COL775: Deep Learning

Assignment 1.1: Resnet over Convolution network and different Normalization schemes

Name: Mikshu Bhatt **Entry NO:** 2023AIB2067

---

1. Image Classification using Residual Network

ResNet, short for Residual Network, focuses on learning residual mapping by passing the input of the t-th layer directly to the (t + 2)-th layers through residual connections. Studies indicate that as networks grow deeper, it becomes challenging for them to learn identity mapping effectively, leading to decreased performance compared to shallower networks.

Skip connections address this issue by preventing the vanishing gradients problem typical in deep networks. These connections facilitate the direct flow of information between distant layers, thereby enhancing the training efficiency of deeper networks.

	Micro F1	Macro F1	Accuracy
Train set	98.52%	98.52%	98.52%
Val set	96.55%	97.35%	96.55%
Test set	91.66%	81.81%	91.66%

Table 1: Performance of best performing model (Pytorch inbuilt batch normalization)

Key parameters used in model training

1. Hyper parameters: Batch size = 32, Max epochs = 50, n = 2, r = 25.
2. Criterion: Cross entropy loss
3. Optimizer: SGD optimizer with 0.001 learning rate.
4. Scheduler: initially 2 schedulers namely **StepLR** and **CosineAnnealingLR** were choosen but for the base case model, CosineAnnealingLR was giving better performance as the loss curve was smooth compared to StepLR.

2. Impact of Normalization

2.1 Sanity checks with comparison of Torch and Custom Batch Normalization

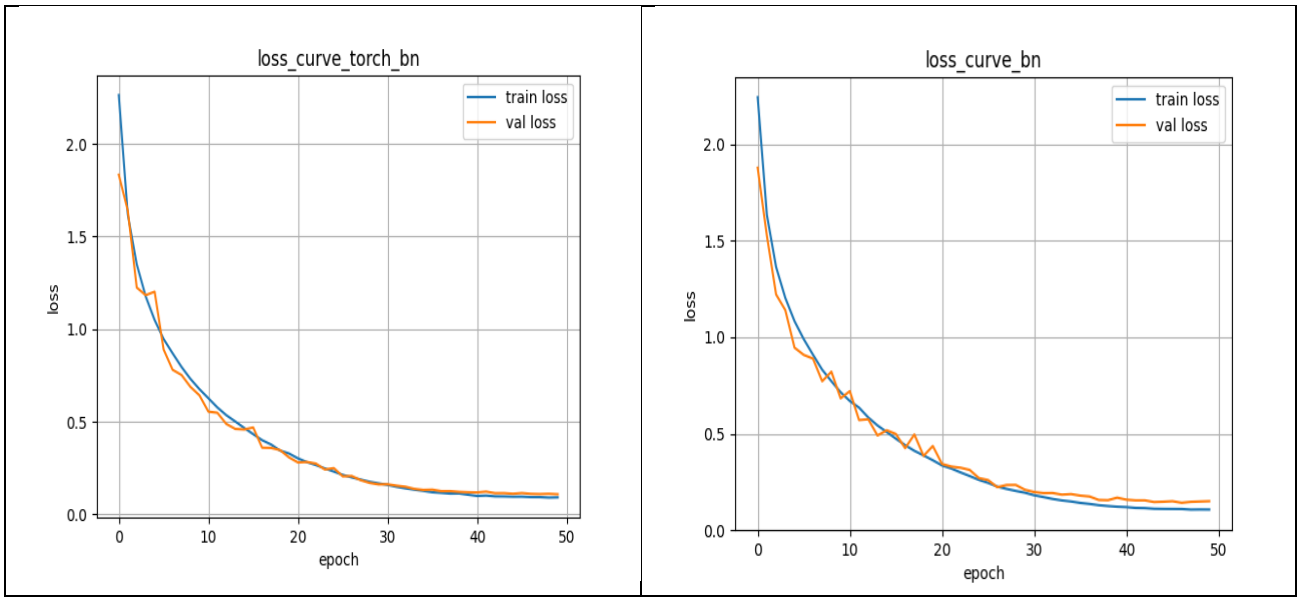


Fig 1: comparison of Train and Validation loss curves of Custom Batch norm and pytorch inbuilt batch norm

- From fig 1, we can see that loss curves of both normalizations are similar, however in case of pytorch batch norm, training and validation loss both converge at similar points but in case of custom batch norm, validation loss is higher than the training loss at the end of the training.
- Training and validation accuracy are also similar in both the cases.

