Low-Resolution Mushroom Classification using Specialized CNNs and Novel Augmentation

Technical Report for Ho Chi Minh AI Olympiad

Group: BFC

Phan Trng Đài – HCMUS
Email: phantrongdaimath@gmail.com

April 26, 2024

Abstract

This report presents the methodology and results for our participation in the Ho Chi Minh AI Olympiad, focusing on the classification of four distinct Vietnamese mushroom species using a challenging low-resolution dataset. The task involved classifying images with a resolution of only 32x32x3 pixels, provided in a dataset comprising 1200 training images and 200 test images across four balanced classes. Addressing the difficulties posed by low resolution, our approach explored Convolutional Neural Network architectures specifically designed for such conditions, namely SPDResnet [1] and LRNet [2], adapted from recent research. We determined that careful selection of spatial data augmentation techniques, such as `RandomCropAndZoom` and `MultiScaleTransform`, was crucial, as overly aggressive transformations proved detrimental. A core component of our strategy involved introducing a novel "Mixup Class" technique, where a fifth class was generated by averaging images from the four original classes and included during model training. This approach, combined with the specialized architectures and targeted augmentations, significantly improved performance, leading to a final accuracy of 92.5% on the competition's public test set. Our work highlights the effectiveness of combining tailored architectures and a novel data augmentation strategy for robust classification despite severe image resolution constraints.

Contents

1 Introduction

Image classification, a fundamental task in computer vision and artificial intelligence, enables machines to categorize images based on their visual content. While significant progress has been made, classifying images with very low resolution presents unique and substantial hurdles. This report details our approach to tackling such a challenge within the context of the Ho Chi Minh AI Olympiad, which required participants to classify four distinct Vietnamese mushroom species: *Nm bào ng* (Oyster Mushroom), *Nm đùi gà* (Shaggy Ink Cap), *Nm linh chi trng* (White Reishi), and *Nm m* (Button Mushroom).

The primary challenge stemmed from the nature of the provided dataset: images were constrained to a low resolution of 32x32x3 pixels. This severe limitation drastically reduces the availability of fine-grained visual details necessary for distinguishing potentially similar mushroom species. Compounding this difficulty was the small dataset size (1400 images total), heightening the risk of model overfitting. Furthermore, the competition rules prohibited the use of external data, and pre-trained models from similar domains (like low-resolution mushroom classification) were unavailable, mandating that models learn relevant features primarily from the limited data provided. Observations also suggested a potential distribution shift between the training and testing datasets, adding complexity to model generalization. Finally, the evaluation methodology utilized varying random seeds, demanding solutions robust to initialization differences.

Given these constraints, the objective of our work was twofold: first, to develop and adapt deep learning architectures specifically suited for learning from low-resolution images, rather than relying on standard transfer learning approaches. Second, to systematically investigate the impact of various techniques – including data augmentation strategies, regularization methods, learning rate schedules, and training duration – on model performance within this challenging regime. A key focus was ensuring the robustness of our methodology and evaluation, employing strategies like cross-validation, careful monitoring of training dynamics, and rigorous validation set design.

Our approach centered on exploring and implementing models based on recent research targeting low-resolution vision tasks, specifically LRNet [2] and SPDResnet [1], while also comparing their performance against other architectures like Capsule Networks (CapsNet) and Vision Transformers (ViT). We conducted experiments to determine the optimal data augmentation techniques, finding a balance that aided generalization without destroying crucial low-resolution features. We also introduced a novel data augmentation strategy, the "Mixup Class", to further enhance model robustness. Emphasis was placed on robust validation practices, including a novel "Attack Validation" technique, to select the final models and assess their generalization capability accurately.

This report is organized as follows: Section 2 reviews related work in low-resolution classification. Section 3 details the dataset, preprocessing, data augmentation, model architectures, training procedures, validation strategy, and evaluation metrics. Section **??** presents our experimental setup, results, and comparisons. Section **??** discusses the findings and limitations, and Section 4 concludes the report.

2 Related Work

Classifying images at very low resolutions (e.g., 32x32 pixels) poses significant challenges due to the inherent loss of fine-grained visual information. Research in this area broadly follows two directions: approaches leveraging high-resolution (HR) data during training,

and those designed to work directly with low-resolution (LR) inputs, particularly when HR data is unavailable.

