

# 使用方法

## 0.環境構築

ImageExtractWin をインストールする。

[yodaka0/ImageExtractWin](#)

The screenshot displays the GitHub repository page for `yodaka0/ImageExtractWin`. The repository is on the `master` branch, with 5 branches and 0 tags. The commit history shows a series of updates to various files, including `position.json`, `visualization`, `LICENSE.txt`, `README.md`, `classifier.py`, `environment.yml`, `exec_mdet.py`, `gpu_check.py`, `image_demo.py`, `image_separation_demo.py`, `make_batch_gui.py`, `mdet_gui.py`, and `mdet_setup.py`. The README file is selected, showing the title `ImageExtractWin` and the text `What's this : このプログラムについて`. The README content 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 the repository's statistics: 0 stars, 1 watching, and 0 forks. It also includes sections for Releases, Packages, Languages (Python 100.0%), and Suggested workflows.

**Repository Structure:**

File/Folder	Commit Message	Time Ago
<code>anotate</code>	Update position.json	1 minute ago
<code>visualization</code>	second commit	2 years ago
<code>LICENSE.txt</code>	Add files via upload	2 months ago
<code>README.md</code>	Update README.md	14 minutes ago
<code>classifier.py</code>	Add files via upload	4 months ago
<code>environment.yml</code>	Update environment.yml	2 weeks ago
<code>exec_mdet.py</code>	Update exec_mdet.py	4 minutes ago
<code>gpu_check.py</code>	Add files via upload	8 months ago
<code>image_demo.py</code>	Update image_demo.py	2 weeks ago
<code>image_separation_demo.py</code>	Add files via upload	7 months ago
<code>make_batch_gui.py</code>	Update make_batch_gui.py	last month
<code>mdet_gui.py</code>	Update mdet_gui.py	19 minutes ago
<code>mdet_setup.py</code>	Update mdet_setup.py	8 minutes ago

