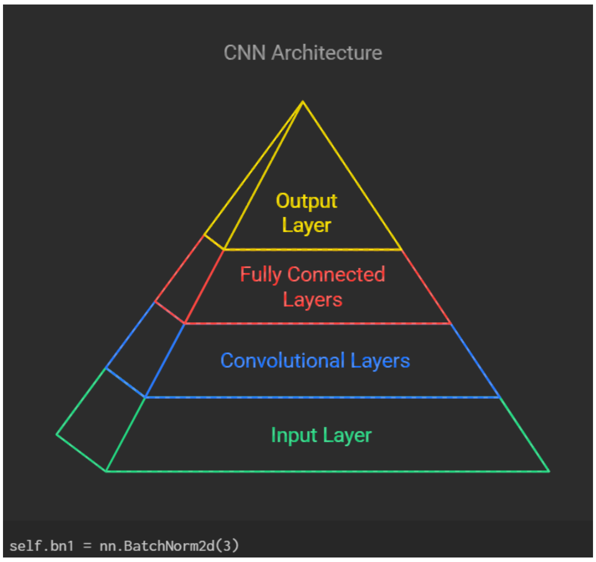
Indian Institute of Technology Delhi

COL333: Principles of Artificial Intelligence

Assignment - 3 Part - 1

Learning with Neural Networks

Model Architecture

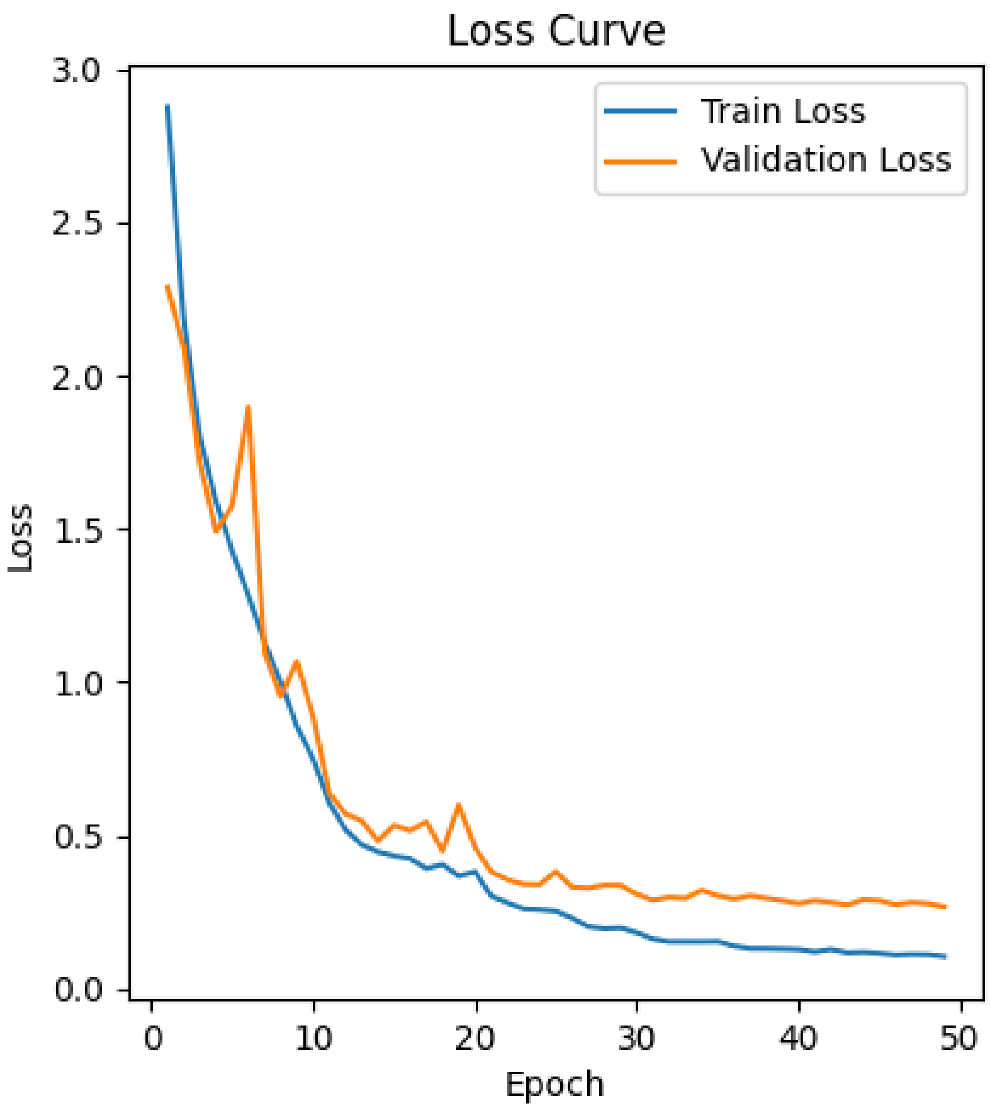


The details of the various layers are provided in the table below

Table 1: CNN Model Architecture

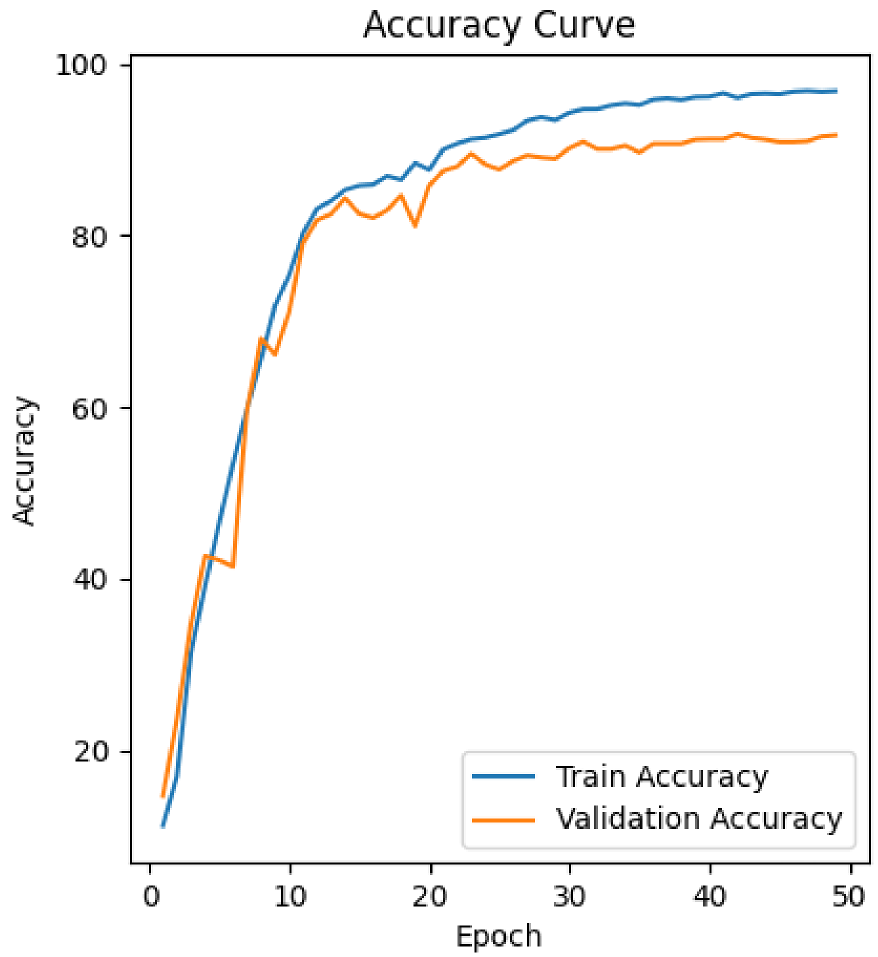
Layer	Type	Input Channels	Output Channels	Kernel Size	Stride	Padding	Output Dimension	Dropout
bn1	BatchNorm2d	3	3	-	-	-	3 x 128 x 128	-
conv1	Conv2d	3	16	3 x 3	1	1	16 x 128 x 128	-
pool1	MaxPool2d	16	16	2 x 2	2	0	16 x 64 x 64	-
bn2	BatchNorm2d	16	16	-	-	-	16 x 64 x 64	-
conv2	Conv2d	16	32	3 x 3	1	1	32 x 64 x 64	-
pool2	MaxPool2d	32	32	2 x 2	2	0	32 x 32 x 32	-
bn3	BatchNorm2d	32	32	-	-	-	32 x 32 x 32	-
conv3	Conv2d	32	64	3 x 3	1	1	64 x 32 x 32	-
pool3	MaxPool2d	64	64	2 x 2	2	0	64 x 16 x 16	-
bn4	BatchNorm2d	64	64	-	-	-	64 x 16 x 16	-
conv4	Conv2d	64	128	3 x 3	1	1	128 x 16 x 16	-
pool4	MaxPool2d	128	128	2 x 2	1	0	128 x 15 x 15	-
bn5	BatchNorm2d	128	128	-	-	-	128 x 15 x 15	-
conv5	Conv2d	128	256	3 x 3	1	1	256 x 15 x 15	-
pool5	MaxPool2d	256	256	2 x 2	1	0	256 x 14 x 14	-
dropout_conv	Dropout	-	-	-	-	-	256 x 14 x 14	0
fc1	Linear	256 x 14 x 14	256	-	-	-	256	0.25
fc2	Linear	256	64	-	-	-	64	0.25
fc3	Linear	64	10	-	-	-	10	-

Train and Validation Loss vs Epochs



Cross Entropy Loss as a function of number of epochs

Train and Validation Accuracy vs Epochs



Train and Validation Accuracies as a function of number of epochs

Effects of Model Optimisation

1. L2 Regularisation

Without L2 Regularisation, the model was overfitting to the training dataset with training dataset reaching a very high accuracy of about 99% whereas validation accuracy was only about 65%. Applying L2 Regularisation and fine tuning the weight decay parameter decreased the training accuracy to about 97% and validation accuracy increased to 70% without any data augmentation.