2.2 Comparison of the performance of all the Normalizations

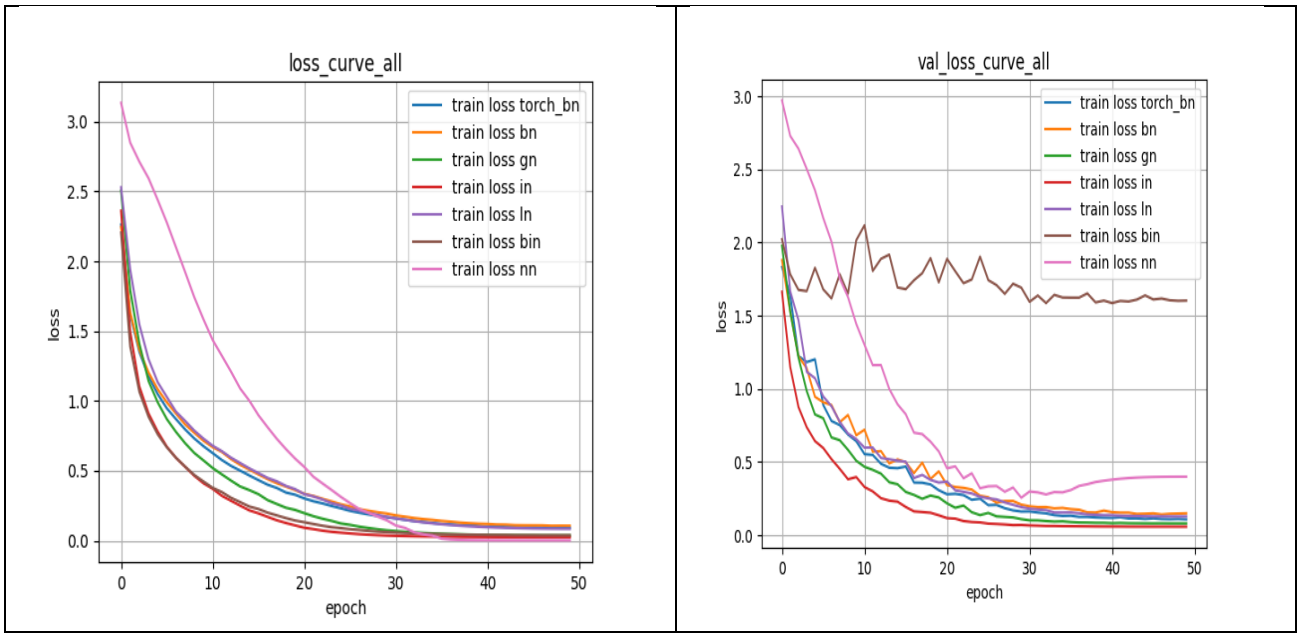


Fig 2: **comparison of Train and Validation loss curves of all normalization including pytorch inbuilt batch normalization**

- We are using 7 different types of normalization from which Batchnormalization (BN), Group normalization (GN), Instance normalization (IN), Layer normalization (LN), Batch instance normalization (BIN) are custom normalizations.
- For all the schemes, train loss showcase the similar behavior of decay, however, the rate of decay and final convergence is different.
- Initially train loss decay is faster for instance norm and batch instance norm followed by all the others. Decay of NN is the slowest among all.
- For all the normalizations, validation loss shows fluctuations in the beginning but after certain epochs it smooths out. In case of Batch instance norm, validation loss hardly converges, in case of NN, validation loss increases after certain epoch show casing the need of early stopping.
- BIN tends to overfit the data.

2.3 **performance comparison of all Normalizations**

	Training			Validation			Test		
	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1
Torch BN	98.52%	98.52%	98.52%	96.55%	97.35%	96.55%	91.66%	81.81%	91.66%
Custom BN	98.17%	98.17%	98.17%	96.55%	96.55%	97.49%	75%	75%	63%
NN	99.98%	99.98%	99.98%	100%	100%	100%	75%	75%	61.11%
GN	99.97%	99.97%	99.97%	100%	100%	100%	83.33%	83.33%	80%
LN	99.57%	99.57%	99.5%	100%	100%	100%	58.33%	58.33%	58%
IN	100%	100%	100%	100%	100%	100%	66.6%	66.6%	66%
BIN	99.96%	99.96%	99.96%	58.62%	58.62%	54.86%	33%	33%	21%

Table 2: performance results of models trained with different normalization schemes

- From the table above we can infer that Torch BN performs better compared to other normalizations followed by Group normalization.
- There are variations in validation accuracy and test accuracy of LN with great margin compared to other norms. BIN has high train accuracy but low validation and test accuracy which goes to show that BIN is overfitting the data.

