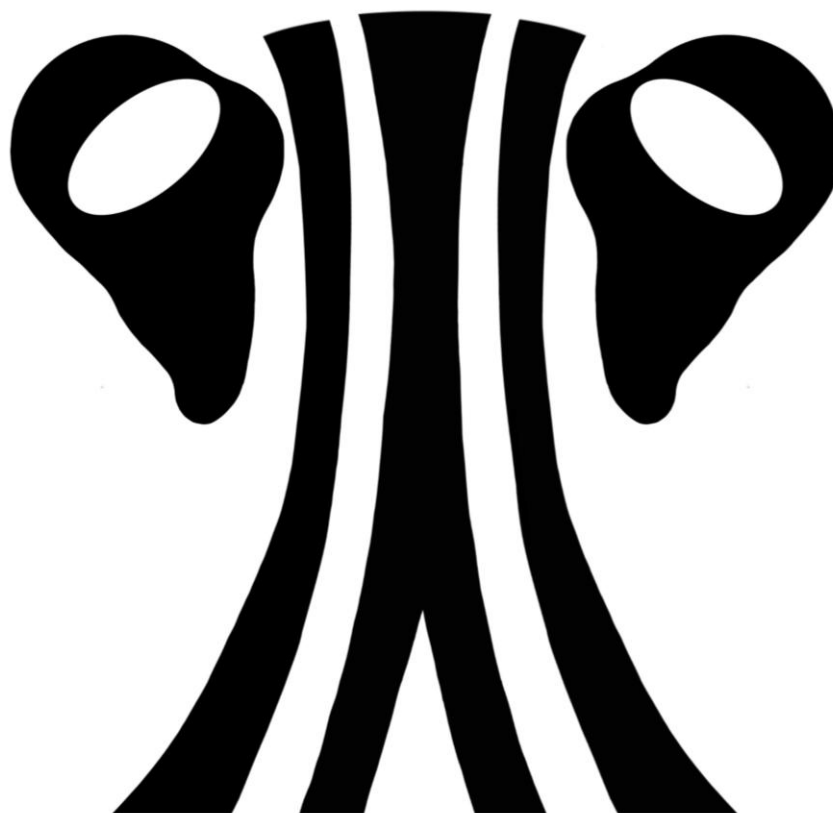


Wild Catcher

(野生動物 AI 検出 & 動画管理アプリ)

ユーザーマニュアル



バージョン 2.0

合同会社 ReCorta

目次

1. 概要
2. インストール・初期設定
 - 2.1 システム要件
 - 2.2 インストール手順（スタンドアロン版）
 - 2.3 インストール手順（開発者向け）
3. インターフェース
 - 3.1 全体レイアウト
 - 3.2 サイドバー
 - 3.3 設定パネル（ドラッグ可能）
 - 3.4 メインエリア
 - 3.5 ログエリア（ドラッグ可能）
 - 3.6 フォルダエクスプローラー
4. AI モデル管理
 - 4.1 モデルの種類
 - 4.2 パイプライン設定
 - 4.3 モデルの追加・削除
 - 4.4 クラス別フィルタリング
5. 処理機能
 - 5.1 処理パイプライン
 - 5.2 画像の処理
 - 5.3 動画の処理
 - 5.4 ファイル名変更ロジック
 - 5.5 CSV 分析レポート
 - 5.6 ディレクトリ構造のミラーリング
6. 設定パネルの詳細
7. 言語オプション
8. 動画処理ツール
9. ライセンスシステム
10. GPU アクセラレーション

