Yash Bansal (2022CS51133) Nikhil Zawar (2022CS11106)

COL333 Assignment 3:- Part 1

1 Introduction

This report illustrates the architecture, results and various optimisations for training a CNN-based image classification model for 10-class supervised classification of birds. We have achieved validation accuracy of around 95.5% for 80-20 train-validation split on the given training dataset.

2 Architecture

- **Input image size:-** 336×336
- **Batch size:-** 128
- **Loss function:-** Weighted Cross Entropy Loss
- **Optimizer:-** Adam's optimizer with initial Learning rate 0.001
- **Regularisation:-** L2 regularisation with weights decay 0.001
- **Learning rate scheduler:-** Step scheduler with step size 10 and reduction factor 0.5
- **Total parameters:-** Approx. 11 million

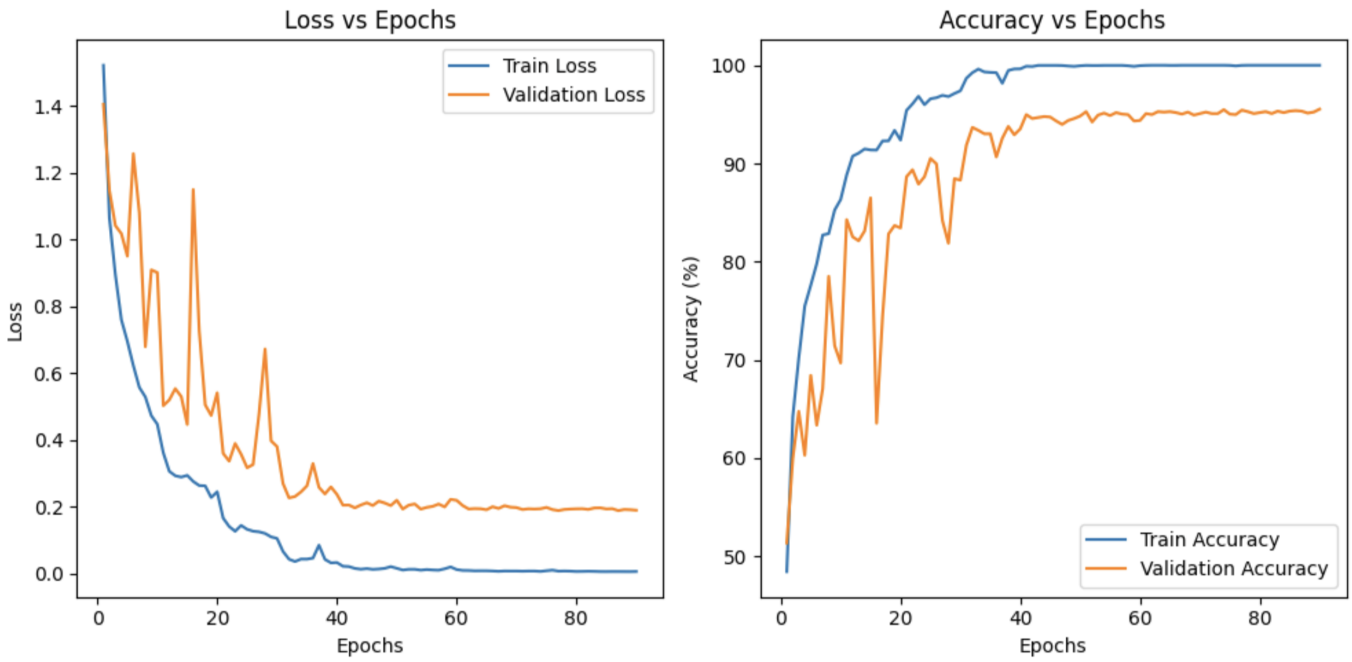


Figure 1: Plot for the final model used

Layer	Output Size	Layer Details
Conv1	$64 \times 168 \times 168$	Conv2d(3, 64, kernel_size=7, stride=2, padding=3), BatchNorm2d(64), ReLU, MaxPool2d(kernel_size=3, stride=2, padding=1)
Layer1	$128 \times 84 \times 84$	2 × ConvBlock (in_channels=64, out_channels=128, stride=1)
		- ConvBlock 1: Conv2d(64, 128, kernel_size=3, stride=1, padding=1), BatchNorm2d(128), ReLU - Conv2d(128, 128, kernel_size=3, stride=1, padding=1), BatchNorm2d(128), ReLU, Downsample last layer - Squeeze-and-Excitation: AdaptiveAvgPool2d(1), Linear(128, 8), ReLU, Linear(8, 128), Sigmoid
Layer2	$256 \times 42 \times 42$	2 × ConvBlock (in_channels=128, out_channels=256, stride=2)
		- ConvBlock 1: Conv2d(128, 256, kernel_size=3, stride=2, padding=1), BatchNorm2d(256), ReLU - Conv2d(256, 256, kernel_size=3, stride=1, padding=1), BatchNorm2d(256), ReLU, Downsample last layer - Squeeze-and-Excitation: AdaptiveAvgPool2d(1), Linear(256, 16), ReLU, Linear(16, 256), Sigmoid
Layer3	$512 \times 21 \times 21$	2 × ConvBlock (in_channels=256, out_channels=512, stride=2)
		- ConvBlock 1: Conv2d(256, 512, kernel_size=3, stride=2, padding=1), BatchNorm2d(512), ReLU - Conv2d(512, 512, kernel_size=3, stride=1, padding=1), BatchNorm2d(512), ReLU, Downsample last layer - Squeeze-and-Excitation: AdaptiveAvgPool2d(1), Linear(512, 32), ReLU, Linear(32, 512), Sigmoid
Avg. Pooling	$512 \times 1 \times 1$	AdaptiveAvgPool2d((1, 1))
Flatten	512	Flatten
FC (Classifier)	10	Linear(512, 10)

