Bird Images Classification Using Convolutional Neural Network

Runyi Liu UID: 506296154
AOS C204 final project

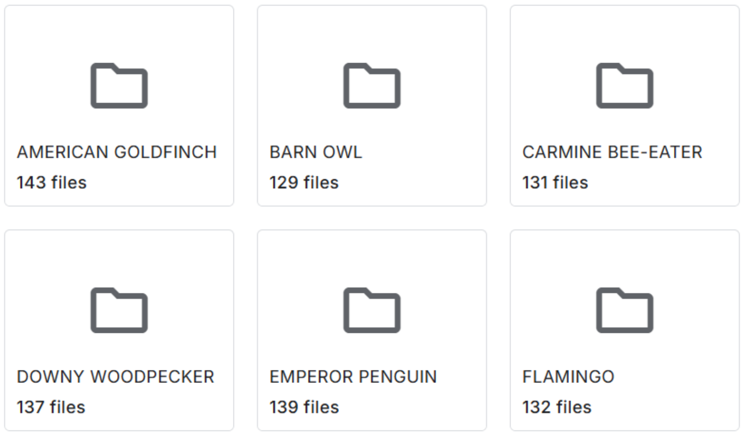
11/24/2024

1 Introduction

Automatic image classification can be applied to many fields, e.g., wildlife species research. Here we want to classify different bird species in different images. Given that convolutional neural networks (CNN) are powerful in this task, we constructed a small-size CNN model that can be operated well in personal computational devices. We concluded this model can classify different images in our dataset.

2 Data

Here is an overview of the dataset. This dataset contains 6 species of birds: (1) American Goldfinch, (2) Barn Owl, (3) Carmine Bee-eater, (4) Downy Woodpecker, (5) Emperor Penguin, (6) Flamingo. Figure 1 shows the species and their corresponding image numbers. This dataset can be found online via https://www.kaggle.com/datasets/rahmasleam/bird-speciees-dataset.


AMERICAN GOLDFINCH
143 files

BARN OWL
129 files

CARMINE BEE-EATER
131 files

DOWNY WOODPECKER
137 files

EMPEROR PENGUIN
139 files

FLAMINGO
132 files

Figure 1: Overview of bird species and their image numbers.

The images in each folder are in jpg format with a size of 224×224. Figure 2 shows an image of American Goldfinch in the dataset.

The image size like Figure 2 is (224, 224, 3) where 3 comes from 3 channels in RGB picture. We first rescale all values in all pixels from 0-255 to 0-1 for coming gradient descent calculation efficiency. We convert the 6 species of birds to 6 labels (0-5). We randomly choose images from the dataset and construct 3 sets: (1) training set (70%), (2) validation set (10%), (3) test set (20%). For the training set, we apply the random rotation to images up to 20% of a whole rotation from its original orientation. Figure 3 shows the data preprocessing and model structure in coding. Codes are written in Python, using the TensorFlow package, and run in Google Colab.

Figure 2: An image of American Goldfinch in the dataset.

3 Model

The model consists of a 2-D convolution layer, a batch normalization layer (batch size = 32), an activation function of ReLU, then a MaxPooling. In the end, all data was converted to a 1-D array, using dropout to randomly discard 50% of neural connections to connect with the softmax function to give the probability that the model predicts for each category. The predicted category is the one with the highest probability.

We use categorical cross-entropy as the loss function and Stochastic gradient descent (SGD) as the optimizer with a learning rate of 0.005. We set the maximum epochs to be 50, but the model has an early stopping algorithm that stops the iteration if the loss function doesn't decrease in 5 epochs in the validation set. This algorithm also returned the model parameters that gave the lowest loss function value on the validation set.

4 Results

Figure 4 shows the accuracy and loss function value calculated for both the training and validation sets. The training process stops at epoch 20, where there is no decrease in loss function value on the validation set in the past 5 epochs. Then, the model predicts the labels for the test set, which gives an accuracy of 82.8 %. Figure 5 shows the confusion matrix for the test set. The vertical label is the actual category and the horizontal label is the predicted category. Correct predictions lie in the diagonal of this matrix. Label 5 (Flamingo) image prediction has the lowest accuracy among all labels (bird species).

