

MegaDetector 自動処理使用方法

0.環境構築

ImageExtractWin をインストールする。

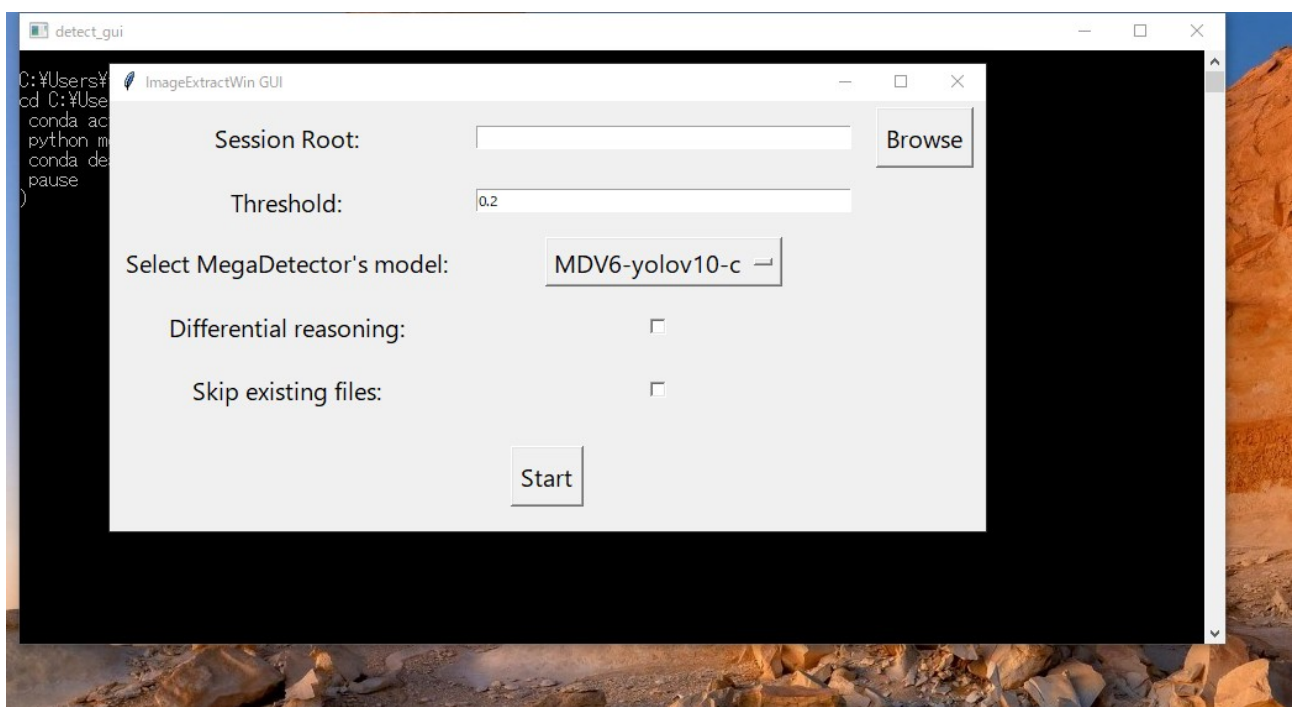