2.1 High-Resolution Guided Approaches

When HR images are available, several techniques attempt to bridge the resolution gap. One common strategy is Super-Resolution (SR), where LR images are first upscaled using an SR model before being fed into a standard classifier. While potentially effective, this requires either paired LR-HR data for training the SR model or a suitable pre-trained SR network .

Another prominent technique involves Dual-Branch Networks . These architectures often process LR and HR versions of an image (or pseudo-HR generated via SR) in parallel branches. The key idea is to transfer knowledge, often via feature distillation or shared feature subspaces, from the HR branch (which learns richer features) to guide the training of the LR branch, thereby improving its feature extraction capabilities . This allows the LR branch to learn more discriminative representations than it could alone. However, like SR-based methods, these approaches typically depend on the availability of corresponding HR data during the training phase.

2.2 Approaches for Direct Low-Resolution Classification

In scenarios where only LR images are available, such as the constraints of this competition, models must directly learn effective representations from the limited input. Several architectural paradigms have been explored:

2.2.1 Capsule Networks (CapsNets)

Proposed by Sabour, Frosst, and Hinton , CapsNets aim to better capture hierarchical spatial relationships between features using vector "capsules" instead of scalar neurons. Their emphasis on equivariance (how features change predictably with viewpoint transformations) rather than just invariance (discarding pose information, e.g., via pooling) holds theoretical promise for recognizing objects robustly even when fine details are scarce or viewpoints vary, which could be beneficial for LR images . However, ensuring true equivariance/invariance can be complex , and their performance relative to CNNs varies depending on the task. We specifically experimented with a hybrid architecture proposed in .

2.2.2 Vision Transformers (ViTs)

While originally designed for large-scale datasets , ViTs have been adapted for smaller, low-resolution datasets like CIFAR-10 (32x32) . ViTs process images as sequences of patches using self-attention mechanisms, allowing them to model long-range dependencies. However, training ViTs effectively on small datasets from scratch is challenging due to their lack of strong inductive biases compared to CNNs, often requiring significant data augmentation, regularization, specific training strategies, or hybrid architectures . Some works like Astroformer or TaylorIR combine transformer and convolutional elements to address these challenges.

2.2.3 Specialized Convolutional Neural Networks (CNNs)

Recognizing the limitations of standard CNN components for LR data, researchers have proposed modifications. A key issue is that traditional strided convolutions and pooling layers aggressively downsample feature maps, discarding potentially crucial spatial information . To mitigate this, Lee et al. [1] proposed the SPD-Conv block, which replaces strided convolutions/pooling with a sequence of space-to-depth (SPD) transformation followed by a non-strided convolution . This preserves finer spatial details by rearranging spatial information into channel depth. Similarly, Do et al. [2] introduced LRNet, featuring the MKBlock (Multi-Kernel Block), designed to capture features at multiple scales within the LR input using parallel convolutions with different kernel sizes.

2.3 Positioning Our Work

Given the competition constraints prohibiting external or high-resolution data, our work falls into the second category, focusing on methods that operate directly on the provided 32x32 images. We specifically investigated and adapted specialized CNN architectures designed for low-resolution input, namely SPDResnet (based on the SPD-Conv block [1]) and LRNet (based on the MKBlock [2]). We also explored the potential of alternative architectures like CapsNets and ViTs adapted for this low-resolution, small-dataset context, alongside investigating crucial data augmentation techniques, including the novel "Mixup Class", tailored for this challenging scenario.

3 Methodology

This section details the dataset, data handling procedures, model architectures, training regime, and evaluation strategies employed in our work. All implementations were developed using the PyTorch framework.

3.1 Data Augmentation Strategies

Data augmentation was critical for improving generalization and robustness. We carefully evaluated the impact of different augmentation types.

3.1.1 Color Augmentation

Experiments revealed that applying a diverse and relatively strong set of color transformations significantly benefited performance and training stability. These aim to make the model invariant to variations in lighting, camera sensors, and color casts. Using libraries like Pillow, OpenCV, and NumPy, we applied the following custom techniques randomly during training:

- `RandomColorDrop (p=0.2)`: Zeros out a random RGB channel.
- `RandomChannelSwap (p=0.2)`: Randomly permutes RGB channels.
- `RandomGamma (p=0.3, range=(0.5, 1.5))`: Applies random gamma correction.
- `SimulateHSVNoise (p=0.3, shifts H:±0.05, S:±0.1, V:±0.1)`: Adds small random shifts in HSV space.