Validation Accuracy

without L2 Regularization: 65%

with L2 Regularization: 70%

2. Increasing the Number of Convolution Layers

With only 3 convolution layers being used, the validation accuracy was much lower for the model. Increasing the number of convolution layers to 5 had a significant impact on the validation accuracy.

Validation Accuracy

With 3 Convolution Layers: 70%

with 5 Convolution Layers: 81%

3. Data Augmentation

Transforming the training dataset by introducing random horizontal flips, vertical flips, random rotation of the training images and color jitter to fit the model better has a significant impact on the Model Accuracy. The training accuracy remained similar but the validation accuracy increased to about 84%. This was because it helped the model to generalise better for making predictions on the validation set.

Validation Accuracy

Without Data Transformation: 81%

with Data Transformation: 84%

4. Batch Normalization

Earlier applying batch normalization to only the initial input gave a validation accuracy around 84%. Applying batch normalization before every convolution layer increased the validation accuracy from 84% to 87% while training accuracy consistent around 97%.

Validation Accuracy

With Batch Normalization in first Convolution layer: 84%

with Batch Normalization in all Convolution layers: 87%

5. Scheduler for Learning rate

Used a learning rate scheduler to decrease the learning rate on reaching high validation and training accuracies. Started with a initial learning rate of 0.002 and halved it after 10 epochs to 0.001 and then halved it every 5 epochs. This helped the model to converge better. The validation accuracy increase from 87% to 89% with training accuracy remaining around 97%.

Validation Accuracy

Without learning rate Scheduler: 87%

With Learning rate Scheduler: 89%

6. Dropouts

Added dropout with hyper-parameter 0.25 to the neurons in the fully connected layer to help model avoid overfitting the training dataset and for more generalised training. This decreased the training accuracy from 97% to 95% while validation accuracy increased from 89% to 91%.