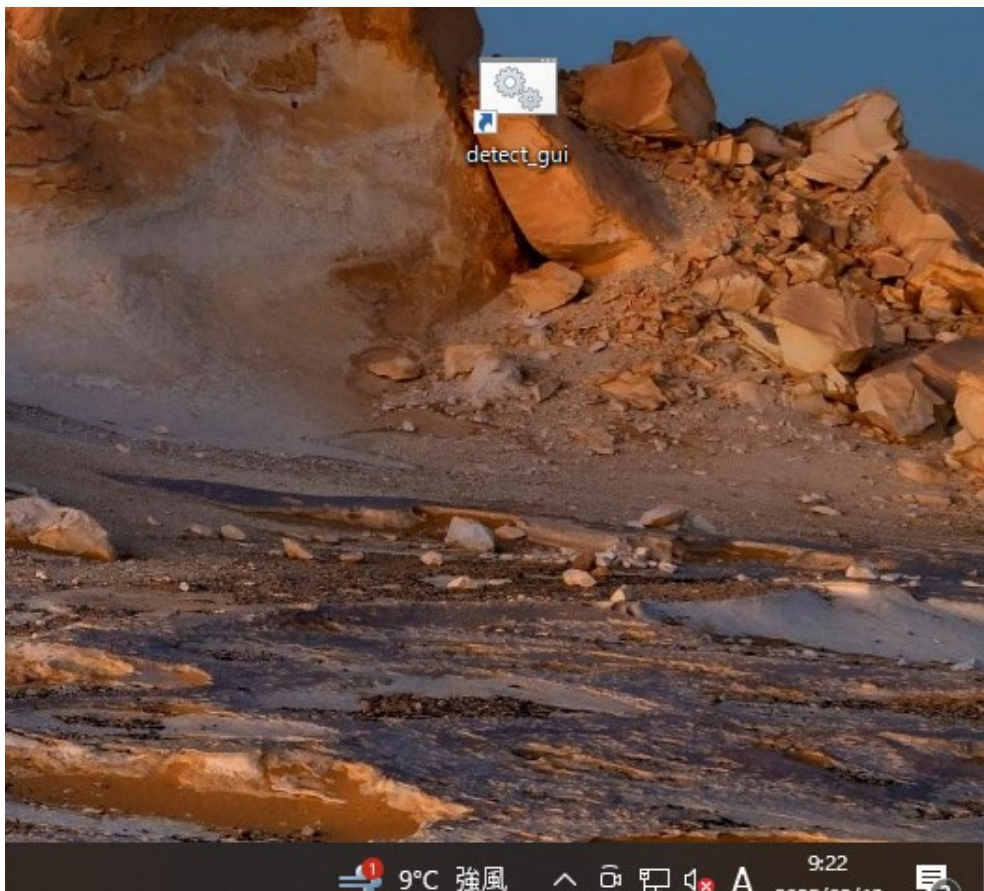
[yodaka0/ImageExtractWin](#)