- `SimulateLightingCondition (p=0.3)`: Applies one of several effects ('warm', 'cool', 'bright', 'dark').

An experiment with `RandomPixelNoise` (altering individual pixels) resulted in severe training instability, likely because each pixel holds significant information at 32x32 resolution, making random pixel corruption highly disruptive.

3.1.2 Spatial Augmentation

In contrast, spatial transformations required a conservative approach. Strong augmentations (Cutout, CutMix, Mixup) and even standard flips often led to instability, potentially due to erasing critical patterns in the low-resolution grid or challenges related to feature equivariance (discussed further in Section **??**). We therefore used subtle spatial augmentations mimicking capture variations:

- `MultiScaleTransform`: Randomly rescales the image within a defined range to simulate changes in camera distance.
- `RandomCropAndZoom`: Takes a random crop and resizes to 32x32, simulating off-center framing.

3.2 Mixup Class Technique

Adapting concepts from Mixup/Mosaic augmentation, we introduced a novel technique termed "Mixup Class" to handle ambiguity and improve robustness.

3.2.1 Implementation

An additional class (label index 4) was created. Images for this class were generated offline by randomly selecting one image from each of the four original classes, calculating their pixel-wise mean, and assigning the new label. These synthetic images were added to the training set, creating a 5-class problem during training.

3.2.2 Rationale

This yielded a notable performance increase (e.g., validation accuracy improved from 0.90 to 0.92). We hypothesize this helps by (1) providing an explicit category for highly ambiguous inputs, simplifying learning for original classes, and (2) encouraging more global feature learning to distinguish these composites, improving robustness against low-resolution challenges.

3.2.3 Comparison

Related concepts ("Mosaic Class", "Overlay Class") were tested but proved less effective in robustness evaluations (Attack Validation, see Section 3.5.2).

3.3 Model Architectures

We explored various CNN architectures, starting with adaptations of standard models before focusing on specialized low-resolution designs.

3.3.1 Standard CNN Adaptations

- *Modified ResNet*: A standard ResNet-was modified by replacing 3x3 convolutions with 7x7 kernels, aiming for larger receptive fields. This yielded relatively low accuracy.
- *Modified DenseNet*: To preserve spatial information, all pooling layers were removed, and all convolutions used `stride=1`. This maintained feature map size at 32x32 and achieved 0.82 accuracy, aligning with findings in works like SRBNAS that avoid downsampling.
- *DilatedGroupNet*: A custom network using Dilated Group Convolutions, inspired by , did not yield promising results.

3.3.2 Multi-Branch Network Explorations

Inspired by InceptionNet/XceptionNet, we designed custom multi-branch networks:

- *InceptionFSDNet*: Featured Inception-like blocks (multi-scale filters, SE attention), primarily without downsampling. Included a specific multi-branch Downsampling Module (used sparingly) and an SSD-inspired parallel feature pyramid structure.
- *MiniXceptionNet*: Adapted XceptionNet using depthwise separable convolutions, scaled down for the 32x32 task.

These architectures did not ultimately outperform the specialized models below.

3.3.3 Specialized Low-Resolution Architectures

- *SPDResnet*: Based on Lee et al. [1]. Implemented by replacing all strided convolutions and pooling layers in a base ResNet-with the SPD-Conv block (space-to-depth followed by non-strided convolution). This preserves spatial information by rearranging it into channels. Achieved 0.89-0.90 accuracy (before Mixup Class).
- *LRNet (`rlnet`)*: Implementation based on Do et al. [2], featuring the Multi-Kernel Block (MKBlock) with parallel convolutions of different kernel sizes to capture multi-scale features. Minor adaptations were made for 32x32 input. Achieved 0.89-0.90 accuracy (before Mixup Class).
- *Custom DualBranch Network*: Combined strengths of SPDResnet and global context learning.
  - Branch 1 (Local): Used the SPDResnet architecture.
  - Branch 2 (Global):
  - Fusion:

  Achieved 0.90 accuracy (before Mixup Class).