3. **Impact of Batch size on BN and GN**

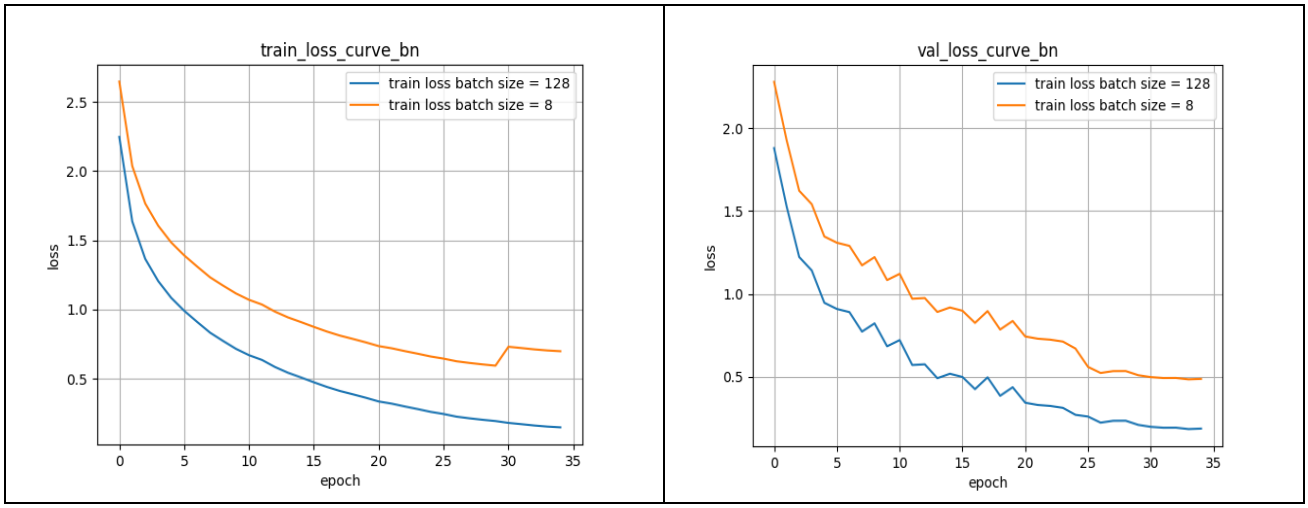


Fig 3: Train and val loss curves for custom batch normalization for batch 8 and 128