11. トラブルシューティング

12. 付録

1. 概要

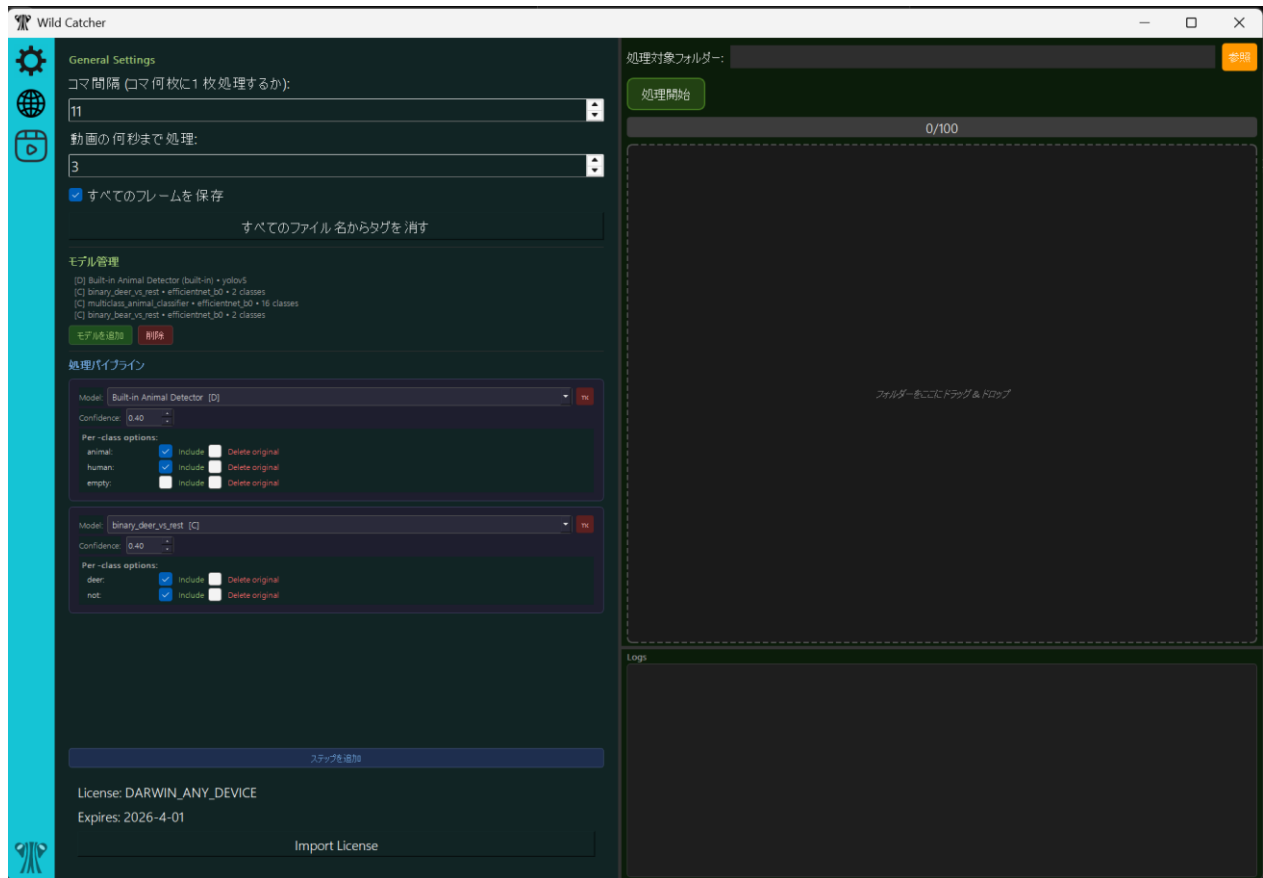
Wild Catcher は、カメラトラップ（自動撮影カメラ）等で取得した画像や動画を自動処理するデスクトップアプリケーションです。AI（深層学習）モデルを用いて、人間・動物・車両・空のフレームを検出・分類し、ファイルの自動整理やリネームを行います。

本アプリケーションは、野生動物の生態調査・保全活動を行う研究者や自然保護団体を主な対象としていますが、セキュリティカメラ映像の仕分けなど幅広い用途にも対応可能です。

主な機能

- **マルチモデルパイプライン**：検出モデル（Detector）＋複数の分類モデル（Classifier）を連結し、段階的に処理を実行。
- **クラス別フィルタリング**：各モデルステップで検出・分類されたクラスごとに、処理の有効/無効をチェックボックスで制御。
- **ドラッグ可能な UI**：設定パネル（横方向）とログエリア（縦方向）をスプリッターで自由にリサイズ可能。
- **フォルダエクスプローラー**：処理対象フォルダ内のファイル一覧をツリー表示。ドラッグ&ドロップでフォルダを指定可能。
- **GPU アクセラレーション**：NVIDIA CUDA（Windows/Linux）および Apple MPS（macOS）による高速推論。
- **多言語対応**：日本語・英語・スペイン語・中国語・韓国語の 5 言語に対応。
- **CSV 分析レポート**：処理結果の統計情報を CSV ファイルとして自動出力。
- **ライセンスシステム**：RSA-2048 署名によるデバイス単位のライセンス管理。
- **動画処理ツール**：統合動画プレーヤーでの動画レビューとリネーム機能。

● クロスプラットフォーム：Windows・macOS・Linux に対応（PyInstaller によるスタンドアロン配布）。



2. インストール・初期設定

2.1 システム要件

項目	要件
OS	Windows 10/11 (64 ビット)、macOS 11+、Linux (Ubuntu 20.04+)
CPU	x86_64 プロセッサ (ARM: macOS Apple Silicon 対応)
RAM	最低 8GB (推奨 16GB 以上)
GPU (推奨)	NVIDIA GPU (CUDA 11.8+対応) / Apple M1 以降 (MPS)
ストレージ	アプリ本体: 約 2GB、モデルファイル: 各 100MB~500MB
Python (開発者のみ)	Python 3.9 以上

