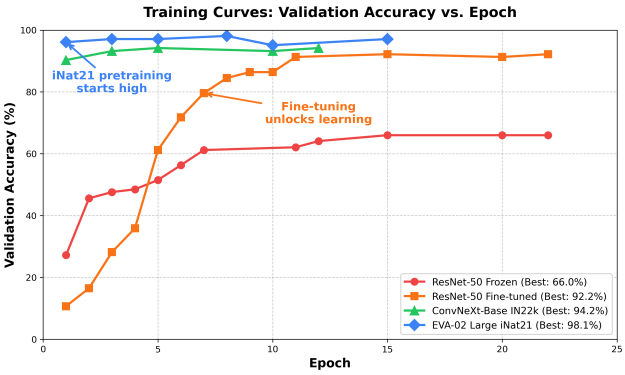

Figure 4. Training curves for baseline models. ResNet-50 frozen plateaus at 66%, while fine-tuning enables learning to 92.2%. ConvNeXt-Base with ImageNet-22k pretraining starts at 90% and saturates quickly. EVA-02 Large with iNaturalist-21k pretraining begins at 96% on epoch 1, demonstrating that domain-specific pretraining already encodes species-discriminative features.

accuracy by +16.5%, confirming that task-specific adaptation of the backbone is essential for fine-grained classification. The pretrained features are a starting point, not a sufficient representation.

ConvNeXt-Base (0.870). Switching to ConvNeXt-Base [13] pretrained on ImageNet-22k at 384×384 resolution provides another +15.5% improvement. This reflects two factors: (1) IN-22k pretraining provides richer features than IN-1k (21,000 vs. 1,000 classes), and (2) the higher input resolution (384 vs. 224) preserves more fine-grained detail.

5.3. Full pipeline: ablation study

Table 3 shows the progressive improvement as we add pipeline components. Each row adds exactly one component to isolate its contribution.

Impact of domain-specific pretraining. The switch from ConvNeXt-Base (IN-22k) to EVA-02 Large (iNat-21k) within the full pipeline yields the most important improvement in our progression, as all the previous experiments we have conducted with ConvNeXt (including performing CV, intensifying augmentations, building complex ensembles with 3 more architectures: Swin, EfficientNetV2, EVA) plateaued at 0.88 accuracy on Kaggle leaderboard. This is expected: iNaturalist-2021 contains images of 10,000 species including many birds, so the pretrained features are already "tuned" for the subtle morphological differences our task requires. The backbone converges rapidly (best fold accuracy typically reached by epoch 4–10), confirming that the pretrained features need minimal adaptation.

Table 3. Ablation study. Each row adds one component to the previous configuration. "CV Acc" is the mean 5-fold out-of-fold accuracy; "Kaggle" is the public leaderboard score. The largest single gain (+5.5%) comes from switching to iNaturalist pretraining.

Configuration	CV Acc	Kaggle	Δ
ConvNeXt-Base (simple pipeline)	94.17%	0.870	—
+ Detection + 5-fold CV + TTA	94.55%	0.865	-0.5%
EVA-02 iNat21 (simple pipeline)	98.06%	0.910	+4%
+ Detection + 5-fold CV + TTA	96.95%	0.925	+5.5%
+ Specialist classifiers	—	**0.930**	**+6%**

Table 4. Per-fold cross-validation results with EVA-02 iNat21. Low variance (±0.43%) across folds indicates stable training. Most folds reach peak accuracy early (epoch 4–10).

Fold	Best Val Acc	Best Epoch	Early Stop
1	97.53%	4	Ep 10
2	97.12%	5	Ep 11
3	96.69%	4	Ep 10
4	96.28%	10	Ep 16
5	97.11%	10	Ep 16
Mean	**96.95% ± 0.43%**		

5.4. Cross-validation analysis

The 5-fold CV with EVA-02 achieves 96.95% ± 0.43% out-of-fold (OOF) accuracy and 97.02% macro-F1. The low inter-fold variance (±0.43%) indicates stable training.

5.5. Error analysis

The OOF confusion matrix (Fig. 5) reveals that errors are tightly concentrated in three species pairs:

Table 6 reports per-class accuracy. Notably, 14 classes achieve *perfect* 100% OOF accuracy, and only 2 classes fall below 95% (both crow species at ∼82%).

