**Object detection for image cropping: Detailed documentation for Eli**

**Aves**

For additional information, see [github.com/aubricot/object\_detection\_for\_image\_cropping](https://github.com/aubricot/object_detection_for_image_cropping)

17 Feb 2020

Model used for final crops

YOLO

**Step 1) Object Detection**

Script(s)

aves\_yolo.ipynb

Input file(s)

https://editors.eol.org/other\_files/bundle\_images/files/images\_for\_Aves\_breakdown\_download\_000001.txt

https://editors.eol.org/other\_files/bundle\_images/files/images\_for\_Aves\_20K\_breakdown\_download\_000001.txt

Output file(s)

aves\_det\_crops\_1000.tsv

aves\_det\_crops\_20000.tsv

Detailed steps

Open aves\_yolo.ipynb. Run all cell blocks until “Load in sample images and 'flow' them through the object detector.” Under “A) Load individual images in by URL,” test the object detector with a few images to see that everything is built and running properly. Next, under “B) Load multiple images (from EOL image URL bundles) through object detector,” load in EOL image bundle (ex: <https://editors.eol.org/other_files/bundle_images/files/images_for_Aves_breakdown_download_000001.txt>). Export results to aves\_det\_crops\_1000.tsv. Note: for large numbers of images, can run detection in batches and export results to aves\_det\_crops\_20000\_a.tsv, aves\_det\_crops\_20000\_b.tsv, etc.).

**Step 2) Convert bounding boxes to square, centered image cropping coordinates and pad by 11%**

Script(s)

aves\_convert\_bboxdims.py

Input file(s)

aves\_det\_crops\_1000.tsv

aves\_det\_crops\_20000.tsv

images\_for\_Aves\_breakdown\_000001.txt

images\_for\_Aves\_20K\_breakdown\_000001.txt

Output file(s)

bird\_crops\_yolo\_1000img.tsv

bird\_crops\_yolo\_20000img.tsv

bird\_crops\_yolo\_1000img\_display\_test.tsv

bird\_crops\_yolo\_20000img\_display\_test.tsv

Detailed steps

For bounding box coordinates exported from Step 1 (ex: aves\_det\_crops\_1000.tsv), use aves\_convert\_bboxdims.py to remove out of bounds values, convert dimensions to square, pad squares by 11% image size, and re-format results to EOL crop coodinate formatting standards. Line 44 matches image\_url to other EOL image identifiers (dataObjectVersionIDs and identifier) using EOL breakdown file (ex: images\_for\_Aves\_breakdown\_000001.txt). Export square, padded cropping dimensions to display test file (ex: bird\_crops\_yolo\_1000img\_display\_test.tsv) for use in Step 3 below. Final crop dimensions are formatted to EOL cropping standards and exported (ex: bird\_crops\_yolo\_1000img.tsv).

Note: If you exported a large image dataset in batches (ex: aves\_det\_crops\_20000\_a.tsv, aves\_det\_crops\_20000\_b.tsv, etc.), you must first combine them using merge\_tsvs.py, then proceed with “Detailed steps” above.

**Step 3) Display converted cropping coordinates on images**

Script(s)

coordinates\_display\_test.ipynb

Input file(s)

bird\_crops\_yolo\_1000img\_display\_test.tsv

bird\_crops\_yolo\_20000img\_display\_test.tsv

Output file(s)

none

Detailed steps

Upload display test file to Google Drive (ex: bird\_crops\_yolo\_1000img\_display\_test.tsv). Open coordinates\_display\_test.ipynb and include the path to display test file. Then, display square, padded crop coordinates from Step 2 on images in Google Colab and verify that the transformations in aves\_convert\_bboxdims.py are appropriate (or to fine tune accordingly and return to Step 2, ex: adjust padding amount).