💡 **注意:** GPU が無くても CPU モードで動作しますが、処理速度は大幅に低下します。大量のファイル进行处理する場合は GPU の使用を強く推奨します。

2.2 インストール手順 (スタンドアロン版)

1. 配布された WildCatcher フォルダ (またはインストーラー) をダウンロードします。
2. フォルダを任意の場所に展開 (解凍) します。
3. WildCatcher.exe (Windows) または WildCatcher (macOS/Linux) を実行します。
4. 初回起動時にライセンスファイル (.wcl) の読み込みを求められる場合があります。

2.3 インストール手順 (開発者向け)

5. Python 3.9 以上をインストール: <https://www.python.org/downloads/>
6. 必要なパッケージをインストール:
7. アプリケーションのソースコードと assets フォルダをダウンロード。
8. `python detector_animaes_diego.py` で起動。

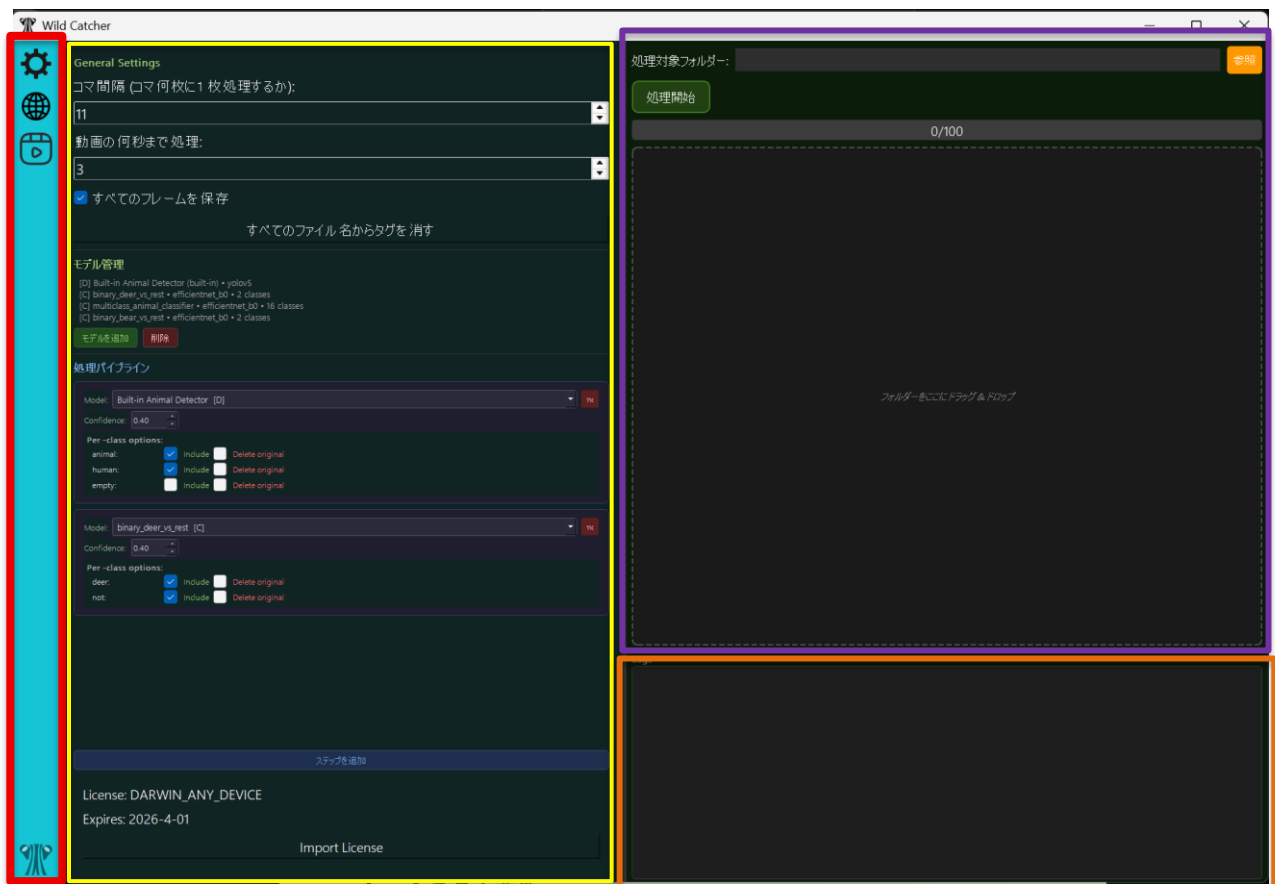
```
pip install PyQt5 opencv-python torch torchvision
```