5.6. Specialist classifier results

Table 7 reports specialist cross-validation accuracy. The crow specialist achieves 86.67%, better than the main

Table 5. Confusion analysis from 5-fold OOF predictions. The top three confused pairs account for 90% of all 36 misclassifications. The Crow pair alone causes 56% of errors.

Confused pair	Errors	% Total
American Crow ↔ Fish Crow	20	55.6%
Rusty Blackbird ↔ Brewer Blackbird	8	22.2%
Black-billed Cuckoo ↔ Yellow-billed Cuckoo	2	5.6%
Other (scattered)	6	16.7%

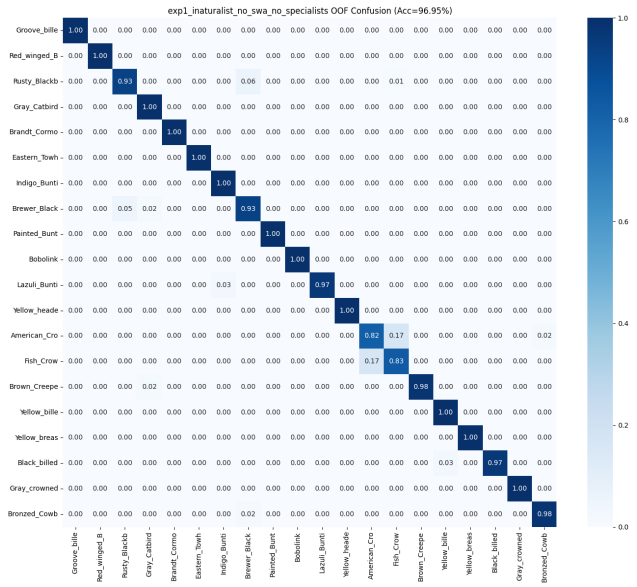

Figure 5. Normalized confusion matrix from 5-fold OOF predictions (EVA-02 iNat21, before specialists). The matrix is nearly diagonal: 12 of 20 classes achieve 100% accuracy. Visible off-diagonal mass is concentrated in the American Crow/Fish Crow pair (82–83% accuracy each, with 17% mutual confusion) and the Rusty Blackbird/Brewer's Blackbird pair.

Table 6. Per-class OOF accuracy for EVA-02 iNat21 (before specialists). Only classes below 100% are shown; 14 classes achieve perfect accuracy.

Class	OOF Acc	Errors / Total
American Crow	81.7%	11 / 60
Fish Crow	83.3%	10 / 60
Rusty Blackbird	93.1%	6 / 87
Brewer Blackbird	93.2%	4 / 59
Lazuli Bunting	96.6%	2 / 58
Black-billed Cuckoo	96.7%	2 / 60
Brown Creeper	98.3%	1 / 59
Bronzed Cowbird	98.3%	1 / 60
14 other classes	100%	0

model's ∼82% on crows but with high variance reflecting the tiny validation sets (24 images per fold).

Table 7. Specialist classifier 5-fold CV accuracy. High variance is expected given only ∼24 samples per validation fold.

Specialist	Mean Acc	Std	Best Fold
Crow	86.67%	±6.12%	91.67%
Blackbird	93.15%	±2.18%	96.55%
Cuckoo	98.33%	±2.04%	100.00%

Table 8. Distribution of inference methods across the 400 test images. The specialist classifiers intervene on confused pairs, while remaining images use multi-view blending or full-image inference.

Method	Count	Percentage
Full image only	123	30.8%
Multi-view blend	142	35.5%
Crop only	18	4.5%
Crow specialist	38	9.5%
Blackbird specialist	40	10.0%
Cuckoo specialist	39	9.8%
Mean confidence	0.801	
Images with conf. < 0.6	12.8%	
Predictions changed by specialists	3 (crow only)	

Figure 6. Examples of low-confidence test predictions (<0.6). These are predominantly crow and blackbird images where the model is uncertain. Green bounding boxes show YOLO detections. Note that some birds are very small relative to the frame, explaining the difficulty.

Despite modest individual accuracy, the specialists improve the overall test score from 0.925 to 0.930 when integrated via the confidence-gated blending described in Sec. 4.6. The improvement is small but consistent, confirming that targeted intervention on error-concentrated pairs provides value even with imperfect specialist models.

