MVA RecVis 2021 Assignment 3: Bird image classification

Chloé Sekkat
ENSAE and ENS Paris-Saclay
chloe.sekkat@ensae.fr

Abstract

This assignment's goal is to classifify accurately 20 classes of birds. Our dataset is made of 1702 images which are a subset of the Caltech-UCSD Birds-200-2011 bird dataset.

1. Data preprocessing

1.1. Data distribution

The data is divided in train, val and test sets with respectively 1082, 103 and 517 images. The 20 classes are relatively balanced in the train set but not so much in the validation set. I merged train and validation sets to use cross validation to evaluate our models.

1.2. Data augmentation

I used data augmentation techniques on the training set such as Resize to 299x299, RandomHorizontalFlip, RandomRotation, ColorJitter and RandomPerspective.

1.3. Bird detection

By looking at random images from training and test sets, one can see that we have a huge variation in terms of where the bird is on the image, at which scale and which depth. This prevents our baseline algorithm (section 2.1) to learn meaningful features and to focus (heatmap) on the part of the image where the bird is. Therefore I use a pretrained Mask R-CNN on COCO with a ResNet+FPN backbone (implemented in the Detectron2 [1]). All images are cropped using the detected birds bounding box (if no bird is detected, the image is not modified). With a detection threshold of 0.7 and a NMS threshold of 0.4, respectively 1.66%, 2.91% and 3.48% of images in the original train, val and test sets are not cropped (no birds detected).

2. Models

2.1. Baseline with low data preprocessing

The first model created was a simple EfficientNetB7 pretrained on ImageNet where the last layer was modified to output 20 classes. The model was evaluated on 5 folds without cropping the image using the Mask R-CNN (see Table 2.1).

Model	Input	Preprocessing	Mask R-CNN	CV (accuracy)	Kaggle (accuracy)
EfficientNetB7	299x299	No	No	65.23	56.77
EfficientNetB7	299x299	**Yes**	No	69.01	58.06
EfficientNetB7	299x299	**Yes**	**Yes**	75.38	**65.80**

Table 1. Baseline models comparison

2.2. Classification using embeddings

I tried to convert each image into a 2048 dimensions feature vector using InceptionV3 and/or ResNext101 models and then classify these embeddings using a simple 3-layers MLP. Each embedding was extracted using preprocessed and cropped data. However the results were not better than the previous ones. I reached 91% of accuracy on 5 folds and 79.354% on Kaggle.

2.3. Classification on images

Using CV and cropped images by Mask R-CNN, I tried two other pretrained models (on ImageNet) **Inceptionv3 and ResNet152**, compared to EfficientNetB7, they gave slightly better results. My idea was then to combine them to provide more robust results. These two heads were merged together by a linear layer taking the outputs of the last linear layers of each model (embedding of dimension 512), concatenating them (dimension 1024) and outputing a probability on 20 classes. **Using all preprocessing techniques mentioned and Mask R-CNN, I reached a training accuracy (average of 5 folds) of 94% and 81.290% on Kaggle.**

3. Failed attempts

I tried many other techniques that either did not increase the accuracy or could not be completely tested due to time pressure. Among them were the Central Loss, Triplet Loss, Test Time Augmentation and Pseudo-labeling.

4. Conclusion

My best model is an Inceptionv3+ResNet152, with data preprocessing and cropping, SGD (lr=0.001, momentum=0.9, weight_decay=3e-4) and CosineAnnealingLR.

Model	Preprocessing & Mask R-CNN	CV (accuracy)	Kaggle (accuracy)
ResNet152	Yes	94.03	81.29
Inceptionv3 + ResNet152	**Yes**	93.45	**81.93**

Table 2. Best models

References

[1] Y. Wu and al. Detectron2. https://github.com/facebookresearch/detectron2. 1