3. インターフェース

3.1 全体レイアウト

Wild Catcher のインターフェースは、以下の主要なエリアで構成されています。各エリア間にはドラッグ可能なスプリッター（仕切り）があり、自由にサイズを調整できます。

- **サイドバー (左端)** : 設定・言語・動画ツールへのアクセスボタン。
- **設定パネル (左側)** : AI モデルのパイプライン設定、処理パラメータの調整。水平スプリッターで幅を調整可能。
- **メインエリア (中央～右側)** : 入力フォルダの選択、処理ボタン、プログレスバー、フォルダエクスプローラー。
- **ログエリア (下部)** : 処理ログの表示。垂直スプリッターで高さを調整可能。常に表示されます。



3.2 サイドバー

サイドバーはアプリケーションウィンドウの左端に配置された縦長のバーです。以下のボタンが配置されています。

● **設定ボタン (歯車アイコン)** : 設定パネルの表示/非表示を切り替えます。非表示にすると、設定パネルのスプリッターが折りたたまれ、メインエリアが広がります。

● **言語ボタン (地球儀アイコン)** : 言語選択パネルを開きます。5つの国旗アイコンから言語を選択できます。

● **動画処理ツールボタン (再生アイコン)** : 統合動画処理ツール (専用動画処理プレイヤー) を起動します。



3.3 設定パネル (ドラッグ可能)

設定パネルはサイドバーの右側に表示され、水平スプリッターで幅を自由に調整できます。スプリッターのハンドル (5ピクセルの灰色バー) をドラッグすることで、設定パネルとメインエリアの比率を変更できます。ハンドルにマウスを合わせると緑色にハイライトされます。

設定パネルには以下の内容が含まれます (詳細はセクション 4 およびセクション 6 を参照)。

- 処理パラメータ (フレーム間隔、確信度しきい値、動画処理時間)
- チェックボックス設定 (検出データ保存、ファイル削除、全フレーム保存等)
- AI モデルパイプライン管理 (モデルの追加・削除・ステップ設定)
- クラス別フィルタリングチェックボックス
- ライセンス情報表示

3.4 メインエリア

メインエリアは以下の要素で構成されています。

入力ディレクトリの選択

処理対象のフォルダを指定する行です。テキストフィールドにフォルダパスが表示され、「参照」ボタンでフォルダ選択ダイアログを開くことができます。また、フォルダエクスプローラーにフォルダをドラッグ&ドロップしても指定可能です。

処理開始/停止ボタン

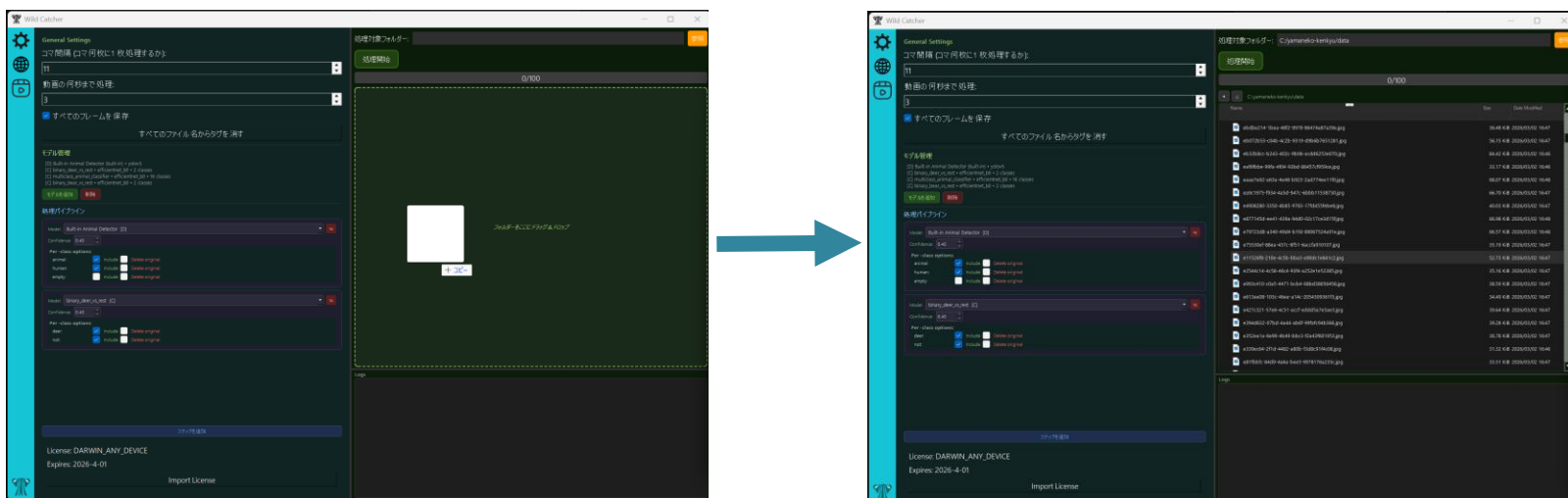
「処理開始」ボタンをクリックすると、処理が開始されます。処理中はボタンのラベルが「処理停止」に変わり、クリックすると確認ダイアログが表示されて処理を中断できます。

