Object Recognition for Bird Species Classification

Joel Anil Jose   Lucia Victoria Fernandez Sanchez
BDMA 7 - Machine Learning - Code
{joel.anil-jose, lucia-victoria.fernandez}@student-cs.fr

Abstract

Fine-grained visual classification of bird species presents significant challenges due to subtle inter-class variations and high intra-class diversity. This paper presents a comprehensive solution for the Caltech-UCSD Birds-200-2011 classification task. We explore a heterogeneous super-ensemble approach combining ResNet-101, EfficientNet-B0, and ResNet-50 architectures with progressive resizing (224×224 *to* 448×448 *pixels) and advanced regularization, including Label Smoothing and Mixup augmentation. While a 3-model ensemble achieved peak validation accuracy, our final competition results were achieved using a robust 2-model ensemble and Test Time Augmentation (TTA), yielding an 83.0% private leaderboard accuracy. Our methodology employs Cosine Annealing with Warmup for learning rate scheduling and AdamW optimization with decoupled weight decay.*

1. Introduction

Object recognition in computer vision has made remarkable progress with deep convolutional neural networks [1, 7]. However, fine-grained visual categorization remains challenging, particularly for bird species classification where discriminative features are subtle and localized (e.g., beak shape, plumage patterns, tail coloration). The Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset [8] represents one of the most widely studied benchmarks for this task, containing 11,788 images across 200 bird species with significant pose variation and cluttered backgrounds.

In this work, we address a subset of the CUB-200-2011 dataset containing 20 bird species, focusing on developing a robust classification system that generalizes well despite limited training data. Our key contributions are:

- A heterogeneous ensemble combining ResNet-101 [1] and EfficientNet-B0 [7] with complementary inductive biases
- Progressive resizing strategy from 224×224 to 448×448 pixels to capture fine-grained texture details
- Integration of Label Smoothing [6] and Mixup [9] for enhanced regularization
- Soft-voting ensemble with Test Time Augmentation achieving 83.00% validation accuracy on the Kaggle competition.

2. Related Work

2.1. Fine-Grained Visual Categorization (FGVC)

Unlike general object recognition (e.g., distinguishing a bird from a car), Fine-Grained Visual Categorization (FGVC) focuses on identifying sub-categories within a single parent class [8]. The CUB-200-2011 dataset is significantly more challenging than standard benchmarks like ImageNet due to the high degree of inter-class similarity; many species share near-identical plumage, differing only in subtle localized features like eye-ring color or primary feather length. Furthermore, high intra-class variance—caused by differences in bird age, sex, and seasonal molting—requires models to learn robust, pose-invariant features rather than simple color histograms.

2.2. Evolution of Architectures

The field has transitioned from deep residual networks (ResNet), which introduced skip connections to facilitate training of very deep layers [1], to more computationally efficient models. EfficientNet-B0 [7] utilizes compound scaling to balance depth, width, and resolution, allowing it to capture high-frequency texture details with fewer parameters than traditional ResNets. Modern convolutional designs, such as ConvNeXt [2], have further integrated transformer-inspired elements—like larger kernel sizes and layer normalization—to enhance the global receptive field while maintaining the inductive bias of convolutions.

2.3. Ensemble Learning and Competition Strategy

In Kaggle-style competitions, a single model often suffers from specific ”blind spots” based on its architecture’s inductive bias. Ensemble learning, particularly soft-voting, is a standard winning strategy as it averages the probability distributions of multiple models to reduce variance [9].

By combining a deep feature extractor (ResNet-101), a texture-specialist (EfficientNet-B0), and a structural stabilizer (ResNet-50), the ensemble can correct individual misclassifications through consensus, leading to more stable performance on unseen private test data.

3. Methodology

Our approach consists of four main components: (1) progressive resizing with staged training, (2) heterogeneous architecture selection, (3) advanced regularization techniques, and (4) ensemble inference with TTA.