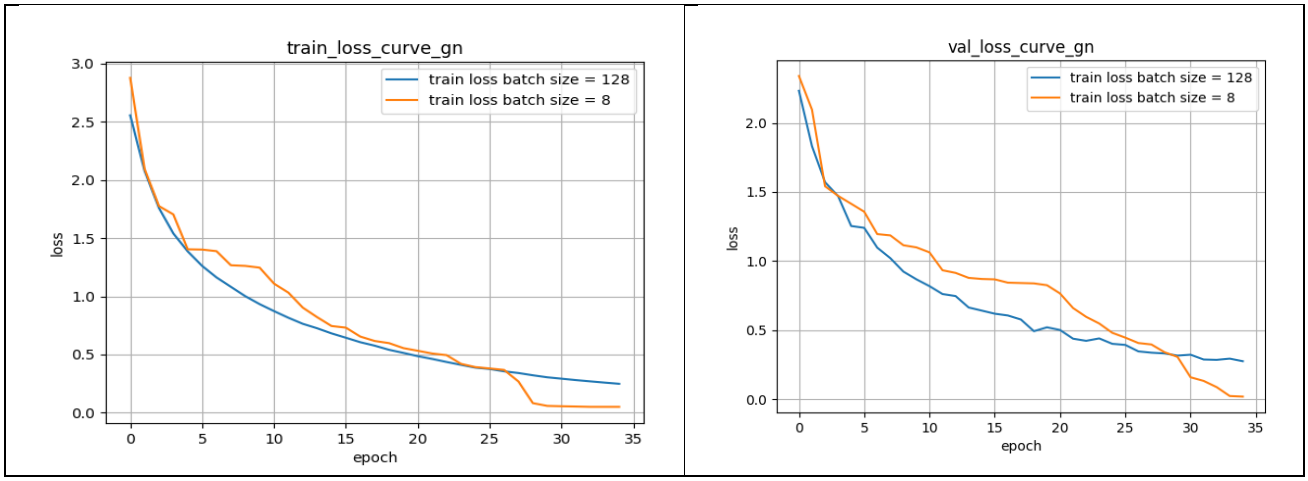


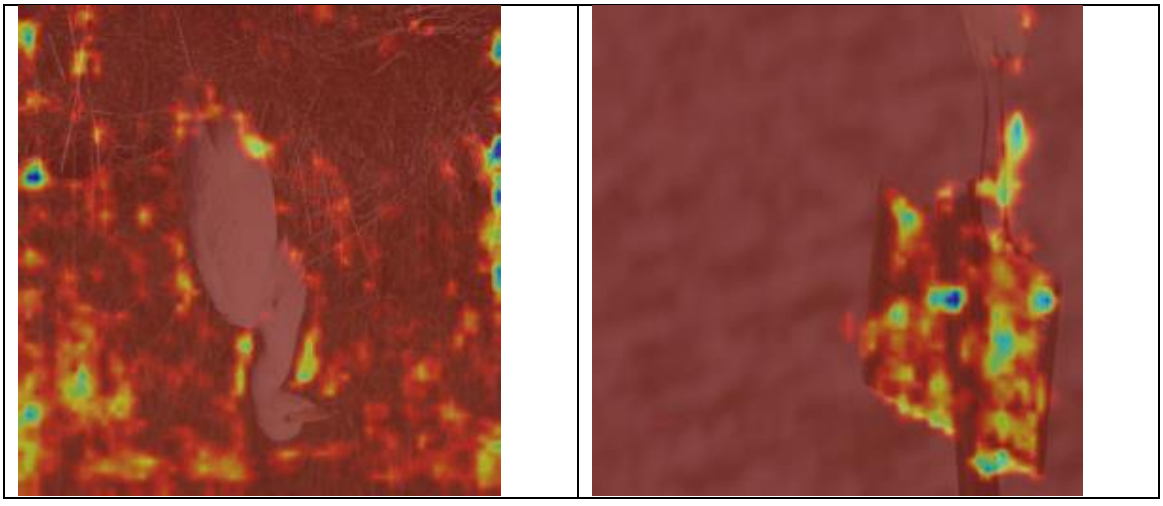
Fig 4: Train and val loss curves for custom group normalization for batch 8 and 128.

- From fig 4, we can see that there is little impact of batch size on the GN, which validates the claim of [Wu and He, 2020].
- Early stopping was applied in this part and training had been conducted till the epoch 35.
- In case of BN, train and validation loss curve of batch size 128 decreases faster than of batch size 8 while in GN, both training and validation loss curves for batch size 8 and 128 decreases at the same rate initially and in the later part the loss of batch size 8 decreases below the curve of batch size 128.

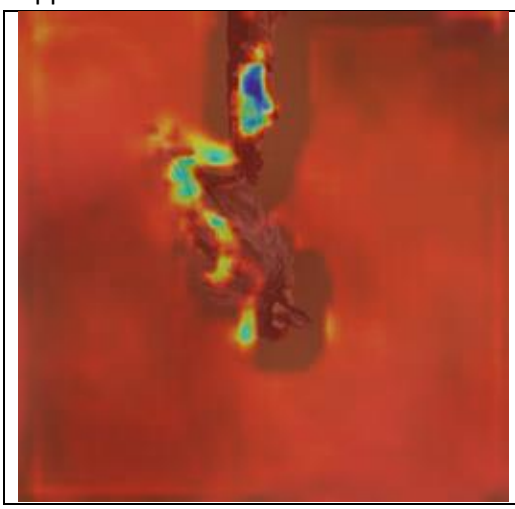
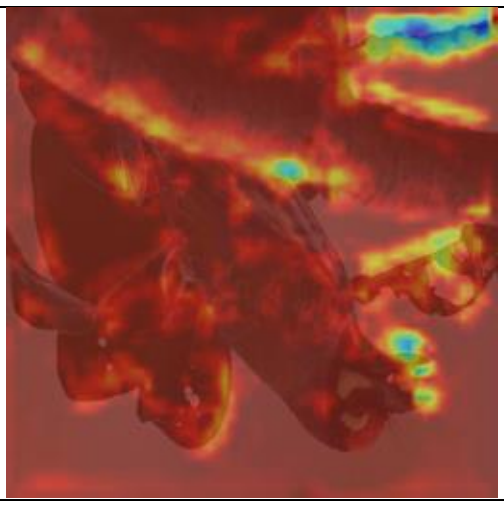
4. Gradient visualization through Grade – CAM