プログレスバー

処理の進行状況を「N/M」形式で表示します（N = 処理済みファイル数、M = 合計ファイル数）。緑色のバーが進行に合わせて動的に更新されます。

フォルダエクスプローラー (FolderDropViewer)

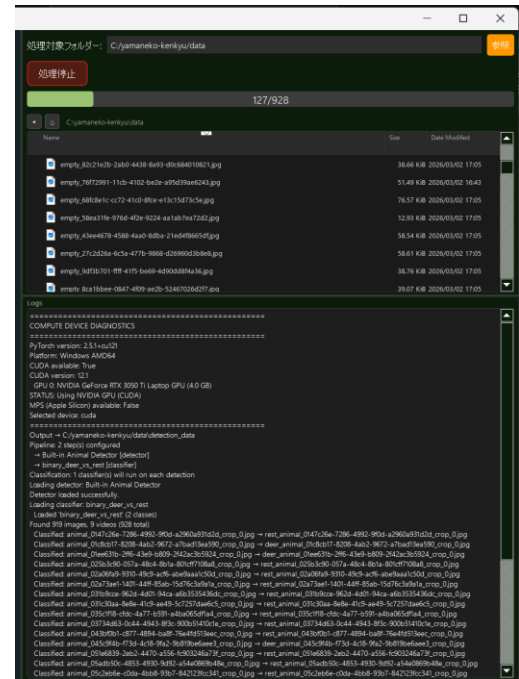
選択した入力フォルダ内のファイル構造をツリー形式で表示します。フォルダをドラッグ&ドロップで指定することもできます。ファイルをダブルクリックするとデフォルトのアプリケーションで開きます。



3.5 ログエリア（ドラッグ可能）

ログエリアはメインエリアの下部に常時表示されます。垂直スプリッターのハンドルを上下にドラッグすることで、ログエリアの高さを調整できます。スプリッターの位置はセッション間で保存されます。

ログには処理中のリアルタイム情報が表示されます：処理中のファイル名、検出結果、リネーム操作、エラー情報など。ログは読み取り専用のテキストエリアです。



3.6 フォルダエクスプローラー

入力フォルダを選択すると、フォルダエクスプローラーにその中のファイル一覧がツリー構造で表示されます。サブフォルダも含めて表示され、画像ファイルと動画ファイルが確認できます。

- フォルダのドラッグ&ドロップ対応
- ファイルのダブルクリックで外部アプリケーションを起動
- 処理後のファイル名変更がリアルタイムで反映

4. AI モデル管理

4.1 モデルの種類

Wild Catcher は 2 種類の AI モデルを使用します。これらを「パイプライン」として連結し、段階的に処理を実行します。

モデルタイプ	説明
Detector (検出モデル)	画像内の物体 (動物・人間・車両) を検出し、バウンディングボックスで位置を特定。YOLO V5 ベース。
Classifier (分類モデル)	種判別するカスタムモデル。ResNet / EfficientNet 対応。

対応アーキテクチャ

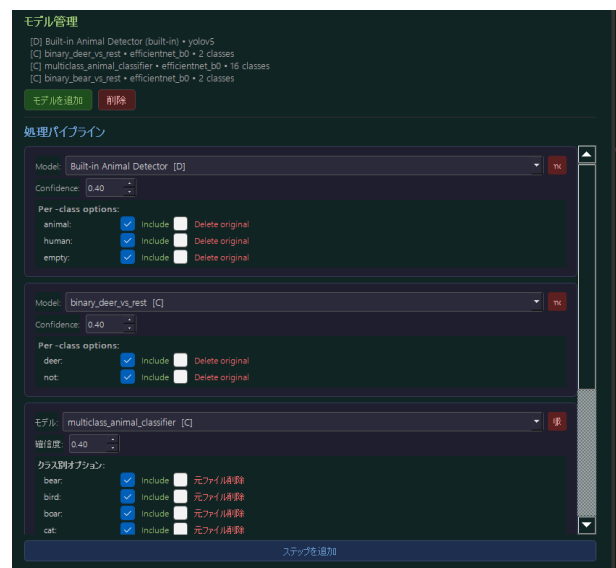
アーキテクチャ	対応バリエーション
ResNet	resnet18, resnet34, resnet50, resnet101
EfficientNet	efficientnet_b0 ~ efficientnet_b4