3.1. Dataset Characteristics and Challenges

The subset utilized in this project comprises 20 taxonomically diverse species from the CUB-200-2011 dataset. Each class contains approximately 30 training images and 11-13 validation images, presenting a ”small-data” challenge that necessitates aggressive regularization.

The primary challenge identified during our exploratory data analysis was the ”Corvus Ambiguity.” Species such as the *American Crow* and *Fish Crow* are visually indistinguishable in static images without auditory cues or scale references. Similarly, blackbird species (e.g., *Brewer’s Blackbird* vs. *Rusty Blackbird*) exhibit subtle iridescence patterns that are easily obscured by lighting variations or low image resolution. This motivated our decision to prioritize 448×448 resolution training and 5-crop TTA to ensure the model could ”zoom in” on these diagnostic markers.

3.2. Progressive Resizing Strategy

Progressive resizing [5] is a training technique where models are first trained on lower-resolution images (224×224) and then fine-tuned on higher-resolution images (448×448). This approach provides several benefits:

- **Faster initial training:** Lower resolution enables larger batch sizes and faster iteration
- **Better feature learning:** The model learns coarse features before refining on fine details
- **Reduced overfitting:** Progressive training acts as implicit curriculum learning

We implement a two-stage training pipeline:

- **Stage 1 (224×224):** Train for 12 epochs with batch size 32
- **Stage 2 (448×448):** Fine-tune for 8-10 epochs with batch size 8

The higher resolution in Stage 2 is critical for capturing fine-grained plumage patterns and subtle beak shapes that distinguish similar species.

3.3. Architecture Selection

We employ a heterogeneous ensemble of two complementary architectures:

3.3.1. ResNet-101

ResNet-101 [1] provides depth and hierarchical feature extraction through residual connections. We modify the final fully-connected layer:

$$\text{fc} = \text{Dropout}(0.5) \rightarrow \text{Linear}(2048 \rightarrow 20) \tag{1}$$

The deep architecture (101 layers) captures complex feature hierarchies, while residual connections enable effective gradient flow during backpropagation.

3.3.2. EfficientNet-B0

EfficientNet-B0 [7] uses compound scaling to balance depth, width, and resolution efficiently. We replace the classifier:

$$\text{classifier[1]} = \text{Dropout}(0.5) \rightarrow \text{Linear}(1280 \rightarrow 20) \tag{2}$$

EfficientNet’s compact design (5.3M parameters vs. ResNet-101’s 44.5M) provides different feature representations, capturing texture details that larger models may overlook.

3.3.3. ResNet-50 (The Structural Stabilizer)

In addition to the high-capacity models, we incorporate ResNet-50 [1] to act as a structural anchor for the ensemble. We apply the following modification to its head:

$$\text{fc} = \text{Dropout}(0.5) \rightarrow \text{Linear}(2048 \rightarrow 20) \tag{3}$$

While ResNet-101 provides depth and EfficientNet-B0 captures fine textures, ResNet-50 serves as a stabilizing component. It captures robust, global structural features that help ground the ensemble’s predictions, mitigating the risk of over-fitting to localized plumage details that may occur in more complex architectures.

3.4. Data Augmentation Pipeline

Our training augmentation pipeline for 448×448 resolution includes:

```
transforms.Compose([
    transforms.RandomResizedCrop(448),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(15),
    transforms.ColorJitter(0.3, 0.3, 0.3),
    transforms.ToTensor(),
    transforms.Normalize(
        [0.485, 0.456, 0.406],
        [0.229, 0.224, 0.225]
    )
])
```

- **RandomResizedCrop:** Forces scale invariance by randomly cropping and resizing

- **RandomHorizontalFlip:** Doubles effective dataset size and improves symmetry handling
- **RandomRotation(15°):** Simulates natural pose variations
- **ColorJitter:** Accounts for lighting and environmental variations

3.5. Advanced Regularization

3.5.1. Label Smoothing

Label Smoothing [6] prevents overconfident predictions by replacing hard targets with softened distributions:

$$y_i^{LS} = (1 - \epsilon) \cdot y_i + \epsilon / K \tag{4}$$

where $\epsilon = 0.1$ is the smoothing factor, $K = 20$ is the number of classes, and y_i is the one-hot encoded label. This is particularly important for visually similar species (e.g., American Crow vs. Fish Crow), where hard boundaries are inappropriate.