3.3.4 Experiments with Alternative Architectures

- *ViT Variants*: Direct pixel/patch transformers (`PixT`/`PatchPixT`) achieved 0.70 accuracy. Hybrid ViT-CNN models inspired by AstroTransformer /TaylorIR reached 0.78-0.80. Underperformance likely due to small dataset size and lack of inductive bias.
- *CapsNet Variants*: Implementation based on achieved 0.75 accuracy. Replacing its backbone with our modified DenseNet reached 0.80. Complexity may have hindered performance compared to simpler, more direct spatial preservation methods.

3.4 Training Procedure

The training process was carefully configured for stability and performance:

- **Loss Function:** Standard Cross-Entropy Loss.
- **Optimizer:** Adam optimizer.
- **Regularization:** L2 weight decay (`1e-5`). Standard Dropout layers were found detrimental and generally avoided.
- **Learning Rate:** Initial LR of 0.01, adapted using `ReduceLROnPlateau` scheduler monitoring validation loss. Effective configurations: (`factor=0.1, patience=15`) or (`factor=0.9, patience=5`).
- **Duration & Early Stopping:** Trained for 100 epochs on average. Early stopping used with high patience (`patience=50`) and warmup (`warmup=75`) to allow convergence.
- **Checkpointing:** Saving the model weights from the **last epoch** (`save_last_model`) yielded more robust final predictions than saving the best validation epoch weights.
- **Cross-Validation:** 6-Fold Stratified CV used (details in Section **??**).

3.5 Validation Strategy

A two-pronged approach ensured robustness:

3.5.1 Training Dynamics and Cross-Fold Consistency

Learning curves were monitored for stable convergence and consistent performance across all 6 CV folds. Models exhibiting instability or high variance were disfavored.

3.5.2 Attack Validation for Robustness Assessment

A novel procedure to test robustness against challenging perturbations using the public test set:

- **Procedure:** Generated 6 augmented versions of the public test set using strong, individual augmentations (Change Background, Colorize Mushroom, Darker Shadow, Random Blur, Rotate Zoom, Simulate Light).

- **Evaluation:** Assessed model accuracy on each attacked set. Models needed reasonable performance across all attacks, with a specific threshold of ≥ 0.70 accuracy on the challenging Random Blur attack.
- **Caveats:** Attacks used for validation only (not training). Performed on public test set only.

This step was crucial for eliminating seemingly good but non-robust approaches.

3.6 Evaluation Metrics and Final Ensemble Strategy

3.6.1 Metric

Accuracy was the primary metric, per competition rules.

3.6.2 Model Selection

SPDResnet and LRNet were chosen for the final submission based on CV performance and superior Attack Validation results.

3.6.3 Ensemble Strategy

1. Trained 6 SPDResnet and 6 LRNet models (one per fold for each architecture) using the `save_last_model` weights.
2. Generated raw **logit** outputs for the test set from all 12 models.
3. Averaged the 12 logit vectors element-wise for each test image.
4. Applied `argmax` to the averaged logits corresponding to the original 4 classes (ignoring the Mixup Class logit) to get the final prediction.

4 Conclusion

In this report, we presented our approach to classifying low-resolution (32x32) mushroom images for the Ho Chi Minh AI Olympiad. Faced with challenges of limited data, low resolution, and the need for robustness, we explored specialized CNN architectures, tailored data augmentation strategies, and novel validation techniques.

Our final ensemble model, combining SPDResnet and LRNet trained with a custom Mixup Class augmentation and selected through rigorous Attack Validation, achieved a public test set accuracy of 92.5%.

Key contributions include the demonstration of SPD-Conv and MKBlock effectiveness in this context, the development and validation of the Mixup Class technique, and the implementation of the Attack Validation procedure for robustness assessment.

References

[1] Lee, K., Narayanan, H., Le, T., Le, H., Pham, T., Nakashima, Y., ... & Rawat, A. S. (2022). *No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects.* arXiv preprint arXiv:2208.03641. `https://arxiv.org/abs/2208.03641`

[2] Do, T. V., Nguyen, T. H., Tran, M. T., Nguyen, H. P., Nguyen, C. V., Tjiputra, E. (2022). *LR-Net: A Block-based Convolutional Neural Network for Low-Resolution Image Classification.* arXiv preprint arXiv:2207.09531. https://arxiv.org/abs/2207.09531