バイナリ分類モデル (出力ニューロンが 1 つ) にも対応しています。この場合、シグモイド関数で確率を計算し、しきい値 0.5 で 2 クラスに分類します。

4.2 パイプライン設定

パイプラインは設定パネル内で管理されます。各モデルが「ステップ」として表示され、順番に処理が実行されます。


- Step 1: Detector — 画像全体から物体を検出し、クロップ画像を生成
- Step 2: Classifier A — クロップ画像を種レベルで分類 (例: シカ、イノシシ等)
- Step 3: Classifier B — 必要に応じて追加の分類 (例: 特定種の二値分類)



4.3 モデルの追加・削除

設定パネル下部の「モデル追加」ボタンから新しいモデルを登録できます。

9. 「モデル追加」ボタンをクリック
10. .pt / .pth ファイルを選択
11. モデル名を入力
12. タイプ (detector / classifier) を選択
13. 分類モデルの場合、クラス名の入力を求められることがあります (カンマ区切り)
14. 登録完了後、パイプラインに自動追加

 **注意:** モデルファイルは自動的に models/ フォルダにコピーされます。アーキテクチャ (ResNet/EfficientNet) とヘッドタイプは自動検出されます。

モデルの削除は、各ステップカードの「×」ボタンから行えます。

4.4 クラス別フィルタリング

各モデルステップには、そのモデルが出力する全クラスに対応するチェックボックスグリッドが表示されます。チェックを外すと、そのクラスに分類されたファイルは処理 (リネーム・出力) の対象外になります。

検出モデル (Detector) の場合、以下の固定プレフィックスが使用されます。

クラス	プレフィックス
動物 (animal)	animal_
人間 (human)	human_
空 (empty)	empty_

分類モデルの場合、クラス名はモデルのチェックポイントから自動検出されます。

5. 処理機能

5.1 処理パイプライン

処理は以下のパイプラインで実行されます。

15. 入力フォルダ内の画像・動画ファイルをスキャン（既にプレフィックス付きのファイルはスキップ）
16. Step 1 (Detector): 各ファイルに対して物体検出を実行
17. 確信度しきい値でフィルタリング
18. クラス別チェックボックスに基づいてフィルタリング
19. Step 2+ (Classifier): 検出された物体のクロップ画像を分類
20. 分類結果に基づいてファイルをリネーム
21. 設定に応じて検出データ（クロップ画像・アノテーション画像）を保存
22. CSV 分析レポートを出力

5.2 画像の処理

画像ファイル（.jpg, .jpeg, .png, .bmp, .gif, .tiff, .tif, .webp）が処理対象です。

- OpenCV で画像を読み込み
- AI 検出モデルで処理（バウンディングボックス + 確信度を取得）
- 確信度しきい値でフィルタリング
- 検出カテゴリに基づいてプレフィックスを決定
- 「タグでファイル名を変更」が有効な場合、ファイルをリネーム
- 「detection_data フォルダを作成」が有効な場合、検出画像とクロップ画像を保存

5.3 動画の処理

動画ファイル

（.avi, .mp4, .mov, .mkv, .wmv, .flv, .mpg, .mpeg, .m4v, .3gp, .webm, .ts, .mts, .m2ts, .asf）が処理対象です。

- 「フレーム間隔」設定に従ってフレームを抽出（例：16 フレームごと）
- 「動画の処理時間」設定に従って処理範囲を制限
- 各フレームを AI 検出モデルで処理
- 確信度しきい値でフィルタリング

- 最高確信度のフレーム（または全フレーム）を記録
- ファイルリネーム・検出データ保存を実行
- 有効な検出がない場合、設定に応じて動画を削除

5.4 ファイル名変更ロジック

処理の結果に基づいて、ファイル名の先頭にプレフィックスが追加されます。

検出結果	プレフィックスの例
動物が検出	animal_元のファイル名.jpg
人間/車両が検出	human_元のファイル名.jpg
何も検出されない	empty_元のファイル名.jpg（設定による）
分類モデルで種を特定	deer_元のファイル名.jpg（種名がプレフィックスになる）