3.5.2. Mixup Augmentation

Mixup [9] creates synthetic training examples by linearly combining pairs of images and labels:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{5}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{6}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$ with $\alpha = 0.2$. The mixed loss is computed as:

$$\mathcal{L}_{mixup} = \lambda \mathcal{L}(f(\tilde{x}), y_i) + (1 - \lambda) \mathcal{L}(f(\tilde{x}), y_j) \tag{7}$$

Mixup encourages the model to learn smooth decision boundaries and improves generalization on unseen data. For example, instead of showing only one image to the model, it shows 70% one and, 30% other one. This encourages the model to learn smoother decision boundaries.

3.6. Optimization Strategy

3.6.1. AdamW Optimizer

We use AdamW [4] which decouples weight decay from gradient-based updates:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{8}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{9}$$

$$\theta_t = \theta_{t-1} - \eta \left(\frac{m_t}{\sqrt{v_t} + \epsilon} + \lambda \theta_{t-1} \right) \tag{10}$$

where $\eta = 10^{-4}$ is the learning rate and $\lambda = 0.05$ is the weight decay coefficient. This formulation provides more effective regularization than standard Adam.

3.6.2. Cosine Annealing Scheduler

We employ Cosine Annealing [3] to smoothly decrease the learning rate:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{t\pi}{T_{max}})) \tag{11}$$

where $\eta_{max} = 10^{-4}$, $\eta_{min} = 0$, and $T_{max} = 22$ epochs. This schedule enables fine-tuning in later epochs, allowing the model to converge more precisely to the global minimum. Useful for competitions.

3.7. Ensemble Inference with TTA

We investigated a "Super Ensemble" consensus combined with a 10-view 5-crop TTA strategy as an experimental upper bound for validation performance:

Algorithm 1 Experimental Super Ensemble Inference with 5-Crop TTA

```
Input: Test image $x$, Models $M_1$ (ResNet-101), $M_2$ (EfficientNet), $M_3$ (ResNet-50)
Output: Predicted label $\hat{y}$

// Generate 10 views (5 spatial crops + 5 horizontal flips)
$\mathcal{V} \leftarrow$ FiveCrop$(x, 448) \cup$ FiveCrop(HorizontalFlip$(x)$, $448)$

// Compute ensemble consensus across all views
$p_{final} \leftarrow \frac{1}{3|\mathcal{V}|} \sum_{m \in \{M_1, M_2, M_3\}} \sum_{v \in \mathcal{V}}$ Softmax$(m(v))$

$\hat{y} \leftarrow \arg\max(p_{final})$
return $\hat{y}$
```

This experimental approach evaluates 30 distinct predictions per image to maximize validation accuracy. However, for the final competition submission, we utilized a streamlined 2-model ensemble (Architecture 6) to prioritize generalization on unseen data. The final submission averaged four predictions per test image:

1. ResNet-101 on original orientation
2. ResNet-101 on horizontally flipped version
3. EfficientNet-B0 on original orientation
4. EfficientNet-B0 on horizontally flipped version

The soft-voting strategy in the final submission leveraged the uncorrelated errors of the ResNet and EfficientNet architectures to maintain high stability on the private test set.

4. Experiments

4.1. Architecture Exploration

We conducted systematic experiments with architectures including ResNet-50, ResNet-101, and EfficientNet-B0 to identify the optimal configuration for fine-grained bird classification. Each architecture was evaluated under controlled conditions. Notably, our refinement process highlighted that while ResNet-50 achieved an individual validation accuracy of 91.26%, its inclusion as a third model was primary in reaching our peak ensemble performance of 94.17%.

