Object Recognition and Computer Vision
Bird image classification

Laurent LAM
laurent.lam@ensta-paris.fr

1. Methodology

1.1. Resampling

The provided validation set did not seem to represent accurately the difficulty of the task since these images were actually clearer with more distinguishable bird features than those from the testing set. To start from scratch, I merged the given training and validation split before randomly resampling with a 10% ratio validation split.

1.2. Region of Interest Extraction

In order to capture only the bird features from the images and leave out the unnecessary background, I performed a Regions of Interest (ROIs) extraction using an Object Detector model.

For this task, I used the Mask-RCNN[1] recent architecture model, pre-trained on the COCO dataset, as it already contains a bird category, in order to infer the ROIs containing the birds on our dataset.

1.3. Data Augmentation

To obtain a high-performing model even when presented with ambiguous, noisy or unclear situations, the model's robustness needed to be improved with such data samples during training.

Therefore I applied data augmentation to increase the training set for the image classification task, using various geometric transformations, noise or kernel filters.

1.4. Image classification

1.4.1 Input

All images have been resized to 500x500 in order to have a common shape along the dataset.

Other more complex solutions have been considered such as resizing while keeping the aspect ratio and padding the remaining borders with black pixels or using a Super-Resolution model to improve the image quality after cropping.

However the results for both approaches did not lead to improvements on the accuracy score.

1.4.2 Architecture - Transfer Learning

Image classification was performed using the EfficientNet[2] state-of-the-art architecture, pre-trained on the ImageNet dataset. The optimal results were obtained by using the B7 variant of the architecture.

After loading the EfficientNet backbone and pre-trained weights, the original top layers used for the ImageNet task have been replaced with a Global Average Pooling layer, a Batch Normalization layer, a Dropout layer and a Dense layer along with a softmax activation function.

1.4.3 Two-stage training - Fine-tuning

The training of the model was performed via a two-stage training, using an Adam optimizer, a sparse categorical cross-entropy loss function and its corresponding accuracy metric.

The first stage of the training consisted in freezing the weights from the pre-trained layers leaving only the randomly initiated layers trainable. This stage aims at the convergence of these layers' weights so that the backpropagation during the second stage would use stabilized weights along the architecture.

The second training stage allows a part of the pre-trained highest layers to be unfrozen (except Batch Normalization layers) during training so that the model can be fine-tuned on our particular task. The learning rate for fine-tuning must be lower for this stage.

2. Results

This pipeline provided at best 84.516% accuracy on the 30% public test set, while providing stable accuracy results between 81-84%.

References

[1] Piotr Dollár Ross Girshick Kaiming He, Georgia Gkioxari. Mask r-cnn. *arXiv*, 2017. 1

[2] Quoc V. Le Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv*, 2019. 1