Validation Accuracy

Without Dropout: 89%

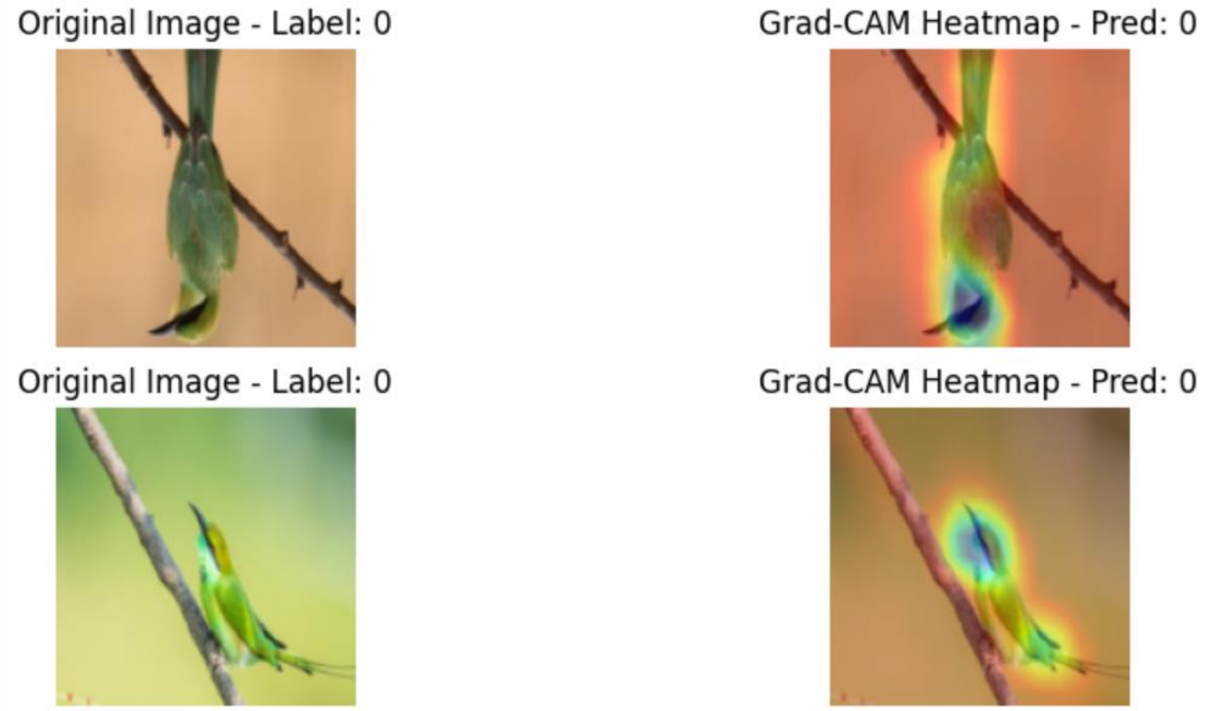
With 25% Dropout : 91%

7. Loss Functions

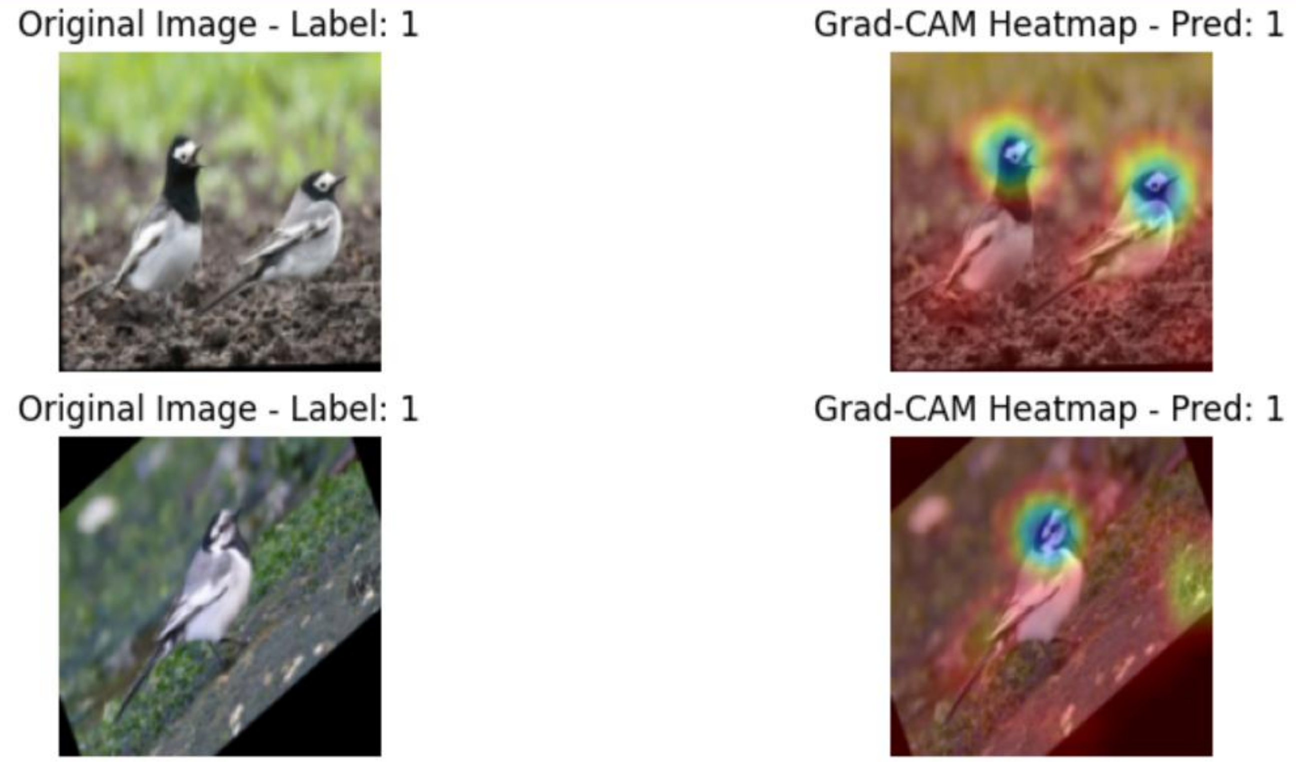
Tried Focal Loss to account for class imbalance in the training dataset but the validation accuracy decreased. So utilised the cross-entropy loss itself for the model training.