Table 1: Architecture Description

3 Optimisations

Our initial basic model gave us an accuracy of 86% on validation data. We applied various optimisations to improve our model up to about 95.5% accuracy.

1. **Batch size:** experimenting with different batch sizes and considering the GPU limit and train time constraints, a batch size of 128 was finalized.
2. **Image Transformations:** random vertical and horizontal flips, and image normalisation: 88.5% accuracy. Also, the image is finally reduced to 336×336.
3. **Dropouts:** Does not converge in the given time limit: Dropped
4. **LR scheduling:** Training losses were not very stable due to the higher learning rate at the latter epochs, so we used step LR scheduler, which reduces LR by a factor of 2 every ten epochs.
5. **Architecture experimentations:** Experimented with various architectures by varying the number of layers, input/output sizes, kernel sizes and strides and found the above-described model best to suit the given dataset.
6. **Bias in conv layers:** Accuracy improved to 89.5 %
7. **Class imbalance:** Got much stable learning and more generalised model: improved to 91%.
8. **Squeeze and excitation block:** Applied after every conv block to extract the most useful features from the layer. improved to 94 %
9. **L2 Regularisation:** Experimented with different weights decay parameters for L2 regularisation, chose weights decay 0.001 to achieve smooth convergence and less overfitting on the train data. Improved accuracy to 95.5 %

4 Class Activation Maps

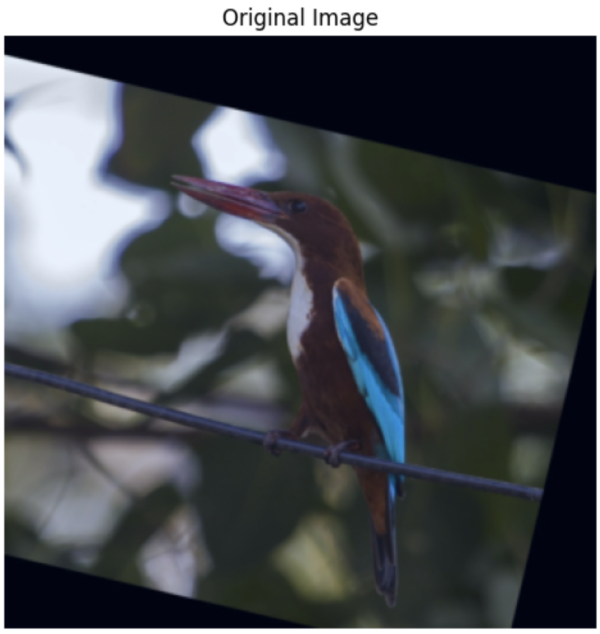


Figure 2: Original Image

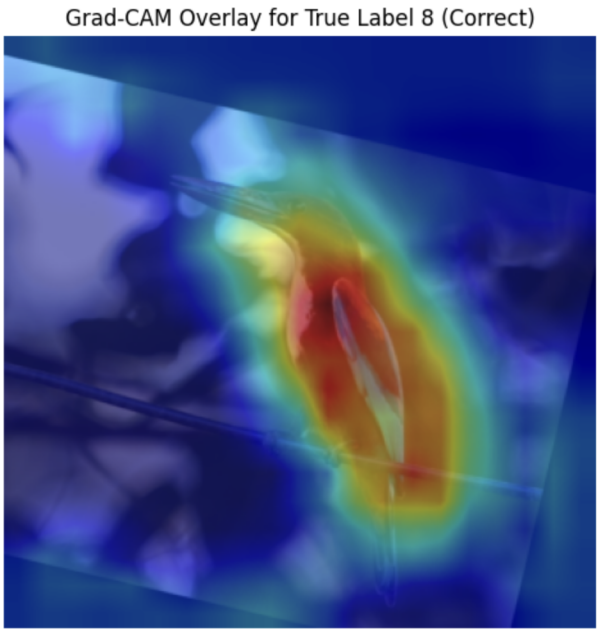
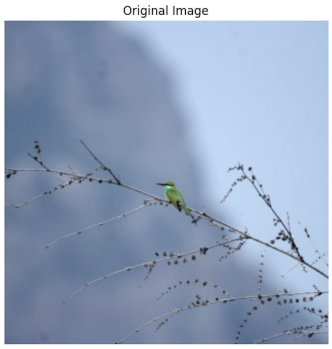
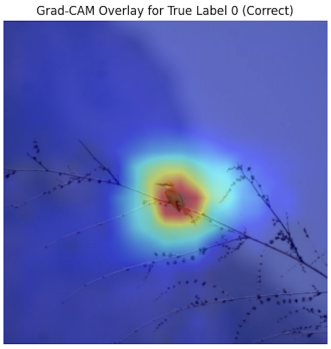