The screenshot shows the GitHub repository page for `yodaka0/ImageExtractWin`. The repository has 121 commits and 0 stars. The README section is titled "ImageExtractWin" and "What's this : このプログラムについて". The README text describes the program's purpose: to detect wildlife from camera trap images using MegaDetector (Beery et al. 2019) and to extract images in which animals were detected. It also mentions that the document is a minimal description and will be updated as needed. The right sidebar shows repository statistics: 121 commits, 0 stars, 1 watching, 0 forks, and no releases or packages published. Suggested workflows for Python application, Python package, and SLSA Generic generator are also visible.

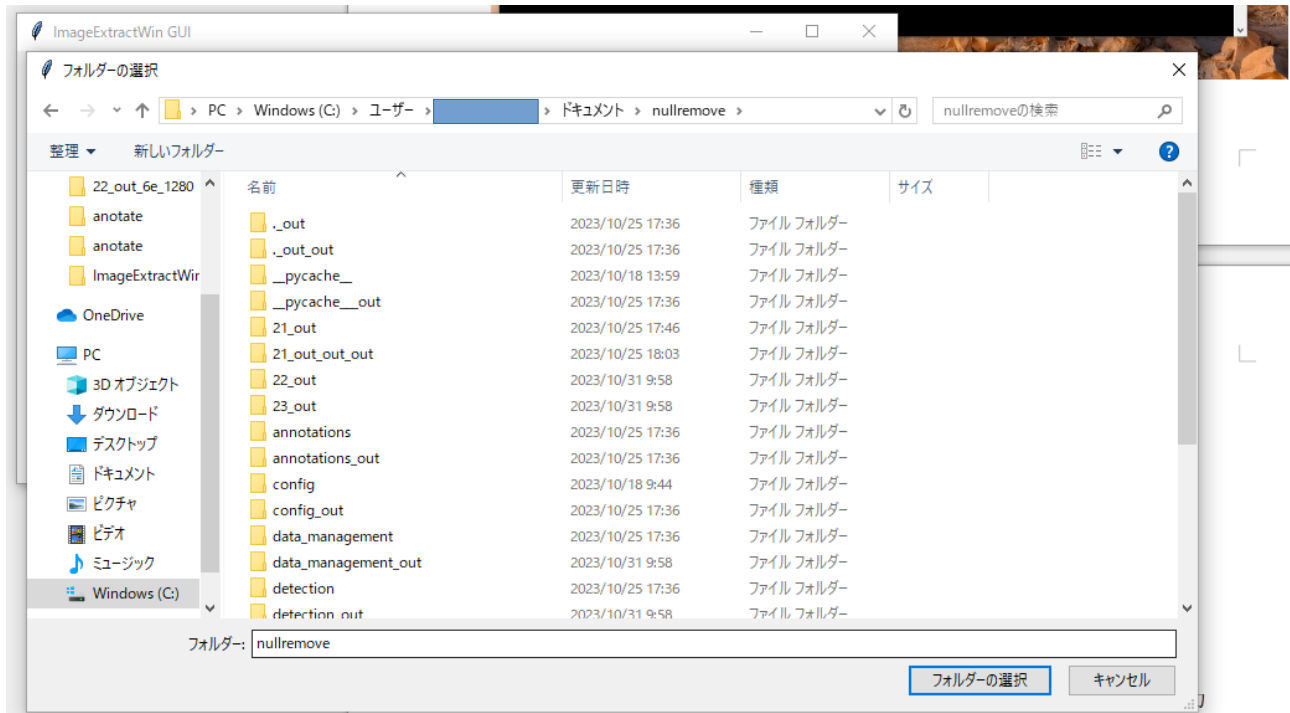
The screenshot shows the GitHub file viewer for the `mdet_setup.py` file in the `ImageExtractWin` repository. The file is 150 lines long (121 loc) and 5.36 KB. The code is a Python script that sets up the environment for running MegaDetector. The code shows imports for `os`, `subprocess`, `zipfile`, `platform`, `urllib.request`, `stat`, `shutil`, `requests`, and `ImportError`. The code is a Python script that sets up the environment for running MegaDetector.

```
>python mdet_setup.py
```