Gradient-weighted Class Activations Maps

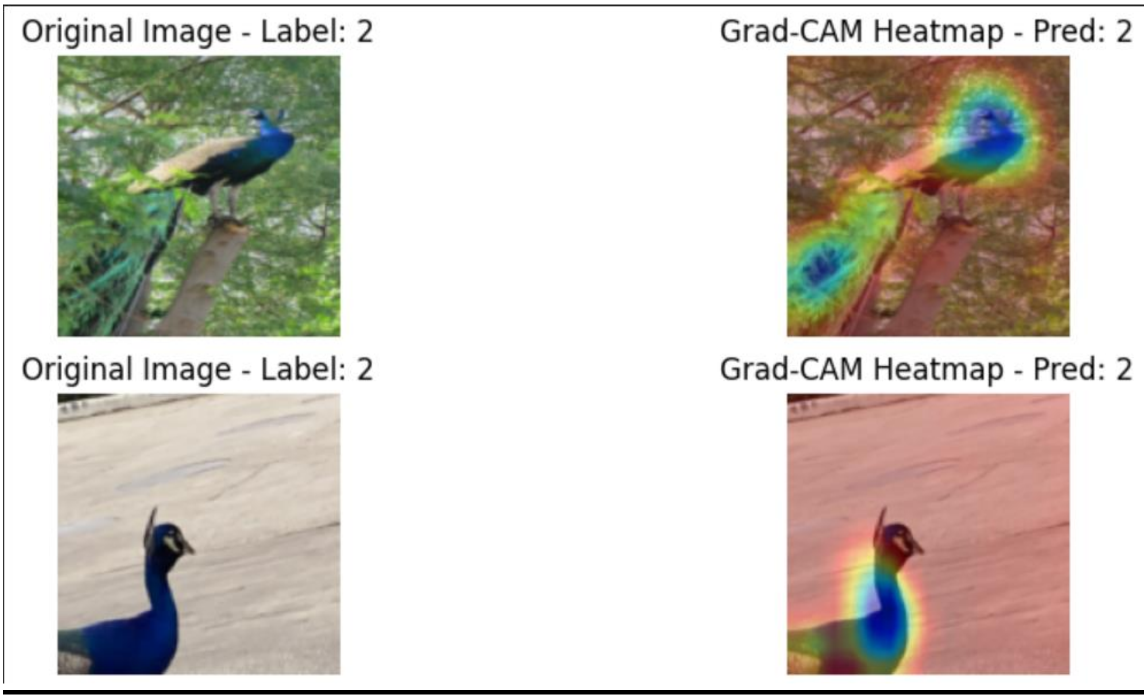
Observations by Class



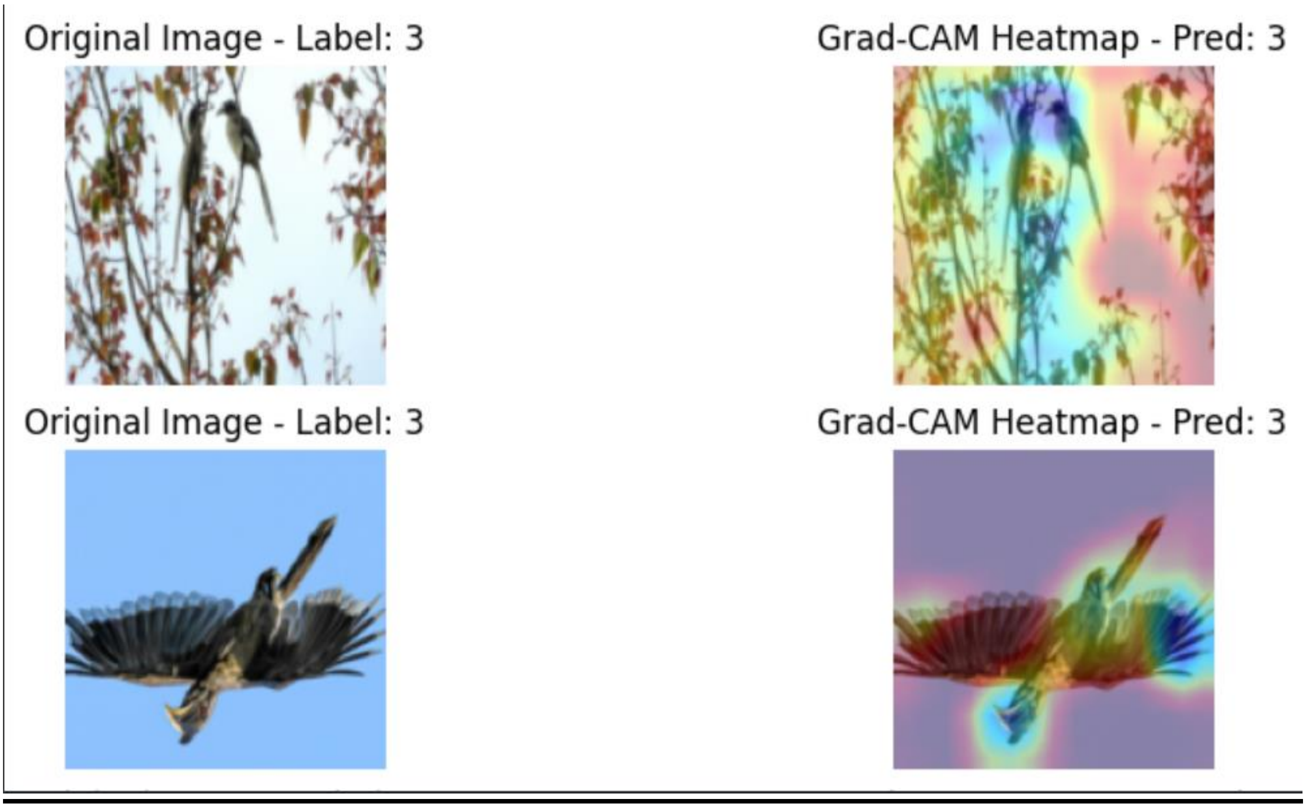
1. **Class 0**: The highlighted areas focus primarily on the bird's head and upper body, indicating these features are critical for the model to identify Class 0.