- Grade CAM (Grad-CAM), short for Gradient-weighted Class Activation Mapping, is a technique used to visualize and understand the decision-making process of a convolutional neural network (CNN). It helps to identify which regions of an input image are crucial for the model's prediction of a particular class.
- In our case we are Considering following 7 classes for visualization: Cattle Egret , Coppersmith Barbet, Indian Peacock, Red Wattled Lapwing, Ruddy Shelduck, White Breasted Kingfisher, White Breasted Waterhen.
- Below are the images of correctly classified and wrongly classified images from each class.

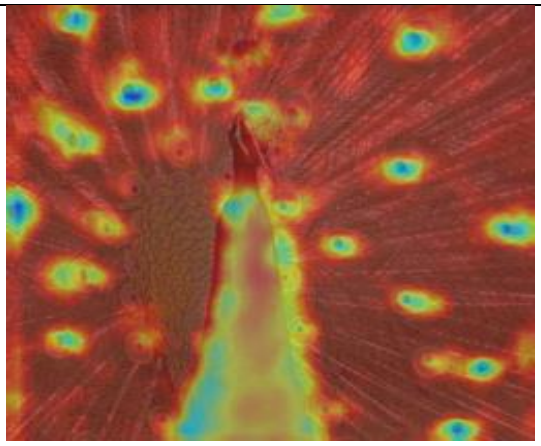
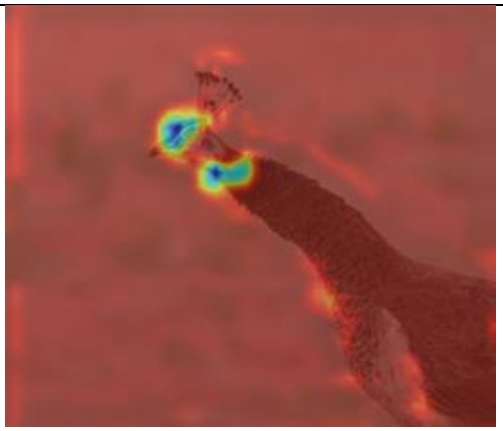
1. Cattle Egret



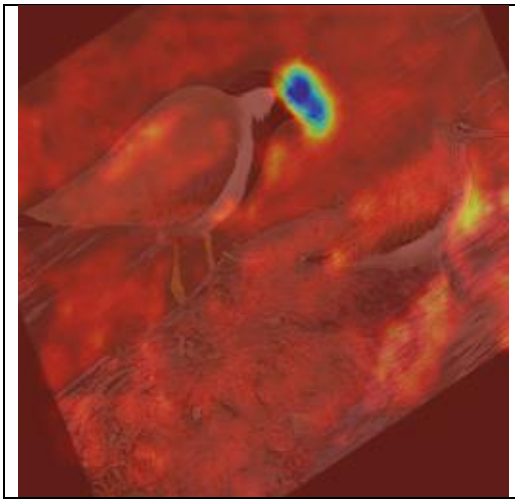
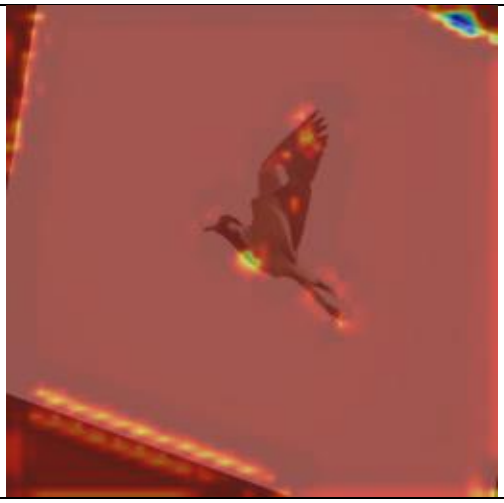
2. Coppersmith Barbet