5.7. Multi-view inference analysis

Table 8 shows how the confidence-gated multi-view strategy distributes test predictions across the three inference modes.

Table 9. Complete experimental progression. Each row represents a distinct Kaggle submission. The progression demonstrates diminishing returns: the first architectural decisions (backbone, pretraining) provide the largest gains, while later refinements yield incremental improvements.

#	Configuration	Kaggle
1	ResNet-50 frozen, simple pipeline	0.550
2	ResNet-50 fine-tuned, simple pipeline	0.715
3	ConvNeXt-Base, simple pipeline	0.870
4	ConvNeXt-Base, full pipeline (det.+CV+TTA)	0.865
5	EVA-02 iNat21, full pipeline	0.925
6	EVA-02 iNat21, full pipeline + specialists	**0.930**

5.8. Final results summary

Table 9 summarizes the complete experimental progression from the simplest baseline to the final system.

6. Negative results and failed experiments

Documenting what *did not* work is as informative as reporting successes. Out of all the unsuccessful experiments we have conducted, we would like to focus two techniques that were expected to improve accuracy but ultimately failed to exceed our best result.

6.1. Stochastic Weight Averaging (SWA)

SWA [8] averages model weights from later training epochs to find broader optima that generalize better. We implemented SWA using PyTorch's `torch.optim.swa_utils`, averaging weights from epoch 15 onward with a fixed SWA learning rate.

Result. The SWA submission scored **0.925**, identical to the non-SWA baseline.

Analysis. We hypothesize three reasons for the lack of improvement:

1. **Insufficient averaging window.** With only 20 total epochs and early stopping typically triggering around epoch 10–12, there are at most 5–7 checkpoints to average. SWA benefits from longer training runs where more diverse weight snapshots are available.
2. **Learning rate already near zero.** By epoch 15, the cosine-annealed learning rate is approximately 10^{-7}, producing weight updates so small that consecutive checkpoints are nearly identical. This eliminates the weight diversity that SWA exploits.
3. **Pre-converged backbone.** The iNaturalist-pretrained backbone already sits near a good optimum; the fine-tuning trajectory explores a narrow region of weight space, limiting the potential for SWA to find a meaningfully different average.

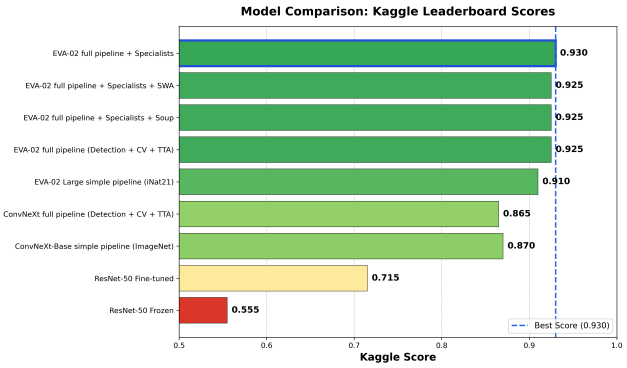

Figure 7. Kaggle score progression across experiments. Starting from a frozen ResNet-50 baseline (0.555), scores improve through fine-tuning (0.715), adopting ConvNeXt (0.870), and switching to the full detection + cross-validation + TTA pipeline. The largest single gain (+4.5%) comes from switching to iNaturalist-pretrained EVA-02 Large. Specialist classifiers provide the final boost to 0.930, while SWA and model soup fail to improve over the specialists-only baseline.

6.2. Model soup

Model soup [20] averages the weights of independently trained models into a single model, maintaining the same inference cost while potentially benefiting from the diversity of different training runs. We attempted to "soup" checkpoints from 25 training runs (5 folds $\times$ 5 restarts with minor hyperparameter variations) using greedy selection: starting from the best single model, iteratively adding models to the average only if validation loss improves.

Result. The best soup scored **0.925**, matching but not exceeding the best single-run submission.

