**Deep Learning Bird Classification: Cheat Sheet**

**1. Problem Overview**

* **Objective:** Classify bird species based on input images.
* **Challenges:** High variability in poses, lighting, backgrounds, and species appearances.
* **Approach:** Use a convolutional neural network (CNN), specifically a pre-trained ResNet model, for feature extraction and classification.

**2. Model Selection: Why CNNs and ResNet?**

* **CNNs: Ideal for image data because they:**
  + Automatically learn spatial hierarchies of features.
  + Reduce parameters compared to fully connected networks by using shared weights (kernels).
* **ResNet (Residual Network):**
  + Solves the vanishing gradient problem by introducing skip connections.
  + Allows for very deep networks while maintaining efficient training.
  + Pre-trained ResNet models (e.g., ResNet-50 or ResNet-101) leverage features learned on large datasets like ImageNet, boosting performance on smaller datasets.

**3. Workflow**

1. **Data Preparation:**
   * **Preprocessing**: Resize images, normalize pixel values, and apply data augmentation (e.g., rotation, flipping, cropping) to increase variability and reduce overfitting.
2. **Model Architecture:**
   * Use a pre-trained ResNet (e.g., ResNet-50).
   * Replace the fully connected (FC) layer with a custom classifier for bird species (e.g., softmax layer with 200 outputs for 200 classes).
3. **Training:**
   * Fine-tune the model by unfreezing some layers (e.g., the last few ResNet blocks) while keeping others frozen to retain pre-trained features.
   * Use a learning rate scheduler to optimize training.
4. **Evaluation:**
   * Use metrics like accuracy, F1-score, and confusion matrix to evaluate performance.

**4. Common Questions and Answers**

**Why did you choose ResNet over other architectures (e.g., VGG, Inception)?**

* ResNet addresses the vanishing gradient problem, enabling the use of deeper networks, which often perform better for complex tasks like bird classification.
* ResNet is computationally efficient compared to architectures like VGG due to fewer parameters.
* Pre-trained ResNet models are widely available and have demonstrated high performance on similar tasks.

**Why is data augmentation important?**

* Augmentation increases the diversity of training samples, reducing overfitting by helping the model generalize better to unseen data.

**How does transfer learning help in this case?**

* The pre-trained ResNet model leverages features learned on a large, diverse dataset (ImageNet), such as edge detection and texture recognition.
* Fine-tuning the model tailors these features to bird-specific patterns, reducing the amount of data and time required for training.

**What loss function and optimizer did you use?**

* **Loss function:** Cross-entropy loss is standard for multi-class classification tasks.
* **Optimizer:** Adam optimizer is often used for faster convergence, combined with a learning rate scheduler.(not sure if we used a learning rate scheduler or not).

**How do you handle class imbalance?**

* **Use techniques like:**
  + Data augmentation for under-represented classes.
  + Weighted loss functions to assign higher penalties to misclassified minority class samples.

**How do skip connections in ResNet work?**

* Skip connections allow the network to bypass certain layers, passing the input directly to later layers.
* This helps gradients flow through the network during backpropagation, avoiding vanishing gradients and ensuring effective training.

**How did you evaluate the model's performance?**

* **Metrics:**
  + **Accuracy:** Measures overall correctness but can be misleading if classes are imbalanced.
  + **F1-score:** Balances precision and recall, especially useful for imbalanced datasets.
  + **Confusion Matrix:** Identifies which species are most often misclassified.
  + ‘*not sure if we have used a confusion matrix or F1 score or not but we could mention the issue with the Kaggle submissions and if we had more time we would have used them for evaluation.*’
* **Validation Strategies:** Used k-fold cross-validation to ensure robust evaluation.

**What challenges did you face, and how did you address them?**

* **Challenge:** Overfitting due to limited data.
  + **Solution:** Applied data augmentation and used dropout regularization.
  + *‘Again we thought we were over fitting so we used these techniques, we didn’t have enough time to really know if that was our problem or not.’*
* **Challenge:** Class imbalance.
  + **Solution:** Adjusted loss function weights and used augmentation.

**How does ResNet generalize better compared to other architectures?**

* ResNet’s skip connections simplify the learning of identity mappings, allowing the network to focus on learning more meaningful transformations.

**5. Key Hyperparameters**

* **Learning Rate:** Start with 0.001; use a scheduler to reduce it dynamically.(not sure if we used a scheduler or not)
* **Batch Size:** Experiment with sizes like 32 or 64, depending on GPU memory.(Beware of data leakage and compute crash XD)
* **Number of Epochs:** Monitor validation loss to decide when to stop (early stopping if necessary).
* **Augmentation:** Random rotations (0–30°), horizontal flips, and random crops.

**6. Visualization Tips**

* **Feature Maps:** Use tools to visualize feature maps learned by convolutional layers. This helps explain what the model focuses on (e.g., bird beaks, wings, etc.).(**AKA LIME**)
* **Confusion Matrix:** Present a confusion matrix to highlight specific strengths and weaknesses (e.g., which species are most misclassified).

**7. Practical Advice for Handling Questions**

* **Stay confident in explaining your choices and methods.**
* **If unsure of a question, admit it and discuss how you’d approach finding the answer (e.g., further experiments or research).**
* **Bring visual aids (e.g., graphs, examples of bird misclassifications) to make your answers more compelling.**