5 Discussion

In Figure 4, we can see strong accuracy and loss function fluctuations for the validation set. Here are some potential reasons. First, the validation is only 10 percent of the whole dataset. If we increase the size of the validation set, the fluctuations may decrease. However, this will lead to a decrease in the size of the

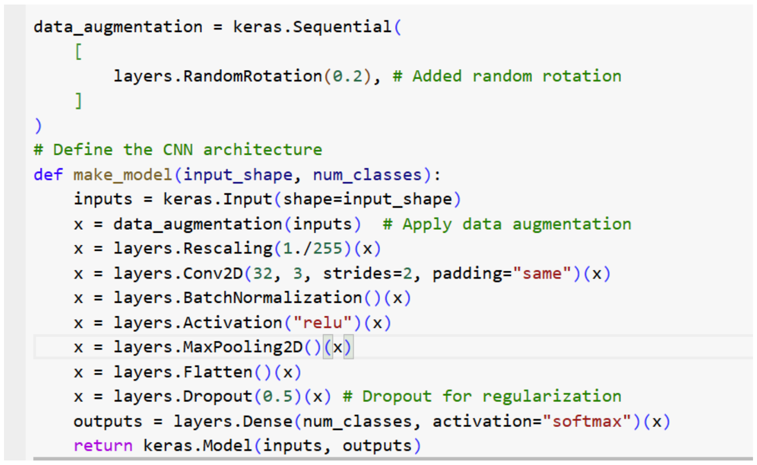

```
data_augmentation = keras.Sequential(
    [
        layers.RandomRotation(0.2), # Added random rotation
    ]
)
# Define the CNN architecture
def make_model(input_shape, num_classes):
    inputs = keras.Input(shape=input_shape)
    x = data_augmentation(inputs)  # Apply data augmentation
    x = layers.Rescaling(1./255)(x)
    x = layers.Conv2D(32, 3, strides=2, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)
    x = layers.MaxPooling2D()(x)
    x = layers.Flatten()(x)
    x = layers.Dropout(0.5)(x) # Dropout for regularization
    outputs = layers.Dense(num_classes, activation="softmax")(x)
    return keras.Model(inputs, outputs)
```

Figure 3: Data preprocessing and model structure.

test set, which adds new problems, e.g., an imbalance of data. Second, we choose SGD as the optimizer, which cannot adjust the learning rate by itself. A potential algorithm to adjust the learning rate here will be helpful. We tried Adam as an optimizer that can adapt the learning rate, but it may lead to a local minimum.

The training time could be long, especially when choosing a smaller learning rate or a larger model. In the preprocessing of the dataset, converting RGB images into gray ones can decrease 3 channels into 1, which reduces the data size. However, this may lead to worse training and following prediction. One reason is the color itself is a characteristic of each bird species.

The role of data augmentation is to prevent overfitting here potentially. We choose image rotation here due to its physical meaning: the bird's orientation can vary in each image. So, the random rotation in the data preprocessing part can add generality to the training set for better learning in the coming training process. Other data augmentation, e.g., translation, may negatively affect model performance because the translation of images can lead to missing characteristics.

6 Conclusion

The CNN model presented in the model section can classify 6 species of birds in our dataset with an accuracy of 82.8% on the test set. This shows potential application in bird image classification.

Future improvements can focus on adding more species of birds into the dataset to make the model more generalized. Moreover, in reality, lots of images contain more than 1 bird species, so the idea of object recognition should be considered.