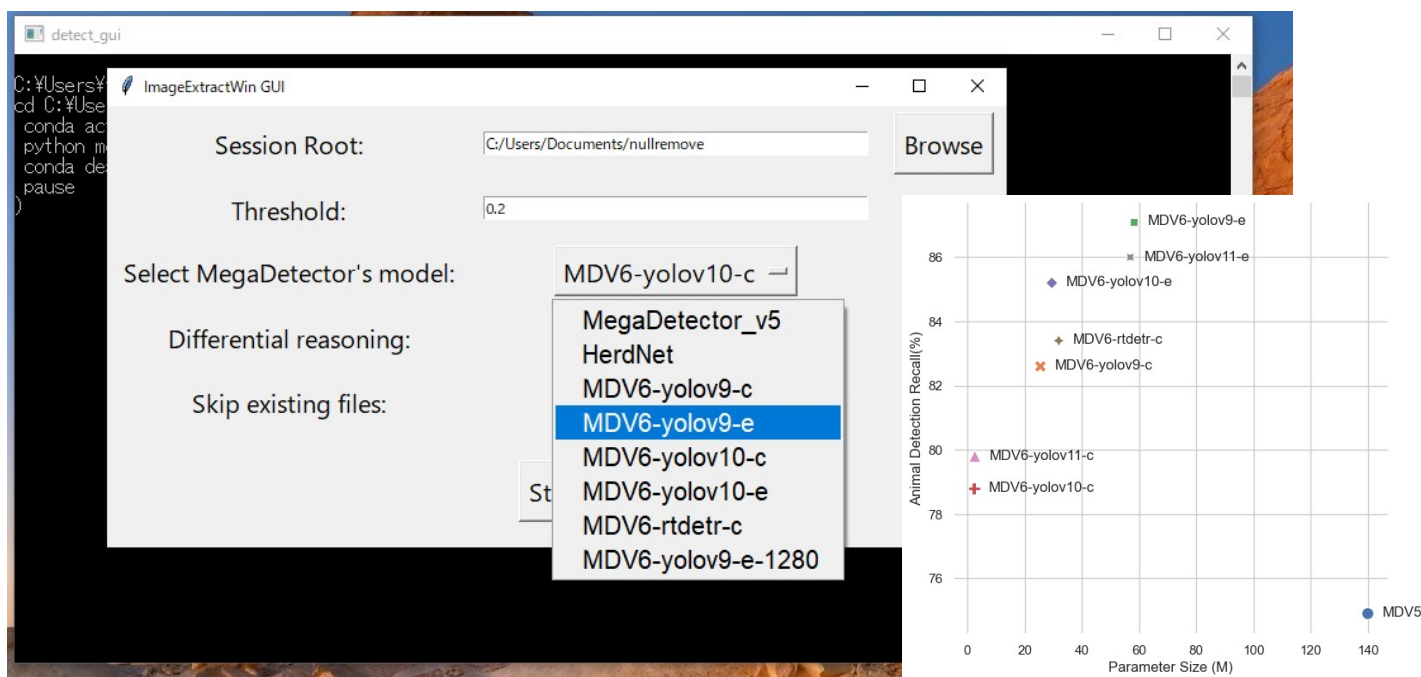
1. ショートカット「detect_gui」を開く



2. Brows を押して処理する画像の入ったフォルダを選択する



3. 検出モデルを選択する。



HerdNet(model designed for the accurate detection and counting of African mammals in aerial images)

4. その他の設定

threshold:検出の閾値 (0~1);

skip:既に処理済みファイルがある場合スキップする ;

differential reasoning:前の画像と同座標にある animal の検出を blank に変換する

5.Start で処理開始

```
process, 78.1ms inference, 0.0ms postprocess per image at shape (1, 3, 480, 640)
no/Documents/nullremove/ogawa.2020.12/22¥10040029.JPG has 0 animals
◆ Python-3.9.21 torch-2.6.0+cpu CPU (Intel Core(TM) i5-6500 3.20GHz)
(fused): 285 layers, 2,695,586 parameters, 0 gradients, 8.2 GFLOPs

ections), 79.7ms
process, 79.7ms inference, 0.0ms postprocess per image at shape (1, 3, 480, 640)
no/Documents/nullremove/ogawa.2020.12/22¥10040030.JPG has 0 animals
◆ Python-3.9.21 torch-2.6.0+cpu CPU (Intel Core(TM) i5-6500 3.20GHz)
(fused): 285 layers, 2,695,586 parameters, 0 gradients, 8.2 GFLOPs

l, 78.1ms
process, 78.1ms inference, 15.6ms postprocess per image at shape (1, 3, 480, 640)
no/Documents/nullremove/ogawa.2020.12/22¥10040031.JPG has 1 animals
◆ Python-3.9.21 torch-2.6.0+cpu CPU (Intel Core(TM) i5-6500 3.20GHz)
(fused): 285 layers, 2,695,586 parameters, 0 gradients, 8.2 GFLOPs

ections), 87.1ms
process, 87.1ms inference, 0.0ms postprocess per image at shape (1, 3, 480, 640)
no/Documents/nullremove/ogawa.2020.12/22¥10040032.JPG has 0 animals
◆ Python-3.9.21 torch-2.6.0+cpu CPU (Intel Core(TM) i5-6500 3.20GHz)
(fused): 285 layers, 2,695,586 parameters, 0 gradients, 8.2 GFLOPs

ections), 90.5ms
process, 90.5ms inference, 0.0ms postprocess per image at shape (1, 3, 480, 640)
no/Documents/nullremove/ogawa.2020.12/22¥10040033.JPG has 0 animals
◆ Python-3.9.21 torch-2.6.0+cpu CPU (Intel Core(TM) i5-6500 3.20GHz)
(fused): 285 layers, 2,695,586 parameters, 0 gradients, 8.2 GFLOPs
```

6.終了

”元フォルダ+_out”に処理結果が保存される。