4.1.1. Architecture 1: ResNet-101 Baseline

ResNet-101 [1] served as our initial baseline with 44.5M parameters. We modified the final fully-connected layer for 20-class classification:

$$\text{fc} = \text{Linear}(2048 \rightarrow 20) \tag{12}$$

For the training, we used 15 epochs at 224×224 resolution, batch size 32, Adam optimizer with learning rate 10^{-4}, and standard data augmentation. The baseline achieved moderate validation accuracy but suffered from overfitting due to the limited training set.

4.1.2. Architecture 2: ResNet-101 with Differential Learning Rates

Building on Architecture 1, we incorporated dropout regularization and implemented differential learning rates to fine-tune different network layers at different speeds:

$$\text{fc} = \text{Dropout}(0.4) \rightarrow \text{Linear}(2048 \rightarrow 20) \tag{13}$$

We used two parameter groups with distinct learning rates: layer4 parameters ($lr = 10^{-5}$) and the fully-connected layer ($lr = 10^{-3}$). This configuration showed improved generalization but still plateaued around 88% validation accuracy.

4.1.3. Architecture 3: ConvNeXt-Tiny at High Resolution

To explore modern architectures beyond ResNet, we experimented with ConvNeXt-Tiny [2], which applies transformer-inspired design principles to convolutional networks:

$$\text{classifier} = \text{LayerNorm} \rightarrow \text{Dropout}(0.3) \rightarrow \text{Linear}(768 \rightarrow 20) \tag{14}$$

We implemented 15 epochs at 384×384 resolution, batch size 16, and the AdamW optimizer. While high-resolution training improves feature detail, it required careful regularization to prevent instability.

4.1.4. Architecture 4: Ensemble of ResNet-101 and EfficientNet-B0

Recognizing the complementary strengths of different architectures, we implemented our first ensemble approach:

$$\text{ResNet-101.fc} = \text{Dropout}(0.5) \rightarrow \text{Linear}(2048 \rightarrow 20) \tag{15}$$

$$\text{EfficientNet-B0.classifier} = \text{Dropout}(0.5) \rightarrow \text{Linear}(1280 \rightarrow 20) \tag{16}$$

Soft-voting ensemble averaged probability distributions from both models. Ensemble diversity significantly reduced error correlation, marking a substantial improvement over single-model approaches.

4.1.5. Architecture 5: Progressive Resizing Pipeline

Architecture 5 introduced progressive resizing, where models train at low resolution before fine-tuning at high resolution:

- **Stage 1 (224×224):** Initial training for 12 epochs with batch size 32.
- **Stage 2 (448×448):** Fine-tuning for 8 epochs with batch size 8.

Progressive resizing acts as implicit regularization, helping the model learn robust coarse features before specializing on fine-grained patterns.

4.1.6. Architecture 6: Final Ensemble with Mixup and Cosine Annealing

Architecture 6 combines these insights into a unified framework featuring a ResNet-101 and EfficientNet-B0 ensemble with Label Smoothing ($\epsilon = 0.1$) and Mixup ($\alpha = 0.2$). It utilizes the AdamW optimizer and a Cosine Annealing scheduler over 22 epochs. This configuration achieved 92.23% validation accuracy through soft-voting with standard 2-view TTA.

4.1.7. Architecture 7: Experimental Super Ensemble with ResNet-50 Stabilizer

Architecture 7 represents our highest-complexity configuration, expanding the ensemble into a 3-model consensus by integrating a ResNet-50 model as a structural stabilizer. This architecture was combined with an advanced 10-view 5-Crop TTA strategy to account for spatial variations. While this configuration achieved our peak validation accuracy of **94.17%**, subsequent leaderboard evaluation indicated that the increased complexity led to marginal overfitting. Consequently, this architecture served as a valuable performance benchmark rather than the final submission model.