「すべてのファイル名からタグを削除」ボタンを使用すると、処理後にプレフィックスを一括で除去できます。

5.5 CSV 分析レポート

処理完了後、入力フォルダ内に CSV ファイルが自動生成されます。レポートには以下の情報が含まれます。

- ファイル名
- 検出されたクラス
- 確信度スコア
- 処理日時
- リネーム後のファイル名

5.6 ディレクトリ構造のミラーリング

入力フォルダにサブフォルダが含まれている場合、出力（detection_data フォルダ）にも同じディレクトリ構造が再現されます。これにより、元のフォルダ構成を維持したまま処理結果を整理できます。

6. 設定パネルの詳細

設定パネルには、処理動作を制御するためのさまざまなパラメータが含まれています。

フレーム間隔

コントロール：スピンドックス（数値入力） / デフォルト値：16

動画のフレームを処理する際にスキップするフレーム数。例：16 に設定すると 16 フレームごとに検出処理を実行。値を小さくするとより多くのフレームを処理しますが、処理時間が増加します。

確信度のしきい値

コントロール：ダブルスピンドックス（0.0~1.0） / デフォルト値：0.40

検出が有効と見なされる最小の確信度。低く設定すると検出漏れが減りますが、誤検出が増える可能性があります。推奨範囲：0.15~0.60

動画の処理時間（秒）

コントロール：スピンドックス（数値入力） / デフォルト値：5

各動画の最初の何秒間を処理するかを指定。長い動画の全体を処理する必要がない場合に便利。

「detection_data」フォルダを作成

コントロール：チェックボックス / デフォルト値：OFF

チェックを入れると、検出結果の画像（バウンディングボックス付き）とクロップ画像が detection_data フォルダに保存されます。

検出がない動画を削除

コントロール：チェックボックス / デフォルト値：OFF

有効な検出がない動画ファイルを処理後に自動削除します。注意：この操作は取り消せません。

すべてのフレームを保存

コントロール：チェックボックス / デフォルト値：OFF

最高確信度のフレームだけでなく、検出されたすべてのフレームを保存します。

タグでファイル名を変更

コントロール：チェックボックス / デフォルト値：ON

検出カテゴリに基づいてファイル名の先頭にプレフィックスを追加します。

すべてのファイル名からタグを削除

コントロール：ボタン | デフォルト値：—

入力フォルダ内のファイルから、設定されたプレフィックスを一括で除去します。

7. 言語オプション

Wild Catcher は多言語対応で、以下の 5 つの言語をサポートしています。

言語	国旗アイコン
日本語	JP 日本
スペイン語	ES スペイン
中国語	CN 中国
英語	GB イギリス
韓国語	KR 韓国

言語の変更手順

23. サイドバーの言語ボタン（地球儀アイコン）をクリック
24. 言語選択パネルが表示され、各言語を表す国旗が表示される
25. 希望の言語に対応する国旗をクリック
26. アプリケーション全体のラベル・メッセージが選択した言語に即座に更新

8. 動画処理ツール

動画処理ツール) は、動画ファイルを閲覧および管理するための統合ツールです。サイドバーの再生アイコンボタンから起動できます。

インターフェース要素

- **動画フレーム**：動画が表示されるメインエリア。
- **「動画を開く」ボタン**：動画ファイルを選択するダイアログを開きます。
- **前の動画 / 次の動画ボタン**：現在のフォルダ内の前後の動画に移動します。
- **再生/一時停止ボタン**：動画の再生と一時停止を切り替えます。
- **リプレイボタン**：現在の動画を最初から再生します。

- **速度制御**：コンボボックスから 1.0x ~ 32x の再生速度を選択できます。
- **ファイル名変更テキストボックス**：新しいファイル名を入力し、「OK」ボタンで現在の動画をリネームします。



9. ライセンスシステム

Wild Catcher は RSA-2048 デジタル署名によるライセンスシステムを採用しています。ライセンスファイル (.wcl) により、アプリケーションの使用を制御します。

ライセンスの種類

種類	説明
ユニバーサルライセンス	デバイス ID = "ANY"。任意のデバイスで使用可能。
デバイスロックライセンス	特定のデバイスのフィンガープリントに紐づいたライセンス。そのデバイスでのみ使用可能。

ライセンスの適用手順

27. アプリケーションを起動すると、設定パネルの下部にライセンス情報が表示されます。
28. ライセンスが未設定の場合、「ライセンスを参照」ボタンをクリックし、.wcl ファイルを選択します。
29. デバイスロックライセンスの場合、アプリに表示される「デバイス ID」を販売者に送信してください。
30. 販売者がそのデバイス ID 用のライセンスファイルを発行します。