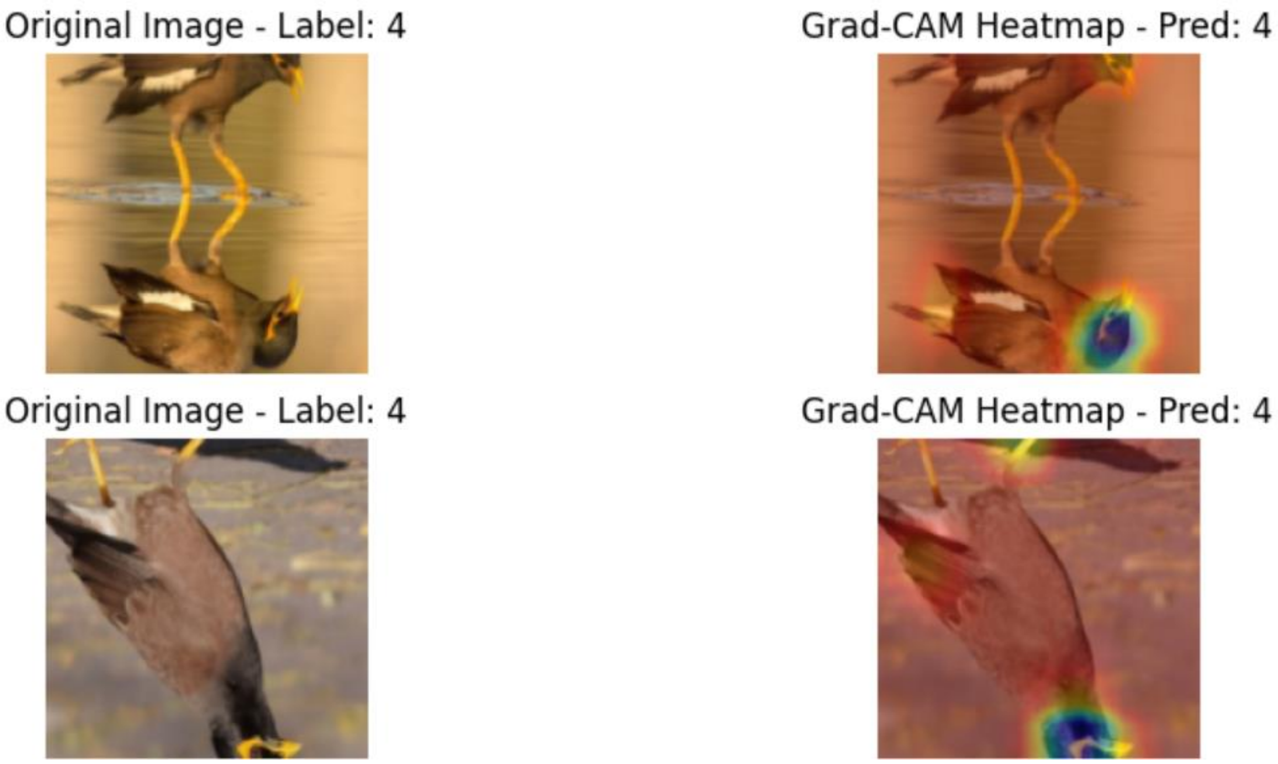
2. **Class 1**: The heatmap shows strong activation on the birds' heads, suggesting the model is using their distinctive head shape and markings as key features to classify the image.



3. **Class 2**: The model focuses on the peacock's vibrant body and feather pattern, which likely contain distinctive texture and colour patterns used to recognize this class.