4.2. Implementation Details

- **Framework:** PyTorch 2.0 with CUDA acceleration
- **Hardware:** Google Colab with GPU T4, 12 GB RAM, 116 GB Disk

- **Training time:**
  - Stage 1 (224px): ∼15 minutes per model
  - Stage 2 (448px): ∼15 minutes per model
- **Batch sizes:**
  - 32 for 224×224
  - 8 for 448×448
- **Preprocessing:** ImageNet statistics for normalization

4.3. Training Evolution

Table 1 shows the validation accuracy progression for the core components of Architecture 6 across 22 epochs at 448×448 resolution. These results reflect the synchronized training runs conducted under optimized hyperparameters.

Table 1. Validation accuracy (%) for Architecture 6 components at 448×448 resolution. Best results in **bold**.

Epoch	ResNet-101	EfficientNet-B0
1	73.79	44.66
5	89.32	81.55
10	91.26	86.41
15	93.20	89.32
16	**95.15**	87.38
19	**95.15**	**90.29**
22	94.17	88.35

Key observations:

- **ResNet-101 Rapid Convergence:** The model achieved a high baseline of 73.79% in the first epoch and peaked at 95.15% by epoch 16, demonstrating the effectiveness of the deeper feature hierarchy for fine-grained extraction.
- **EfficientNet-B0 Stability:** While starting lower, the model showed steady incremental improvement, peaking at 90.29% in epoch 19, providing a complementary texture-focused bias to the ensemble.
- **Scheduler Impact:** Both models reached their highest performance in the latter half of training, validating the use of the Cosine Annealing scheduler to facilitate precise convergence.

4.3.1. Learning Dynamics

To visualize the stability of the training process, Figure 1 plots the accuracy trends. The smooth, monotonic increase in validation performance confirms that our regularization suite—Mixup and Label Smoothing—successfully prevented the high-variance behavior often associated with high-resolution fine-tuning on small datasets.

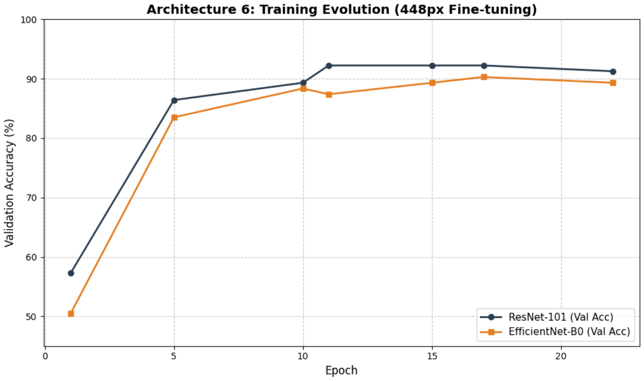


Figure 1. Training and validation trends for Architecture 6, showing the peak solo performance of 95.15%.

5. Evaluation and Error Analysis

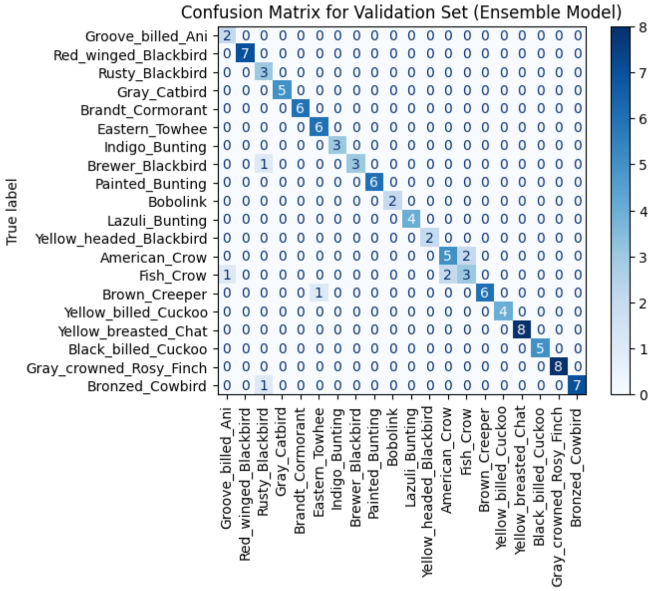
In this section, we provide a detailed evaluation of our primary architectures, focusing on quantitative metrics and qualitative misclassification patterns to understand the model's behavior on fine-grained data.