Analysis. The soup procedure successfully recovered to the expected performance level but could not exceed it. Further investigation revealed that:

- Small hyperparameter changes between runs (e.g., minimum crop threshold: 50px $\rightarrow$ 30px) increased fold variance from $\pm$0.43% to $\pm$1.24%, suggesting training is sensitive to these choices at this data scale.
- The original best submission was "luckier" on the specific test split—its fold ensemble happened to make the right calls on borderline test images. The soup averaged away this variance rather than improving upon it.
- With only $\sim$1,200 training images, the training runs do not explore sufficiently different regions of weight space for souping to find a better average than any individual run.

6.3. Lessons learned

Weight averaging has diminishing returns on small datasets. Both SWA and model soup assume that different weight configurations capture complementary information. On large datasets (ImageNet-scale), this is typically true because different training runs converge to distinct local minima. On small datasets like ours, the training landscape is simpler and the models converge to similar solutions, leaving little room for averaging to help.

The 5-fold ensemble already provides robust aggregation. Our 5-fold CV ensemble already averages over 5 models trained on different data splits. Adding SWA or souping on top provides marginal additional diversity compared to the fold-level diversity already captured.

Engineering effort vs. impact. The specialist classifiers, despite their simplicity, provided a concrete +0.5% improvement because they directly address identified errors. In contrast, SWA and model soup required significant engineering effort (multiple retraining runs, checkpoint management, greedy selection) for zero improvement. This highlights the value of error-driven optimization over general-purpose techniques.

7. Conclusion

We presented a systematic approach to fine-grained bird species classification, progressing from a 55% baseline to 93% test accuracy through principled experimentation and error-driven optimization. Our key findings are:

1. **Domain-specific pretraining is the most impactful single factor.** Switching from ImageNet-22k to iNaturalist-21k pretraining gave +5.5% accuracy—more than all other pipeline components combined. This suggests that for domain-specific FGVC tasks, selecting the right pretrained backbone should be the first priority.
2. **Error analysis enables targeted improvements.** By analyzing the OOF confusion matrix, we identified that 80% of errors were concentrated in just two species pairs, directly motivating the specialist classifier approach. Without this analysis, effort might have been spent on techniques that address already-solved classes.
3. **Simple, targeted interventions outperform general-purpose techniques.** The specialist classifiers (+0.5%) provided concrete improvement despite their simplicity, while theoretically motivated techniques like SWA and model soup yielded zero improvement. On small datasets, the gap between individual training runs is often larger than the potential gain from weight averaging.
4. **Diminishing returns characterize the optimization trajectory.** The first architectural decisions (backbone choice, pretraining domain) provided the largest gains, while later refinements (multi-view blending, specialist tuning) led to incremental improvements at much higher engineering cost. This pattern is typical of competition settings and suggests that most practical value comes from foundational choices rather than elaborate post-processing.

Limitations and future work. The specialist classifiers suffer from small-sample statistics (120 images per pair), leading to high CV variance. Future work could explore few-shot learning or metric learning approaches that leverage the full 20-class feature space rather than training separate models. Learning the multi-view blending thresholds through validation (rather than manual tuning) would also be more principled. Finally, self-supervised pretraining on unlabeled bird images—permitted under the competition rules—could further improve backbone features for the most difficult species pairs.

References

[1] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2):125, 2020. 4, 5

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 2

[3] Terrell DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with CutOut. *arXiv preprint arXiv:1708.04552*, 2017. 4

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 4

[5] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 1, 2, 4

[6] Ju He, Jie-Neng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, and Changhu Wang. TransFG: A transformer architecture for fine-grained recognition. In *AAAI Conference on Artificial Intelligence*, pages 852–860, 2022. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4, 5

[8] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights

leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 876–885, 2018. 1, 2, 8

[9] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8, 2023. 2, 3

[10] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Oriol Vinyals, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, pages 3521–3526, 2017. 4

[11] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5

[12] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2, 3

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. 1, 4, 6

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[15] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1655–1668, 2019. 4

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 2

[17] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8769–8778, 2018. 1, 2

[18] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 2

[19] Ross Wightman. Pytorch image models. *GitHub repository*, 2019. 5

[20] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, pages 23965–23998, 2022. 1, 2, 8

[21] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019. 4

[22] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations (ICLR)*, 2018. 4

[23] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based R-CNNs for fine-grained category detection. In *European Conference on Computer Vision (ECCV)*, pages 834–849, 2014. 2