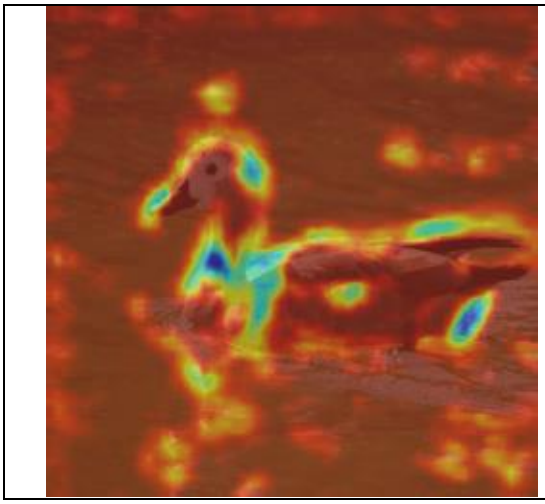
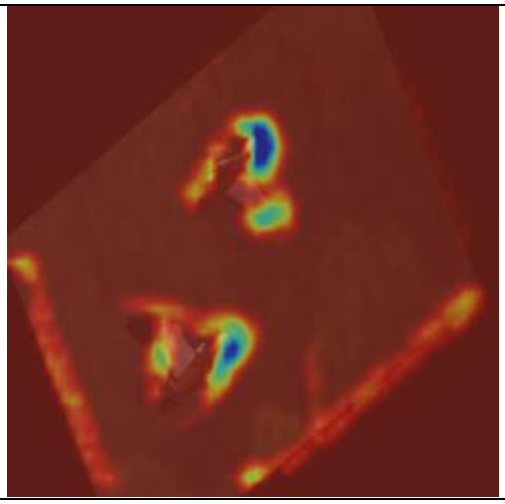
3. Indian Peacock

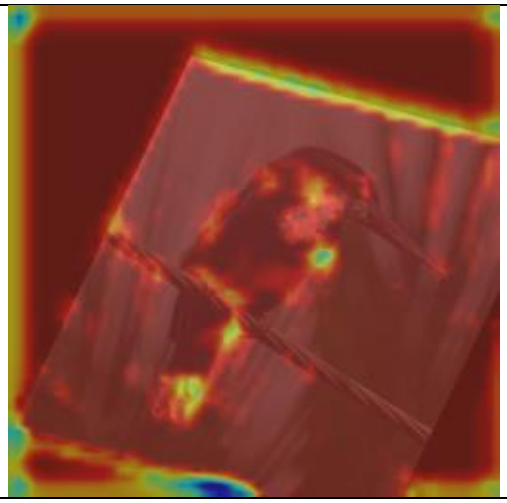
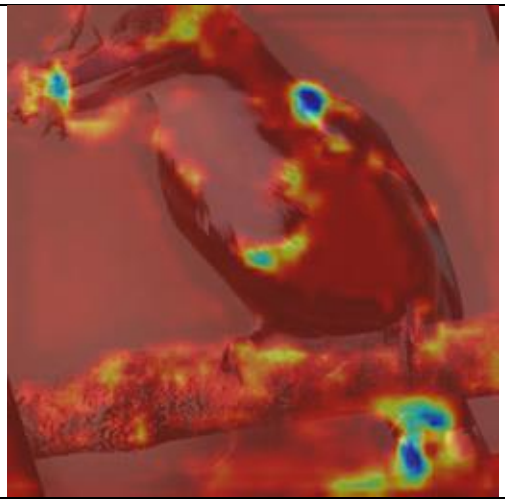
4. Red Wattled Lapwing

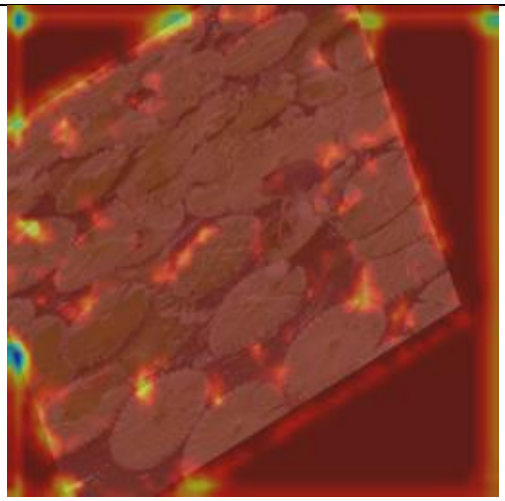
5. Ruddy Shelduck

6. White Breasted Kingfisher

7. , White Breasted Waterhen

- Images above shows the area of picture that our model chooses for the classification task for each class in other words the area that has the highest change in gradients are shown in heat map.

5. Model URL - https://drive.google.com/drive/folders/1A3wvSJxg0pS9O4Ywtv4W_1zLTPojIn1l?usp=sharing