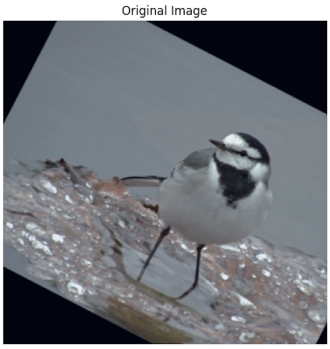
Figure 3: GradCAM overlay for true label



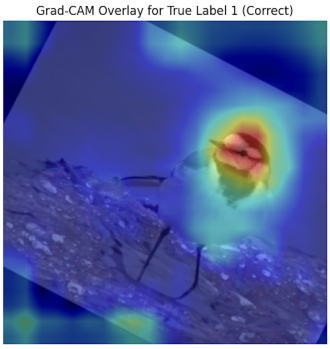
Class 0 Original



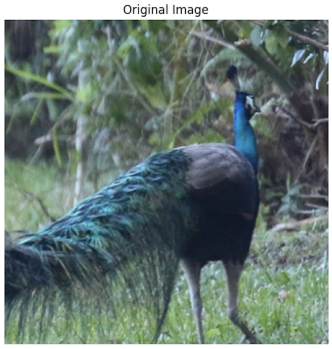
Class 0 Grad-CAM



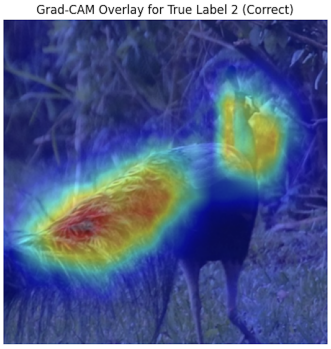
Class 1 Original



Class 1 Grad-CAM

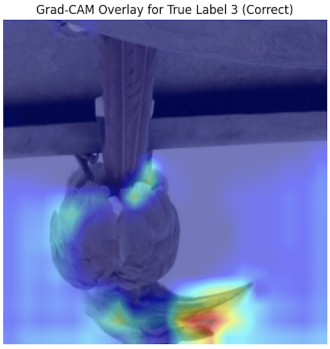


Class 2 Original



Class 2 Grad-CAM

Class 3 Original

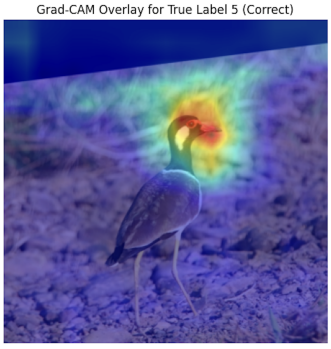


Class 3 Grad-CAM

Class 4 Original

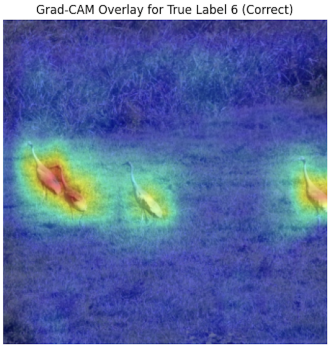
Class 4 Grad-CAM

Class 5 Original



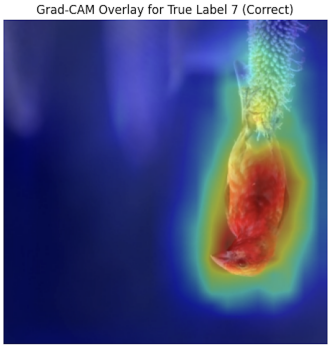
Class 5 Grad-CAM

Class 6 Original



Class 6 Grad-CAM

Class 7 Original



Class 7 Grad-CAM

Class 8 Original | Class 8 Grad-CAM | Class 9 Original | Class 9 Grad-CAM

Figure 4: Comparison of GradCAM overlays with original images for all the labels

- As can be seen from the Grad-CAM overlay for the true label, the model identifies the bird's location in the image with a very high accuracy. It correctly identifies the bird's body, head, and beak, which helps the model classify the bird according to its correct label with a very high probability.
- In some Grad-CAMs, the model sometimes starts identifying the tree branch as a bird's beak; thus, in label 9 Grad-CAM, it can be seen from the highlighted area the branch is identified, as label 9 contains a beaked bird.
- Sometimes, the model can get overfitted to the boundaries of the bird, due to which misclassifications can occur.
- Common features between different birds can also be seen by Grad-CAM, like beak, tail or similar feathers.