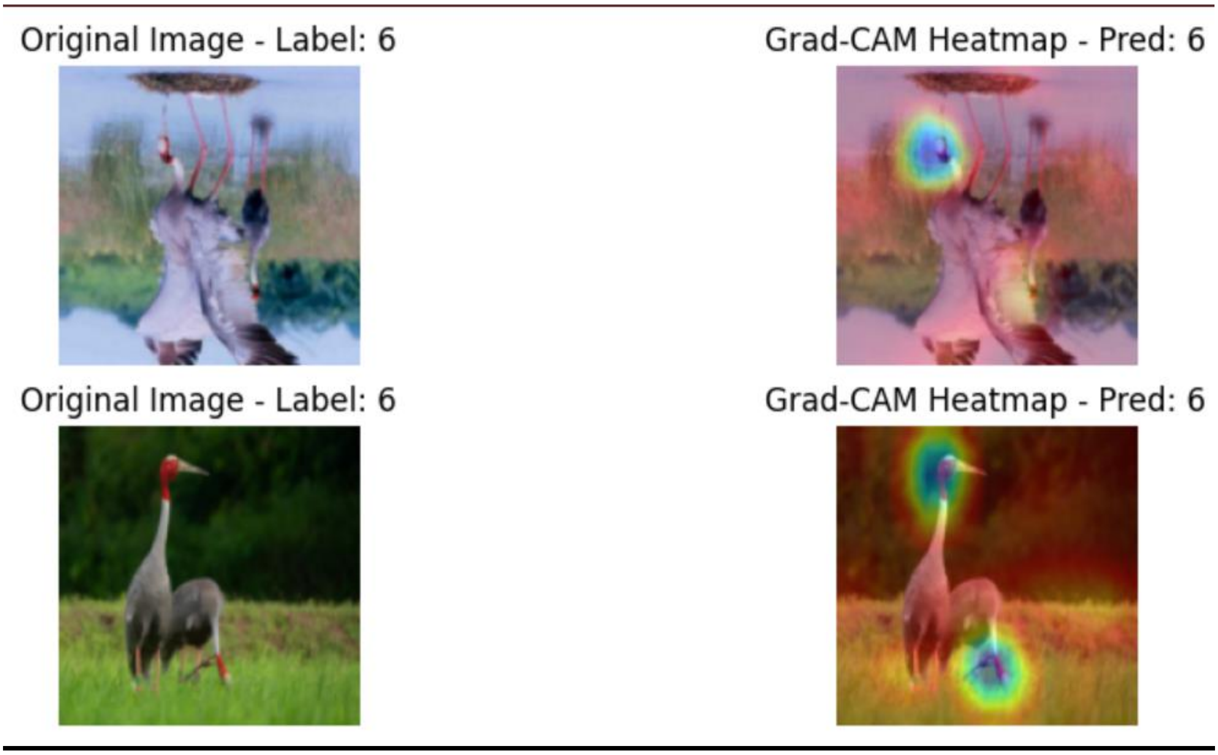


4. **Class 3**: The model concentrates on the bird's body, head and wings, emphasizing these areas for classification.

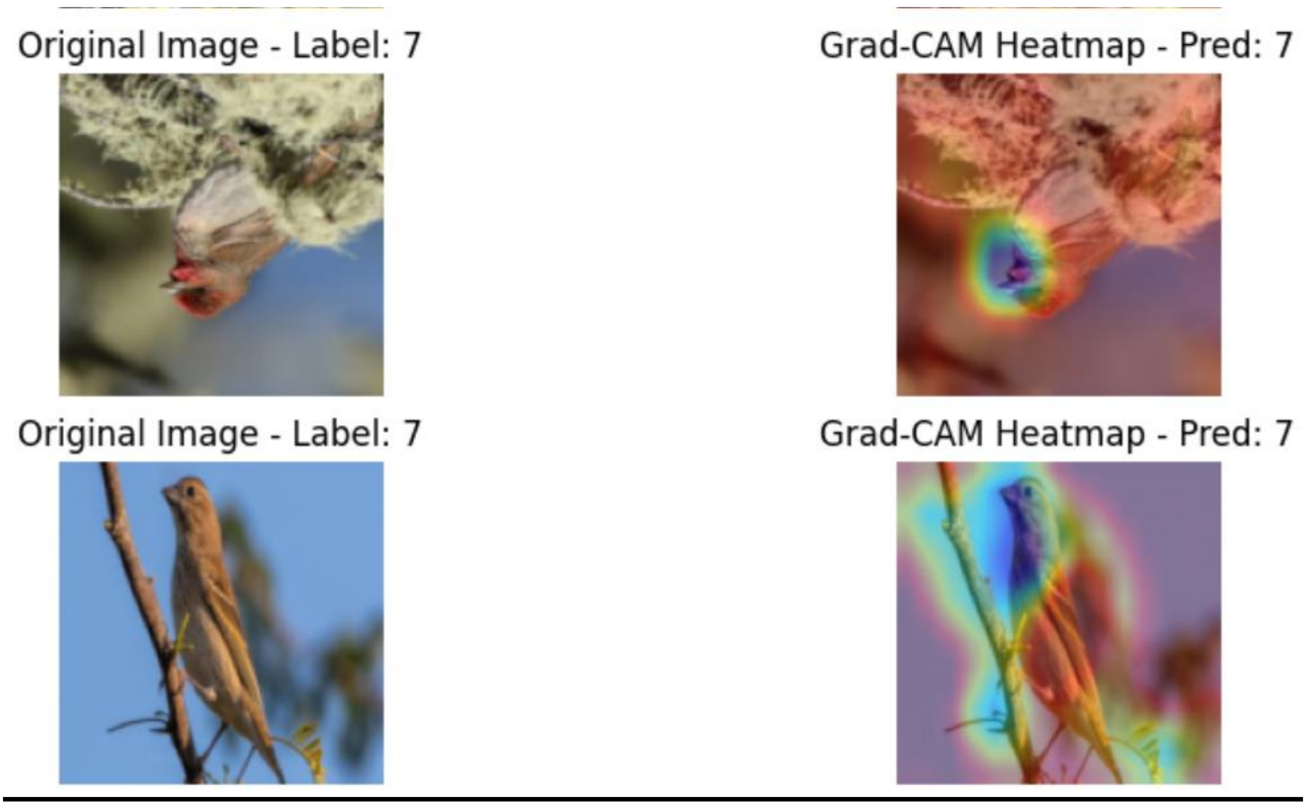


5. **Class 4**: The focus is on the bird’s head, suggesting that the model uses the bird's head to identify this class.