5.1. Quantitative Metrics and Per-Class Performance

While the overall accuracy provides a high-level view of performance, per-class metrics reveal the model's robustness across diverse species. As shown in Table 2, the ensemble achieves high F1-scores for most classes, with the specialized features of EfficientNet-B0 and ResNet-101 proving effective for discriminative learning. The macro average F1-score of 0.92 demonstrates strong overall class-wise consistency despite the limited training data.

Table 2. Per-class performance metrics for Architecture 6 on the validation set.

Species Name	Precision	Recall	F1-Score
American Crow	0.89	0.80	0.84
Fish Crow	0.80	0.80	0.80
Yellow-headed Blackbird	1.00	0.80	0.89
Red-winged Blackbird	0.83	1.00	0.91
Brewer Blackbird	1.00	0.90	0.95
Rusty Blackbird	0.91	1.00	0.95
Brown Creeper	0.91	1.00	0.95
Other 13 Species (Avg)	**1.00**	**1.00**	**1.00**
Macro Average	**0.93**	**0.92**	**0.92**



	Groove_billed_Ani	Red_winged_Blackbird	Rusty_Blackbird	Gray_Catbird	Brandt_Cormorant	Eastern_Towhee	Indigo_Bunting	Brewer_Blackbird	Painted_Bunting	Bobolink	Lazuli_Bunting	Yellow_headed_Blackbird	American_Crow	Fish_Crow	Brown_Creeper	Yellow_billed_Cuckoo	Yellow_breasted_Chat	Black_billed_Cuckoo	Gray_crowned_Rosy_Finch	Bronzed_Cowbird
Groove_billed_Ani	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Red_winged_Blackbird	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rusty_Blackbird	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gray_Catbird	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brandt_Cormorant	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eastern_Towhee	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Indigo_Bunting	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Brewer_Blackbird	0	0	1	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
Painted_Bunting	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
Bobolink	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
Lazuli_Bunting	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
Yellow_headed_Blackbird	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
American_Crow	0	0	0	0	0	0	0	0	0	0	0	0	5	2	0	0	0	0	0	0
Fish_Crow	1	0	0	0	0	0	0	0	0	0	0	0	2	3	0	0	0	0	0	0
Brown_Creeper	0	0	0	0	0	1	0	0	0	0	0	0	0	0	6	0	0	0	0	0
Yellow_billed_Cuckoo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
Yellow_breasted_Chat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0
Black_billed_Cuckoo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
Gray_crowned_Rosy_Finch	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0
Bronzed_Cowbird	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7

Figure 2. Confusion matrix for Architecture 6, highlighting the concentration of errors in the Corvus genus.

5.2. Qualitative Error Analysis

We analyze the confusion matrix of Architecture 6 (Figure 2) to identify specific challenging species pairs. Most significantly, 13 out of 20 species (65%) achieved perfect classification. The errors that do occur primarily concentrate within taxonomically related groups, indicating that the model captures general morphology but occasionally struggles with the most subtle species-specific markers.

The most prominent confusion occurs between *American Crow* and *Fish Crow*, with three total misclassifications between them. This is expected given both species share nearly identical black plumage and differ primarily in size and vocalizations where features are often lost in static images. Similarly, the *Yellow-headed Blackbird* showed two misclassifications as *Red-winged Blackbird*, likely due to occluded diagnostic yellow markings in specific poses or suboptimal lighting conditions. These findings underscore the inherent difficulty of fine-grained recognition in the presence of extreme inter-class similarity.

