Deep Learning Final Project: **Luis Espinoza, Cory Heins, Margaret Mullooly: November 24, 2025 Course: Introduction to Deep Learning**

Abstract This project investigates the performance of three CNN architectures, SimpleCNN, VGG11, and ResNet18, on a Pokémon image classification dataset comprising 110 classes as specified in the categories list, including categories such as Porygon, Goldeen, Hitmonlee, Hitmonchan, and Gloom. Through systematic hyperparameter exploration, the study evaluates optimization strategies and their effects on convergence and generalization. The resulting insights highlight the superiority of residual networks and the importance of learning rate scheduling in handling fine-grained visual tasks.

1. Introduction & Dataset Selection

The dataset selected for this project consists of Pokémon images divided into train, validation, and test sets, featuring 110 distinct classes.This choice in dataset was motivated by its representation of a real-world, fine-grained classification problem, where subtle visual differences (e.g., in Pokémon morphologies and colors) challenge model robustness. Unlike coarser benchmarks, such as CIFAR-10, the 110-class structure allows for the examination of scalability in label space, and the 64x64 resolution ensures computational feasibility in an introductory context.

Data preprocessing includes augmentation techniques, such as random resized cropping and horizontal flipping for training, in conjunction with normalization to standardize inputs. Validation and testing make use of resizing and center cropping in order to maintain consistency and improve generalization. This setup facilitates a focused analysis of how architectural and hyperparameter choices generalize to diverse visual domains, thus providing valuable lessons for practical deep learning applications across the specified Pokémon categories.

2. Model Architectures

The SimpleCNN serves as a lightweight baseline, comprising stacked convolutional layers with batch normalization, ReLU activations, and max pooling, followed by a dropout-regularized classifier outputting to 110 Pokémon classes. SimpleCNN emphasizes simplicity and efficiency, which made it the ideal choice for initial comparisons. However, its shallow design will most likely limit feature extraction on complex datasets, like the Pokémon images in our chosen dataset.

The VGG11 architecture introduces greater depth through repetitive 3x3 convolutions, with classification tailored to the 110 categories. This model relies on uniform layer

stacking to build hierarchical features, but it is prone to optimization challenges in the absence of careful tuning. In contrast, ResNet18 uses residual blocks which enable deeper training via shortcut connections. This helps to mitigate vanishing gradients and promotes stable learning for multi-class output. These architectures were selected for purposes of comparing plain versus residual designs and illustrating trade-offs in depth.

3. Experimental Setup & Hyperparameter Search

All models were trained on 64x64 Pokémon images with dataset-specific normalization, and configured for 110-class classification based on the provided categories. Model training incorporated the use of data augmentation, such as random resized crop and horizontal flip, with the goal of enhancing generalization. We used batch sizes 128 for training and 256 for both validation and testing.

Hyperparameter exploration involved four configurations: SGD with Step decay (LR=0.1, momentum=0.9, weight decay=5e-4, 200 epochs), SGD with Cosine annealing (similar parameters, 200 epochs), Adam without regularization (LR=0.001, 100 epochs), and Adam with weight decay (LR=0.0005, weight decay=1e-4, ReduceLROnPlateau, 150 epochs). Cross-entropy loss helped to guide optimization by allowing systematic assessment of effects on convergence speed, stability, and computational efficiency across the Pokémon categories. This resulted in three architectures with four different optimization techniques each for a total of twelve models for comparison.

4. Results & Analysis

The results reveal consistently low test accuracies, ranging from 0.00% to a maximum of 3.96%, across all model and configuration combinations. The respective results are depicted in the summary dot plot, found on slide 14. The plot illustrates that the highest performances were achieved by VGG11 with Adam_NoReg (3.96%) and ResNet18 with SGD_StepDecay (3.96%), followed closely by SimpleCNN with SGD_Cosine (2.19%) and ResNet18 with SGD_Cosine (2.19%). Lower scores include 0.00% for multiple setups, such as VGG11 with SGD_StepDecay and SimpleCNN with SGD_Cosine, which indicates substantial difficulties in achieving effective classification on the 110 Pokémon classes.

The resulting learning curve plots for key configurations demonstrate that training loss generally decreasing from initial values of approximately 10-12 to 2-4, whereas validation loss stabilizes or plateaus, and validation accuracy remains volatile and below 3.5% (e.g., oscillating sharply in ResNet18, SGD_StepDecay, and SimpleCNN Adam_NoReg). Confusion matrices further highlight sparse diagonal entries which

signifies minimal correct predictions. Moreover, off-diagonal scatters suggest random or unstructured misclassifications across classes. For instance, matrices for ResNet18 (SGD_StepDecay and SGD_Cosine) and VGG11 (SGD_Cosine) exhibit slightly denser clusters than those for SimpleCNN. This implies marginal improvements in capturing class similarities but overall failure in discrimination. Sample predictions from the best model demonstrate qualitative strengths, such as accurate classification of similar Pokémon categories like 'Pikachu' and 'Raichu'. However, the strengths are limited by the low overall accuracy.

The outcomes of the study emphasize the winners: ResNet18 with SGD-based configs, and highlight how scheduling prevents overfitting in deeper networks for multi-class tasks. Despite the results, there is marked room for improvement given the dataset's complexity.

5. Model Generalization and Insights

Although the study focused on a single Pokémon dataset with 110 specified categories, key observations suggest strong transferability: SGD with Cosine annealing, effective here, aligns with successes on similar vision benchmarks. However, the classification problem in 110 classes is sufficiently complex to produce a formidable challenge in accuracy metrics. The foregoing results indicate challenges in label scaling.

ResNet18's residuals provided greater advantages on this fine-grained task which widened the performance gap over VGG11 and SimpleCNN. Adam configurations often exhibited faster overfitting without weight decay.

6. Discussion, Limitations & Lessons Learned

In sum, the most effective strategies involved SGD with learning rate scheduling, and when combined with batch normalization and augmentation yielded improve training across 110 classes. Residual connections proved essential for depths exceeding 10 layers and enabled ResNet18's superior results. Adam matched SGD only with tuned weight decay and lower learning rates. This result revealed Adam’s sensitivity.

The respective limitations include the study’s reliance on a single 64x64 dataset with the specified Pokémon categories to the exclusion of higher resolutions or advanced techniques, such as Mixup. The lessons learned from this study include the realization that hyperparameters transfer well across similar domains, but architecture selection is critical for harder tasks. Future efforts could integrate modern architectures, such as Vision Transformers, for enhanced performance on fine-grained classifications. The dataset could be more thoroughly inspected for not just leakage but also proper

stratification of data to avoid testing/training splits where classes are underrepresented or entirely omitted.