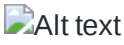


From wildlife.ai
For Wildlife Watcher project

AddaxAI for wildlife.ai

Download PDF

Report regarding using AddaxAI for MVP#1 as an annotation tool with images from WW500.
Why AddaxAI? Here is how and why AddaxAI could be useful in our worflow.

1. Images from WW500
2. Output from AddaxAI

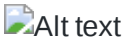
Images from WW500

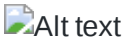
For this test, I used the WW500 and the firmware from GitHub code base branch `exif-app1`.
I recorded 5 images. Extract them from the SD cards and checked the EXIF information.

WW500 deployment folder structure

They have the following structure:
\begin{verbatim} .
└── images
├── image_0001_2025-01-02.jpg
├── image_0002_2025-01-02.jpg
├── image_0003_2025-01-02.jpg
├── image_0004_2025-01-02.jpg
└── image_0005_2025-01-02.jpg
\end{verbatim}

Example of image and exif information in it





Output from AddaxAI

Steps to get the output from AddaxAI

1. Open AddaxAI
2. Extract the target deployment folder from the SD card
3. Drag and drop the whole folder

Steps in AddaxAI

1. Select the folder containing your project on a CampTrap DP format
2. Run the analysis
3. (optionnal): Manually check the annotations
4. (optionnal): Get the post-processing done, which will create two things: `results.csv` and `graphs` folder

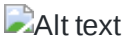
1. Drag and drop folder | 2. Run the analysis

Here I selected MegaDetector cause it was image of a human (aka me).


Alt text

3. Annotation | 4. Post processing

Here I selected the destination folder as `Deployment_0001/addaxAI_results`, a folder that I created before.
The output file format that I chose was `csv` (it's either excel or csv)


Alt text

Results

Result folder structure

\begin{verbatim} .
├── graphs
│ ├── maps
│ │ ├── class-specific
│ │ │ └── person.html
│ │ ├── combined-multi-layer.html
│ │ └── combined-single-layer.html
│ └── pie-charts
│ ├── distribution-detections.html
│ ├── distribution-detections.png
│ ├── distribution-files.html
│ └── distribution-files.png
├── results_detections.csv
├── results_files.csv
└── results_summary.csv
\end{verbatim}

In `results_summary.csv` you'll find a summary of the detections as a CSV file.

In `CamTrap_dataset/image_recognition_file.json` you'll have a summary of the detections as a JSON file.

In `results_detections.csv` you'll find for each image of interest, the result and the exif information read from `media.csv`.

In `graph` you'll find some charts about the analysis.

Content of `results_summary.csv`

label	data_type	n_detections
person	img	3

Sample of `CamTrap_dataset/image_recognition_file.json`

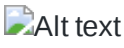

```
{
 "images": [
  {
   "file": "image_0001_2025-01-02.jpg",
   "detections": [
    {
     "category": "2",
     "conf": 0.557,
     "bbox": [
      0,
      0,
      0.9984,
      1.0
     ]
    }
   ]
  },
```


Example of `results_detections.csv`

absolute_path	relative_path	data_type	labe
C:/Users/Deva/Desktop/Wildlifeai/Deployment_0001/images	image_0001_2025-01-02.jpg	img	perso

Example of `graphs/maps/`

AddaxAI is making charts based on the EXIF info. On the CamTrap example, it did charts regarding temporal distribution, here we have spacial distribution and label distribution. On the map you can filter labels.



Example of `graphs/pie-charts/`