**README Content:**

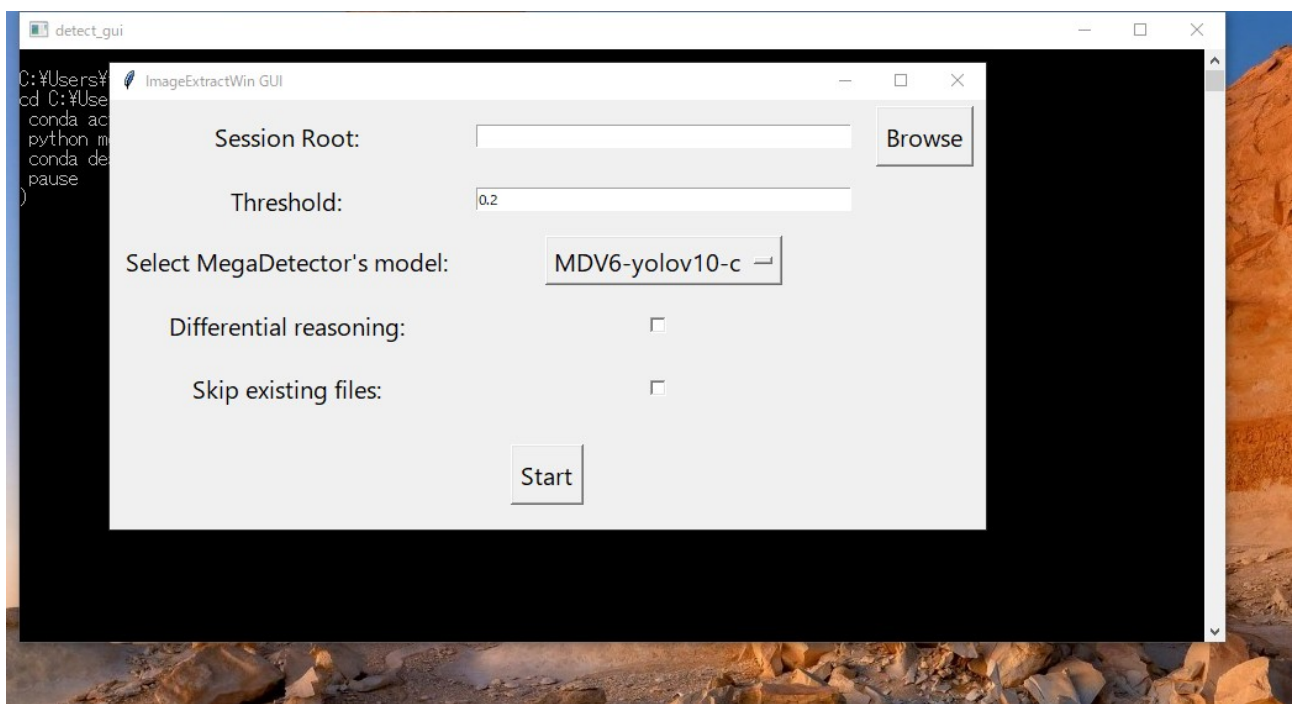
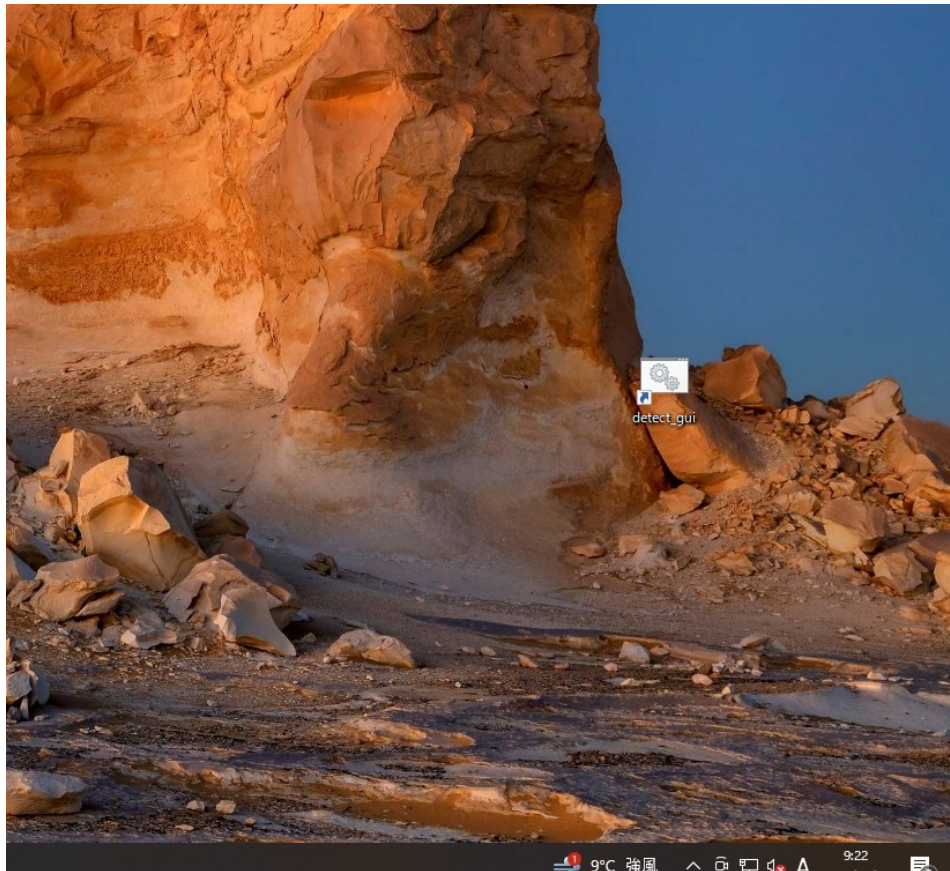
### ImageExtractWin

#### What's this : このプログラムについて

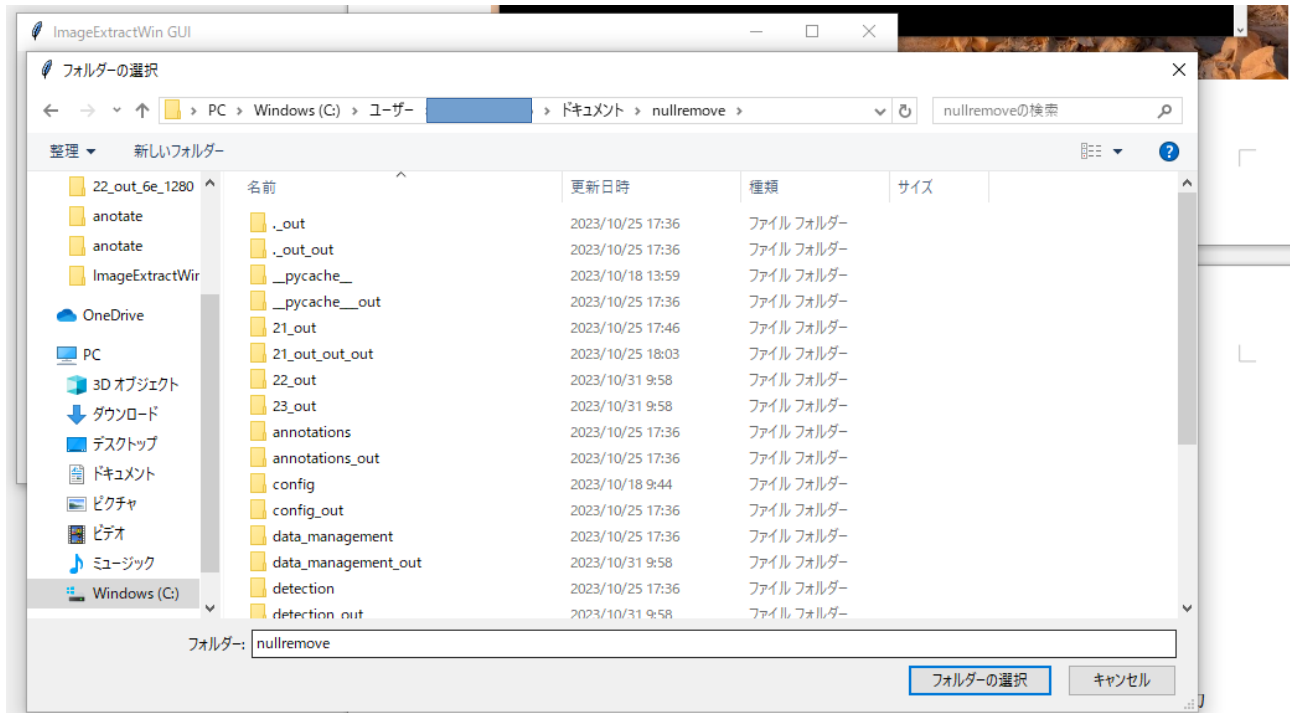
This program aims to detect wildlife from camera trap images using [MegaDetector](#) (Beery et al. 2019) and to extract images in which animals were detected. This document is a minimal description and will be updated as needed.