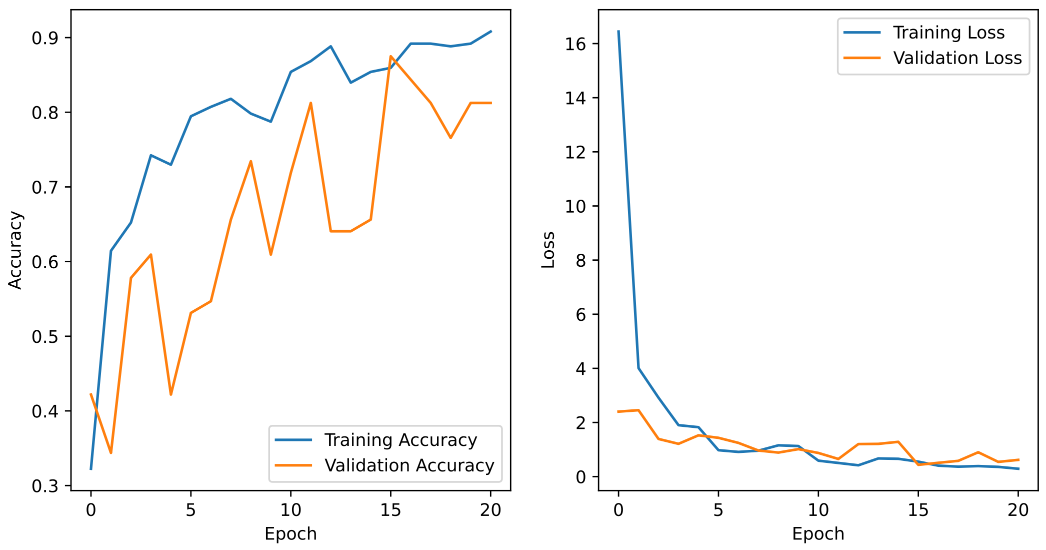
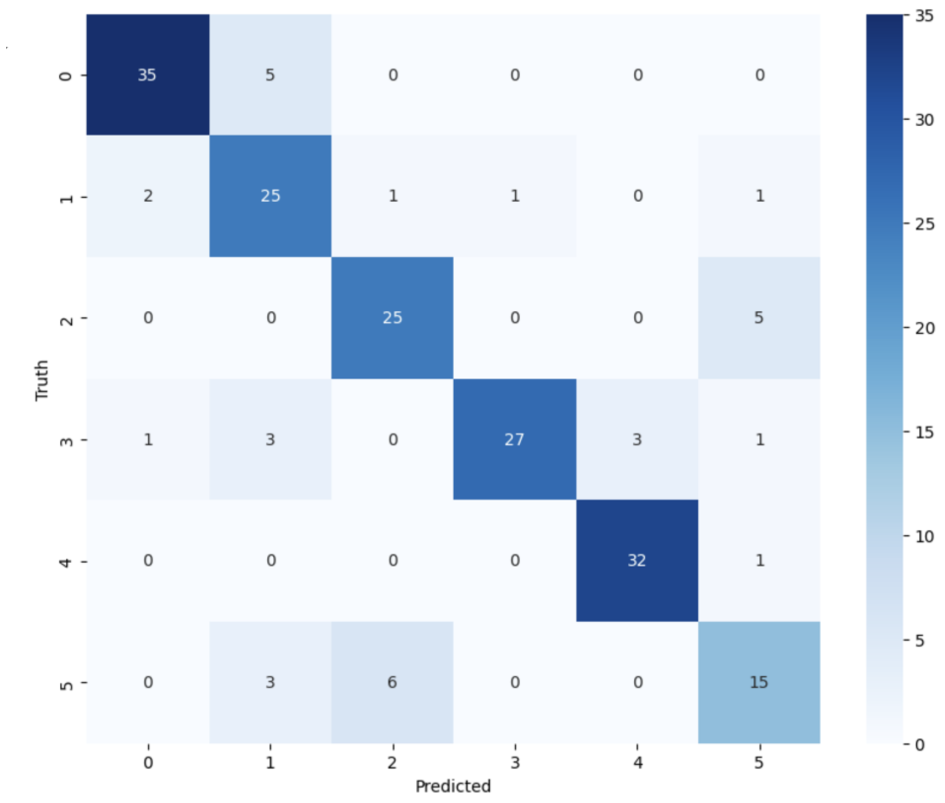


Figure 4: Accuracy and loss function value for the training set and validation set in the model training process.



	0	1	2	3	4	5
0	35	5	0	0	0	0
1	2	25	1	1	0	1
2	0	0	25	0	0	5
3	1	3	0	27	3	1
4	0	0	0	0	32	1
5	0	3	6	0	0	15

Figure 5: Confusion matrix of the predicted results on test set by the model.