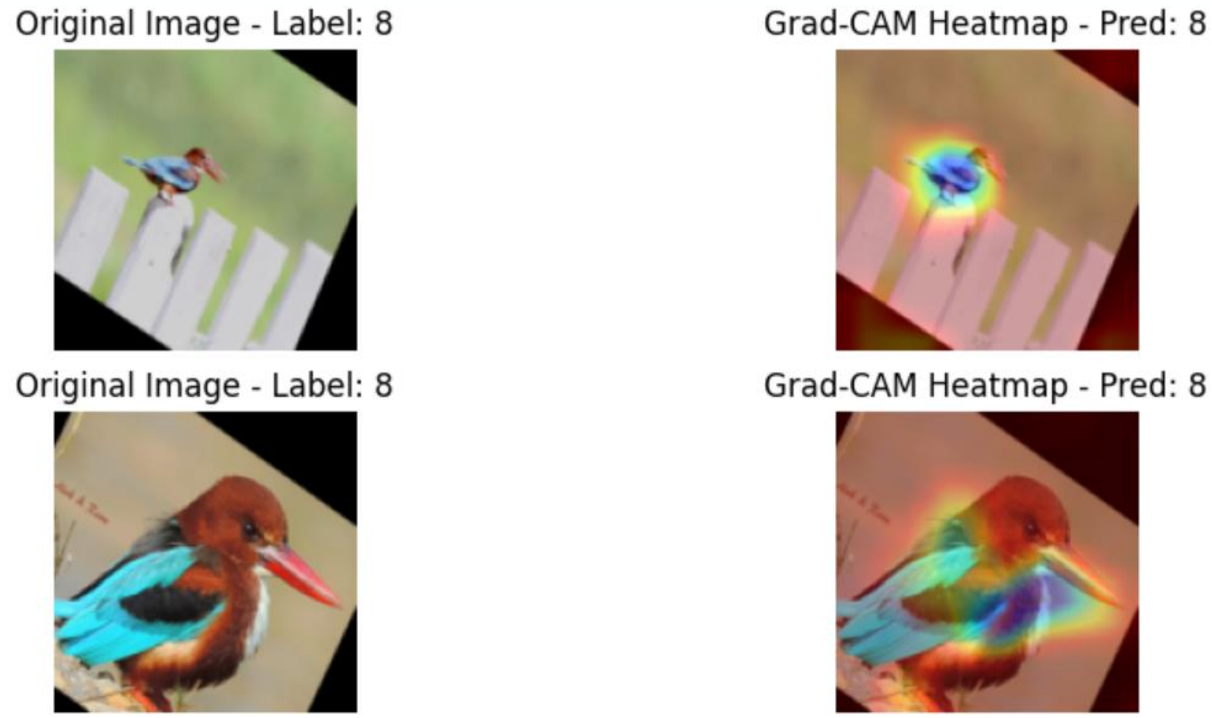
6. **Class 5**: The focus is on the bird’s head, suggesting that the model uses the bird's head to identify this class



7. **Class 6**: The model focuses on specific birds’ head rather than the entire group, indicating it identifies the class based on individual bird features rather than the overall scene. The focus is on individual bird’s head, suggesting that the model uses the bird's head to identify this class.



8. **Class 7**: Strong activation on the bird, particularly around its upper body parts and head, suggesting these specific features are essential for identifying Class 7.



9. **Class 8**: The model focuses heavily on the bird, with less attention to the surrounding pole. This suggests the model can distinguish the bird from its perch for classification. Again, the model focuses on the bird’s head for its classification.

10. **Class 9**: The model identifies this class based on the bird's unique pattern and shape against the complex background.

Inference from The Grad-CAM Results

Insights and Observations

Model Focus: The Grad-CAM visualizations consistently focus on the bird in each image, particularly on distinct parts like the head, showing that head feature is critical for the model's classification.

Class-Specific Features: For certain classes, the model appears to rely on unique textures or colors (like the peacock in Class 2), while for others, it relies mainly on the head and the upper body part of the bird.

Background Influence: From the Grad-CAM visualizations, we observe that the background has very little or no influence on the model’s prediction for various classes. The model mainly relies on the bird and the features associated with it like head, specific colour texture etc. for making predictions.