ライセンス情報の確認

設定パネルの下部に以下の情報が表示されます。

- ライセンス名 (組織名/個人名)
- 有効期限 ("never" = 無期限)
- デバイス ID (現在のデバイスのフィンガープリント)

10. GPU アクセラレーション

Wild Catcher は GPU を使用した高速推論をサポートしています。以下の GPU バックエンドに対応しています。

バックエンド	対応環境
CUDA	NVIDIA GPU (Windows / Linux)。CUDA Toolkit 11.8 以上が必要。
MPS	Apple Silicon (macOS M1/M2/M3)。macOS 12.3 以上が必要。
CPU	GPU が利用できない場合の自動フォールバック。

アプリケーションは起動時に自動的に最適なデバイスを検出します。ログに「Using device: cuda」「Using device: mps」「Using device: cpu」のいずれかが表示されます。

注意：GPU 利用には、対応するバージョンの PyTorch がインストールされている必要があります。スタンドアロン版では、ビルド時の設定に依存します（CUDA 版/MPS 版/CPU 版）。

11. トラブルシューティング

アプリケーションが起動しない

全ての依存関係が正しくインストールされていることを確認してください。スタンドアロン版の場合、アンチウイルスソフトによるブロックがないか確認してください。Windows の場合、Visual C++ Redistributable が必要な場合があります。

検出が行われない

確信度のしきい値を低く調整してください（推奨：0.15～0.40）。メディアファイルに検出可能な物体が含まれていることを確認してください。モデルファイル（.pt）が正しく配置されているか確認してください。

処理速度が遅い

GPU が認識されているか確認してください（ログに "Using device: cuda/mps" と表示されるか）。フレーム間隔の値を大きくすることで動画処理の速度が向上します。動画の処理時間を短く設定すると、長い動画の処理が高速化されます。

ファイル名を変更できない

ファイルが他のアプリケーションで開かれていないことを確認してください。ファイルの書き込み権限があることを確認してください。同じ名前前のファイルがディレクトリに既に存在しないか確認してください。

動画処理ツールが起動しない

VLC メディアプレーヤーのライブラリがアプリケーションディレクトリにあることを確認してください。

ライセンスエラー

ライセンスファイル（.wcl）が正しいデバイスに対して発行されたものであることを確認してください。有効期限が切れていないか確認してください。ハードウェアの変更（ネットワークカードの追加等）によりデバイス ID が変わる場合があります。

モデルの読み込みエラー

モデルファイルのアーキテクチャが対応しているか確認してください（ResNet / EfficientNet）。バイナリ分類モデルの場合、2つのクラス名が必要です（例："deer, rest"）。チェックポイントに `model_state_dict` または `state_dict` キーが含まれている必要があります。

12. 付録

対応ファイル形式

種別	拡張子
画像	.jpg, .jpeg, .png, .bmp, .gif, .tiff, .tif, .webp
動画	.avi, .mp4, .mov, .mkv, .wmv, .flv, .mpg, .mpeg, .m4v, .3gp, .webm, .ts, .mts, .m2ts, .asf

キーボードショートカット

現バージョンではキーボードショートカットは実装されていません。今後のアップデートで追加予定です。

ファイル構成

ファイル	説明
<code>detector_animales_diego.py</code>	メインアプリケーション (UI・起動)
<code>wc_widgets.py</code>	パイプラインウィジェット・モデル管理 UI
<code>wc_models.py</code>	モデルレジストリ・ロード・分類処理
<code>wc_processing.py</code>	画像/動画処理パイプライン
<code>wc_license.py</code>	ライセンス検証
<code>generate_license.py</code>	ライセンス生成ツール (管理者用)
<code>video_player.py</code>	動画処理ツール
<code>assets/</code>	アイコン・国旗画像等のリソース
<code>models/</code>	ユーザー登録モデルファイル

パフォーマンスの考慮事項

- 大量のファイル (数千枚以上) や高解像度の動画を処理する場合、GPU の使用を強く推奨します。
- フレーム間隔を大きくすると動画処理速度が向上しますが、検出漏れのリスクが増加します。
- 動画の処理時間を短く設定すると、長時間の動画でも効率的に処理できます。

- 分類モデルを追加するとパイプラインのステップ数が増え、1 ファイルあたりの処理時間が増加します。

連絡先情報

サポートや問題の報告については、以下にご連絡ください。

サポートメール：DiegoAlonsoJapan@gmail.com