

SEASONS OF CODE PROJECT **REPORT**

PREPARED BY: TIYASH DUTTA

ROLL NUMBER: 23B1544

DEPARTMENT: ENERGY SCIENCE AND
ENGINEERING

COURSE NAME: DEEP LEARNING

MENTOR NAME: SHUBHAM RAJ

PROJECT TOPIC: CONVOLUTIONAL NEURAL
NETWORK

CNN MODEL FOR IMAGE CLASSIFICATION OF BIRDS

- The dataset is available here: [Data](#)
- It is a very large dataset, so it will get downloaded as a zipped file. Unzip this zipped file
- Import the necessary libraries (TensorFlow)
- The image folder contains 200 subdirectories (each of which contains some image files), and these are the number of classes. The image size should be (256, 256) and the dataset should be shuffled.
- Map the data by dividing all the pixel values by 255, so that the pixel values range from 0 to 1.
- Split the image dataset as follows: 70% train, 20% test and 10% validation
- Model Architecture:
Sure! Here's the architecture of your CNN model described in points:

1. Input Layer:

- The input shape is (256, 256, 3) although explicitly not mentioned here.

2. Convolutional Layer:

- Layer Type: Conv2D
- Number of Filters: 8
- Kernel Size: 3x3
- Strides: 1
- Activation Function: ReLU

3. Batch Normalization:

- Normalizes the output of the previous layer to improve training stability and performance.

4. Pooling Layer:

- Layer Type: MaxPooling2D
- Pool Size: 2x2

- Reduces the spatial dimensions (width and height) by taking the maximum value in each 2x2 window.

5. Flatten Layer:

- Flattens the 2D matrix to a 1D vector to prepare it for the fully connected layers.

6. Fully Connected Layer (Dense Layer):

- Units: 8
- Activation Function: ReLU

7. Batch Normalization:

- Normalizes the output of the previous dense layer.

8. Output Layer:

- Layer Type: Dense
- Units: 200 (corresponding to the number of output classes)
- Activation Function: SoftMax (used for multi-class classification)
- Regularization: L2 regularization with a penalty factor of 0.01 to prevent overfitting

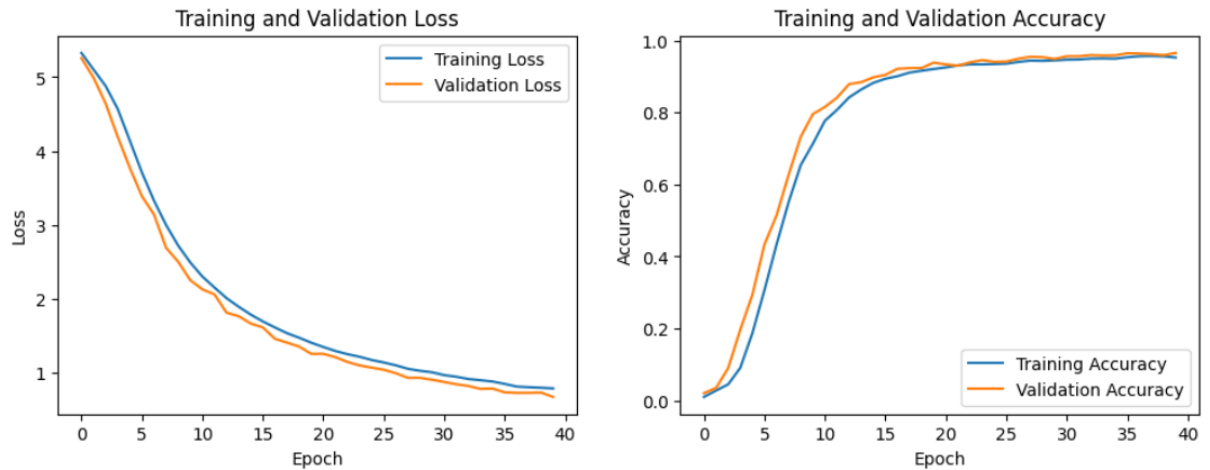
9. Compilation:

- Optimizer: Adam
- Loss Function: Sparse Categorical Crossentropy (suitable for multi-class classification with integer labels)
- Metrics: Accuracy

- **Training details:**

1. Number of epochs = 40
2. Ratio of train:test:val = 7:2:1
3. Train accuracy at the last epoch = 95.45%
4. Val accuracy at the last epoch = 96.58%
5. Test accuracy = 96.47%
6. Time taken per epoch = 24 seconds (mode time)

- Graphs:



- Summary of model:

Model: "sequential_13"

| Layer (type) | Output Shape | Param # |
|---|---------------------|-----------|
| conv2d_35 (Conv2D) | (None, 254, 254, 8) | 224 |
| batch_normalization_30 (BatchNormalization) | (None, 254, 254, 8) | 32 |
| max_pooling2d_29 (MaxPooling2D) | (None, 127, 127, 8) | 0 |
| flatten_13 (Flatten) | (None, 129032) | 0 |
| dense_26 (Dense) | (None, 8) | 1,032,264 |
| batch_normalization_31 (BatchNormalization) | (None, 8) | 32 |
| dense_27 (Dense) | (None, 200) | 1,800 |

Total params: 3,102,994 (11.84 MB)

Trainable params: 1,034,320 (3.95 MB)

Non-trainable params: 32 (128.00 B)

Optimizer params: 2,068,642 (7.89 MB)

- Checkpoint: [Final model checkpoint](#)
- Github link: [Link](#)