このプログラムは、[MegaDetector](#) (Beery et al. 2019)を利用してカメラトラップ映像から野生動物を検出し、動物が検出された画像を抽出することを目的として作成されました。このドキュメントは現時点では最低限の記述しかされていないため、今後随時更新していく予定です。

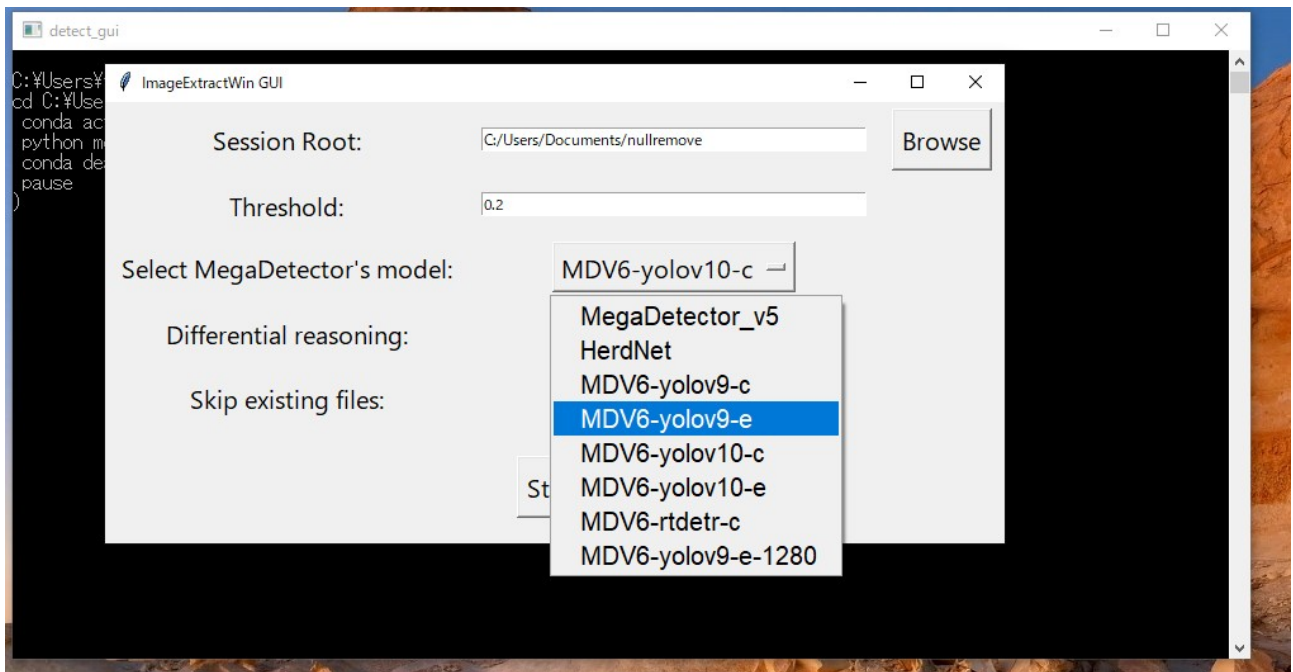
## 1. ショートカット「detect\_gui」を開く



## 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。



## 4. その他の設定

threshold:検出の閾値 ; skip:既にファイルがある場合スキップする ; differential reasoning:  
前の画像と同じ位置の 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。

