

Deep Learning Framework - Keras 2.0

Other framework - OpenCV, Image Augmentation library (imgaug)

- pip install git+<https://github.com/aleju/imgaug>

Language - Python

X_train and X_train_crop - Training Augmented Images and Cropped Training Augmented Images file are quite large. (**~8-9 GB.**) If you need it specifically, then we can upload it on Google Drive link.

Mask R-CNN - Dependencies are uploaded in the folder.

1) Modify_data.py

- Create a folder of training_data
- Function: - **rename_files**:
 - It is used to rename image files sorted according to the number wise in the given train_data folder
 - Images are renamed as "Data Augmentation" + "Class" + "Number of image"
 - Example: 100105 denotes: 10 - Normal Image, 01- Class of Image, 05- Image number
 - Copyfile from source-train_data to destination-training_data

2) data_augmentation.py

- Data Augmentation of images and cropped images
- Type of Augmentation:
 - 10 - Normal Image
 - 20 - Gaussian Noise - $0.1 * 255$
 - 30 - Gaussian Blur - sigma - 3.0
 - 40 - Flip - Horizontal
 - 50 - Contrast Normalization - (0.5, 1.5)
 - 60 - Hue
 - 70 - Crop and Pad
 - Flipped
 - 11 - Add
 - 12 - Multiply
 - 13 - Sharpen
 - 14 - Gaussian Noise - $0.2 * 255$
 - 15 - Gaussian Blur - sigma - 0.0-2.0
 - 16 - Affine Translation 50px x, y
 - 17 - Hue Value
- Function: save_images -
 - Parameters -
 - Augmented_image - data augmentation done on the original image

- Destination - Location where to be saved
 - Number_of_images - number of images in a file
 - Classes - Class of image
 - Types - Each augmentation is denoted by its type as described above with numbers
- Original Images are taken and data augmented to reduce the effects of class imbalance

3) create_validation.py

- Generate Validation data from data augmented training data
- Function: - **create_validation**-
 - Files are moved from train folder to validation folder
 - Number of files moved is dependent on the number of images of each species.
 - If it is less than 78, then 6 images are chosen randomly for validation sample, if the number of images is between 78 and 81, then 8 images are chosen for validation data and if the number of images are greater than 85 then 9 files are chosen.
 - In total, from 1330 training images after augmentation, 133 images were chosen for validation. 1197 images for training.

4) gen_train_data_test_data.py

- Creates training, validation and testing numpy files for training, validation and testing
- Function: - **gen_data**
 - Train sample, validation sample, test sample - Location where training, validation and testing samples are placed
 - Each directory files are sorted in ascending order
 - Each for loop generates an image resized to 416x416 and then appended to X_train, X_valid, X_test accordingly. Y_train, Y_valid, Y_test contains labels of each files.
 - Numpy files are saved

5) inception_v3_finetime.py

- Inception V3 finetuning
- Functions:
 - Swish - Swish Activation function defined
 - Step_decay - Stepwise decay after 4 epochs
 - Precision - Calculation of precision using keras backend
 - Recall - Calculation of recall using keras backend
 - F1 - Calculated value of F1 from precision and recall
 - Build_inception_v3 - Fine-tuning on last 2 layers
 - Loss - categorical_crossentropy
 - Optimizer - Adam

- Metrics - precision, recall, f1
- X_train_crop, y_train_crop - Cropped images loading
- X_train, y_train, x_valid, y_vlaid, x_test, y_test = Dataset loading
- Learning rate decay and score evaluation

6) inception_resnet_v2_finetune.py

- Inception ResNet V2 finetuning
- Functions:
 - Swish - Swish Activation function defined
 - Step_decay - Stepwise decay after 4 epochs
 - Precision - Calculation of precision using keras backend
 - Recall - Calculation of recall using keras backend
 - F1 - Calculated value of F1 from precision and recall
 - Build_inception_v3 - Fine-tuning on last 2 layers
 - Loss - categorical_crossentropy
 - Optimizer - Adam
 - Metrics - precision, recall, f1
- X_train_crop, y_train_crop - Cropped images loading
- X_train, y_train, x_valid, y_vlaid, x_test, y_test = Dataset loading
- Learning rate decay and score evaluation

7) mask_r_cnn_crops.py

- Generates cropped images from original training data
- Mask_rcnn_coco.h5 - model weights loaded to detect birds in images
- Train images are loaded
- Model.detect detects birds ROIs in the images.
- Birds ROI are cropped from original and then cv2.imwrite saves the crops in the new folder

8) test_images.py - Main file for detection

- Mask R_CNN basic files are loaded
- Build_inception_v3 - Inception V3 model loaded
- Model_cropped_inception_v3 - Cropped Images Inception V3 model weights loaded
- Model_final_inception_v3 - Cropped+Original Images Inception V3 model weights loaded
- Build_inception_resnet_v2 - Inception Resnet V2 model loaded
- Model_inception_resnet_v2 - Cropped+Original Images Inception Resnet V2 weights loaded
- Each test image is passed through Mask R_CNN first. **If bird is detected**
 - If birds ROIs are detected then it is appended to batches and then Inception V3 and Inception resnet V2 prediction is done on batches

- The prediction values of each bird is then compared from Inception V3 and Inception Resnet V2 whosoever have maximum value of prediction that bird specie is finally appended to flip_final
- Bincounts of number of birds of in the images with lots of birds
- Final maximum number of specie of birds is appended to final prediction file
- **If bird is not detected:**
 - Whole image is passed through Inception V3 and Inception Resnet V2 weights and whosoever has higher prediction it is appended to final prediction file

9) evaluate.py -

- Ground truth and prediction file are loaded
- Precision, recall - Averaged over classes
 - Calculate True Positive, False Positive and False Negative for each class and then precision and recall for each class. Total sum of precision and recall is divided by 16 to get the average precision and average recall scores.
- F1 score - Averaged over classes - F1 score is calculated from the average precision and average recall obtained.