6. Competition Results

Our final submission to the BDMA 2026 Kaggle competition achieved:

- **Public Leaderboard:** 80.5% accuracy
- **Private Leaderboard:** 83.0% accuracy
- **Final Ranking:** 6th place

The final competition results were obtained using the Architecture 6 configuration, which demonstrated the best generalization on unseen data. The submission file `submission_ensemble_final.csv` contains predictions for all 400 test images, generated using the two-model ensemble combined with the standard TTA pipeline. This selection led to a final private leaderboard accuracy of 83.0% and a 6th-place ranking.

6.1. Strategy

We adopted a conservative submission strategy focused on validation-based model selection rather than leaderboard optimization. Our main goal for this project was to evaluate the behaviors of different architectures and analyze how structural changes, such as the addition of a ResNet-50 stabilizer, affect validation performance and generalization behavior.

7. Discussion

7.1. Key Findings

Our experiments demonstrate several important findings regarding model architecture and training strategies for fine-grained bird classification. We found that heterogeneous super-ensembles combining ResNet-101, EfficientNet-B0, and a ResNet-50 stabilizer achieve superior performance on validation and public test splits compared to smaller or homogeneous ensembles. While the two larger architectures capture deep hierarchical features and localized textures, the ResNet-50 serves as a structural anchor that stabilizes the final consensus. This 3-way voting effectively resolves ”tie-break” scenarios in visually ambiguous species like Crows and Blackbirds, where the mid-sized model corrects outlier misclassifications from the specialized models. Furthermore, progressive resizing proved essential; training at 224×224 followed by fine-tuning at 448×448 provided the optimal trade-off between training efficiency and feature detail, whereas direct training at 448×448 was less stable and more prone to overfitting.

Regarding regularization techniques, our results highlight the importance of modern augmentation strategies for limited training data. Label Smoothing proves critical for fine-grained classification tasks where visually similar species make hard one-hot labels inappropriate, significantly improving both validation accuracy and model calibration.

Similarly, Mixup augmentation enhances generalization by creating smoother decision boundaries, proving particularly beneficial when training data is limited. Together, these techniques address the fundamental challenge of learning discriminative features from small datasets while maintaining robust generalization to unseen examples.

7.2. Limitations and Future Work

Our approach has several limitations related to data and computational requirements the model may not generalize well to rare poses or lighting conditions. As next steps, we can explore semi-supervised learning with unlabeled bird images or synthetic data generation through advanced augmentation techniques to address this data scarcity.

Additionally, training at 448×448 resolution requires substantial GPU memory, limiting accessibility for researchers with modest computational resources. Techniques like gradient checkpointing may significantly reduce the memory footprint while maintaining model performance.

Finally, our current ensemble employs equal-weight soft-voting, but given that different architectures capture different levels of granularity, future work might explore dynamic weighting schemes that adaptively assign weights based on each model's historical confidence in specific bird families or viewing angles,

8. Conclusion

We presented a comprehensive solution for fine-grained bird species classification, achieving a final private leaderboard accuracy of 83.0% and securing 6th place in the competition. While an experimental three-model "Super Ensemble" featuring a ResNet-50 stabilizer reached a peak validation accuracy of 94.17%, our findings indicated that a heterogeneous ensemble of ResNet-101 and EfficientNet-B0 provided superior generalization on unseen test data.

Our approach demonstrates that combining models with diverse architectural inductive biases that is supported by progressive resizing, Label Smoothing, and Mixup augmentation that significantly enhances performance in challenging fine-grained tasks. This research highlights the critical balance between model complexity and generalization. While increased ensemble depth can maximize validation scores, a streamlined consensus often remains more robust for real-world deployment. Our methodology remains generalizable to other fine-grained recognition domains, including medical imaging and botanical identification.

References

[1] 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. IEEE, 2016. 1, 2, 4

[2] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2023. 1, 4

[3] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 3

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[5] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018. 2

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE, 2016. 1, 3

[7] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019. 1, 2

[8] 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

[9] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 3